Unsupervised Learning of Disentangled Video Representation for Future Frame Prediction

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

Ujjwal Tiwari
2019701016
ujjwal.t@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Unsupervised Learning of Disentangled Video Representation for Future Frame Prediction" by Ujjwal Tiwari, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Anoop Namboodiri

To Supreme Personality of Godhead, Krishna

Acknowledgments

I want to thank my supervisor, Dr. Anoop Nambboodiri, for sharing his wisdom and guiding me as a research scholar at IIIT, Hyderabad. I shall forever remain indebted to him for imparting numerous teachings which have shaped my understanding of computer vision and research in general. I would also like to thank my teachers, Prof. C.V. Jawahar, Dr. Avinash Sharma, and Dr. Ravi Kiran Sarvadevabhatla, for their significant contributions to my understanding of machine learning and computer vision.

I am grateful to the research community at the Center for Visual Information Technology lab. Working with the innumerable bright minds here has contributed significantly to my intellectual and professional growth. It has made me realize that hard work coupled with compassionate dedication can solve challenging problems in life and pattern recognition. I have thoroughly enjoyed working at IIIT Hyderabad as I find research an incredibly enriching experience. I want to thank my friend P. Aditya Sreekar who made my life at IIIT Hyderabad a journey to remember. He has always been a source of constant encouragement and motivation. I will forever cherish our never-ending technical discussions, without which it would have been impossible to accomplish meaningful research without losing perspective.

Finally, I would like to thank my mother, father, and brother for their unwavering support. You mean more to me than you could realize. I want to thank my father for being the best life coach possible and my mother for being the voice of pragmatism and faith whenever I got stuck with a problem. I would also like to thank my wife, Ayushi Mishra, for always being there for me patiently. You have always inspired me to be a better human by teaching me the correct value system, which has helped me set challenging professional goals and strive for success. At long last, I would like to thank everyone who was not acknowledged above and has been in any capacity a part of my journey thus far.

Abstract

Predicting what may happen in the future is a critical design element in developing an intelligent decision-making system. This thesis aims to shed some light on video prediction models that can predict future frames of a video sequence by observing a set of previously known frames. These models learn video representations encoding the causal rules that govern the physical world. Hence, these models have been extensively used in the design of various vision-guided robotic systems. These models also have applications in reinforcement learning, autonomous navigation, and healthcare.

Video frame prediction remains challenging despite the availability of large amounts of video data and the recent progress of generative modeling techniques in synthesizing high-quality images. The challenges associated with predicting future frames can be attributed to two significant characteristics of video data - the high dimensionality of video frames and the stochastic nature of the motion exhibited in these video sequences.

Existing video prediction models solve the challenge of predicting frames in high-dimensional pixel space by learning a low-dimensional disentangled video representation. These methods factorize video representations into dynamic and static components. The disentangled video representation is subsequently used for the downstream task of future frame prediction.

In Chapter 3, we propose a mutual information-based predictive autoencoder, MIPAE, a self-supervised learning framework. The proposed framework factorizes the latent space representation of videos into two components - static content and a dynamic pose component. The MIPAE architecture comprises a content encoder, pose encoder, decoder, and a standard LSTM network. We train MIPAE using a two-step procedure, such that in the first step, the content encoder, pose encoder, and decoder are trained to learn disentangled frame representations. The content encoder is trained using the slow feature analysis constraint, while the pose encoder is trained using a novel mutual information loss term to achieve proper disentanglement. In the second step of our training methodology, we train an LSTM network to predict the low-dimensional pose representation of future frames. The predicted pose and learned content representations are decoded to generate future frames of a video sequence.

In this thesis, we present detailed qualitative and quantitative results to compare the performance of our proposed MIPAE framework. We evaluate our approach on standard video prediction datasets like DSprites, MPI3D-real, and SMNIST using various visual quality assessment metrics, namely LPIPS, SSIM, and PSNR. We also present a metric based on mutual information gap, MIG, to quantitatively evaluate the degree of disentanglement between the factorized latent variables - pose and content. MIG

score is subsequently used for a detailed comparative study of the proposed framework with other disentanglement-based video prediction approaches to showcase the efficacy of our disentanglement approach. We conclude our analysis by showcasing the visual superiority of the frames predicted by MIPAE.

In Chapter 4, we explore the paradigm of stochastic video prediction models, which aim to capture the inherent uncertainty in real-world videos by using a stochastic latent variable to predict a different but plausible sequence of future frames corresponding to each sample of the stochastic latent variable. In our work, we modify the architecture of two stochastic video prediction models and apply a novel cycle consistency loss term to disentangle the video representation space into pose and content factors and model the uncertainty in the pose of various objects in the scene, to generate sharp and plausible frame predictions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer science researchers motivated to develop artificial general intelligence (AGI) have always been intrigued by one exceptional capability of humans, which is their ability to imagine vivid dreams and create art forms like literature, paintings, and music. In the recent past, there has been an exponential rise in computational capabilities, leading to large-scale advancements in generative modeling techniques such as VAEs [37] and GANs [24]. Deep learning models based on VAEs and GANs have successfully generated realistic images and videos. In this thesis, we study the core computer vision problem of visual prediction, which has been studied in several contexts, such as early recognition, activity prediction, trajectory forecasting, and video frame prediction. In particular, we focus our attention on the task of video frame prediction, that is, to predict a sequence of future video frames given a sub-sequence of context frames. Machine learning models capable of hallucinating future frames have been shown to find large-scale applications in anomaly detection [44, 52], healthcare[60], and visual robotics[17, 18].

However, video frame prediction is a challenging generative modeling task. Two of the most significant challenges associated with video frame prediction are: (i) learning the non-linear transformations that map a set of context frames to a sequence of future frames is difficult in the high dimensional image-pixel space, (ii) it is imperative to learn a model of uncertainty that captures the inherent stochasticity exhibited in real-world videos to capture the full distribution of plausible outcomes to produce sharp future frame predictions. Many concurrent approaches overcome the difficulty associated with making predictions in the high dimensional image-pixel space by disentangling video representation into two components, one of which is time-independent and remains approximately constant throughout the clip and another that captures the video's low dimensional temporal dynamics and is easy to predict. Another prominent direction of research studies the paradigm of stochastic video prediction, which explores the application of stochastic latent variable models like variational autoencoders (VAEs) and generative adversarial nets (GANs) to learn a probabilistic model of uncertainty to capture the stochasticity in dynamics of real-world videos. While stochastic video prediction models also learn to factorize video representation into two components, they do so in deterministic and stochastic parts rather than static and dynamic components.

In this thesis, we propose two video prediction models; one is a novel mutual information-based predictive autoencoder (MIPAE), which forms the primary subject matter of chapter 3. The proposed MIPAE framework leverages the temporal structure in the latent generative factors of a video to disentangle the representation space of video frames into low dimensional time-dependent, pose, and time-independent content latent factors. Our disentanglement approach assumes that the semantic information in the video remains fixed (i.e., the number and appearance of the constituent objects in the video sequences do not change with time). In contrast, the position of these objects keeps changing from one frame to another. Based on the aforementioned simplifying assumption, in our approach, we train an autoencoder network comprised of a content encoder, a pose encoder, and a frame decoder network.

The content encoder in our MIPAE framework is trained using the slow feature analysis constraint[87], which is motivated by the assumption that the content information in videos should vary slowly over time, encouraging temporally close video frames to have similar content encodings. However, slow feature analysis and the requirement that the content representation of a known frame can be combined with predicted pose representations to generate future frames are insufficient for proper pose/ content disentanglement. Thus, we propose a novel *mutual information* loss term to constrain the pose encoder to ensure that the pose latent representations do not contain any content information. After training the autoencoder network, a standard LSTM network is trained to predict the low-dimensional pose latent variables of future frames. The predicted pose and content factors are decoded to generate future frames.

As discussed earlier, stochastic video prediction models use the latent variables in variational autoencoders and generative adversarial nets to capture the mode of uncertainty associated with the task of video frame prediction. These models tend to factorize video representation into a deterministic and another stochastic component. These models are stochastic because they predict a plausible but divergent sequence of video frames corresponding to each sample of their latent variable.

In chapter 4, we present our second essential contribution in which we learn to disentangle video frame representation into stochastic pose and deterministic content factors using two stochastic video prediction models (SVG-LP [13] and SAVP [40]). This chapter also studies the benefits of using cycle consistency loss terms in training stochastic video prediction models. Our disentanglement approach is based on the insight that the major component of stochasticity in real-world videos arises from the uncertainty in the motion of various objects in the scene. We base our insight on the critical assumption that the objects and scenes in the videos remain fixed. Based on this assumption, we condition the generator of two stochastic video prediction models with time-independent content encodings. This leads to the disentanglement of video representation into deterministic content and low-dimensional stochastic pose representations. We also incorporate a *cycle consistency* loss term in their training objective, leading to sharp long-range video frame predictions. The motivation behind cycle consistency loss is that future frame predictions are more plausible and realistic if they can be used to reconstruct the previous frames.

The rest of this chapter is organized as follows- in Section 1.1, we discuss the advantages of representation learning, specifically the advantages of learning disentangled latent representations from

Figure 1.1 Flowcharts comparing classical computer vision algorithms and representation learning-based approach [23]. Blue boxes in the diagram indicate components capable of learning from data.

unlabelled data in contrast to more traditional computer vision algorithms that rely on handcrafted features. This section is followed by a brief discussion on the reasons behind the enormous impact of deep learning in solving computer vision tasks in Section 1.2. In Section 1.4, we discuss three widely popular generative modeling techniques, which are essential for a complete understanding of the research works presented as a part of this thesis. Section 1.5 sheds some light on mutual information and the current state of research on sample-based mutual information estimation. We end this introductory chapter with the final Section 1.6, which highlights the contributions of this thesis.

1.1 Representation Learning

Classical computer vision research focused on engineering handcrafted features like pixel intensity values, histograms of oriented gradients[12], SIFT[48], SURF[4], and Bag of Visual Words[88]. These hand-crafted features were given as input image descriptors to various machine learning algorithms such as decision trees[68], support vector machines[75], logistic regression[31], and nearest neighbor algorithms [20] to solve complex downstream computer vision tasks. Figure 1.1 (left) depicts the schematic diagram of classical computer vision methods used to solve various computer vision problems such as face recognition, image classification, object detection, instance segmentation, and semantic segmentation. These traditional computer vision algorithms had solid mathematical underpinnings, and one could explain their performance. Still, these methods failed to solve many simple computer vision problems that humans could easily solve. This limitation in their performance can be attributed to their dependence on hand-crafted features since task-specific image features are difficult to describe, let alone formulate mathematically. For example, suppose that we would like to train a system that detects the presence of a cat in an image. We know that cats have pointed ears, so we might like to use the presence of pointed ears as a feature. Unfortunately, describing exactly what a pointed ear looks like in terms of pixel intensity values is challenging. While a cat's ear has a simple geometric shape, the image may be complicated due to occlusion, illumination, background clutter, and large-scale intra-class variation. Moreover, manually designing features for solving complex computer vision tasks requires a huge amount of human labor; it can take decades worth of research to make any significant progress. Any limitation in extracting relevant features limits the performance of the subsequent machine-learning algorithm. As the performance of these simple machine learning algorithms depends heavily on the representation of the data that is given as input.

However, in the past few decades, computer vision research has made tremendous progress in training artificially intelligent systems capable of learning abstract task-specific representations from large amounts of visual data. Modern computer vision algorithms, Figure 1.1(right), use deep learning models like multi-layer perceptrons, convolutional networks, and recurrent neural networks to learn task-specific hierarchical representations of the data. These deep learning-based methods also learn a mapping between the learned representations and the desired output. This approach of learning task-specific representation from large amounts of visual data to build artificially intelligent systems is known as representation learning. While training these deep learning algorithms is computationally expensive, these systems perform inference in real-time and solve tasks they have learned the representations for with human-like accuracy. Furthermore, the application of deep learning models to solve computer vision tasks is a growing trend because of the recent advancements in computational capabilities and the abundance of multimedia data.

1.2 Deep Learning

Deep learning is a flexible and powerful machine learning paradigm that has been widely used to learn task-specific representations of data for various downstream computer vision tasks. Deep learning enables the learning algorithm to represent data samples as a nested hierarchy of concepts in which each concept is defined as a differentiable function of simpler concepts. The simplest example of an artificial neural network is the feed-forward neural network, also known as multilayer perceptron (MLP) which is a composite of many simple functions mapping a set of inputs to output values. We can think of each layer in an MLP as a different mathematical function building complex and more abstract representations from less abstract ones.

Understanding an image from a grid of pixel intensity values is difficult for a computer. For example, let us consider the object recognition task, where the computer vision algorithm has to learn a function mapping from a set of pixel intensity values to the identity of an object. While learning this function seems to be an overwhelming task if tackled directly, it can be resolved easily by using a deep learning model which learns to encode the complicated function as a series of nested simple mappings. Each of these simple mappings can be described by a different layer of a deep learning model. The observed inputs are presented at the visible layer, followed by a series of hidden layers that abstract away an increasingly complex set of features from an image. During training, the weights of the hidden layers are determined to learn task-specific concepts useful for explaining key relationships within the observed data samples. The learned task-specific concepts can then be used to solve the downstream object recognition task.

1.2.1 Convolutional Neural Networks

This section discusses a specific class of artificial neural networks known as the convolutional neural networks (CNNs)[23]. CNNs have played a tremendous role in the popularity of machine learning algorithms, specifically in computer vision. They are probably one of the first biologically inspired deep learning models to remain at the forefront of driving artificial intelligence forward in building complicated commercial applications. These neural networks have found large-scale applications in autonomous navigation, biometrics, and remote sensing, amongst many others. The convolutional neural network architecture provides important benefits over fully connected neural nets, such as sparse interactions, parameter sharing, and equivariant representations. The improvements in computational efficiency due to sparse interaction between hidden layers and parameter sharing are usually quite significant, making them comparatively much easier to train. This also enables a developer to effectively tune the model hyperparameters on large datasets by running multiple experiments. Convolutional neural networks have been highly successful with tasks specific to the grid-structured topology of images and videos.

In our MIPAE framework, both the pose and the content encoder are instances of deep convolutional neural networks. We next turn our attention to a very powerful specialization of artificial neural net-

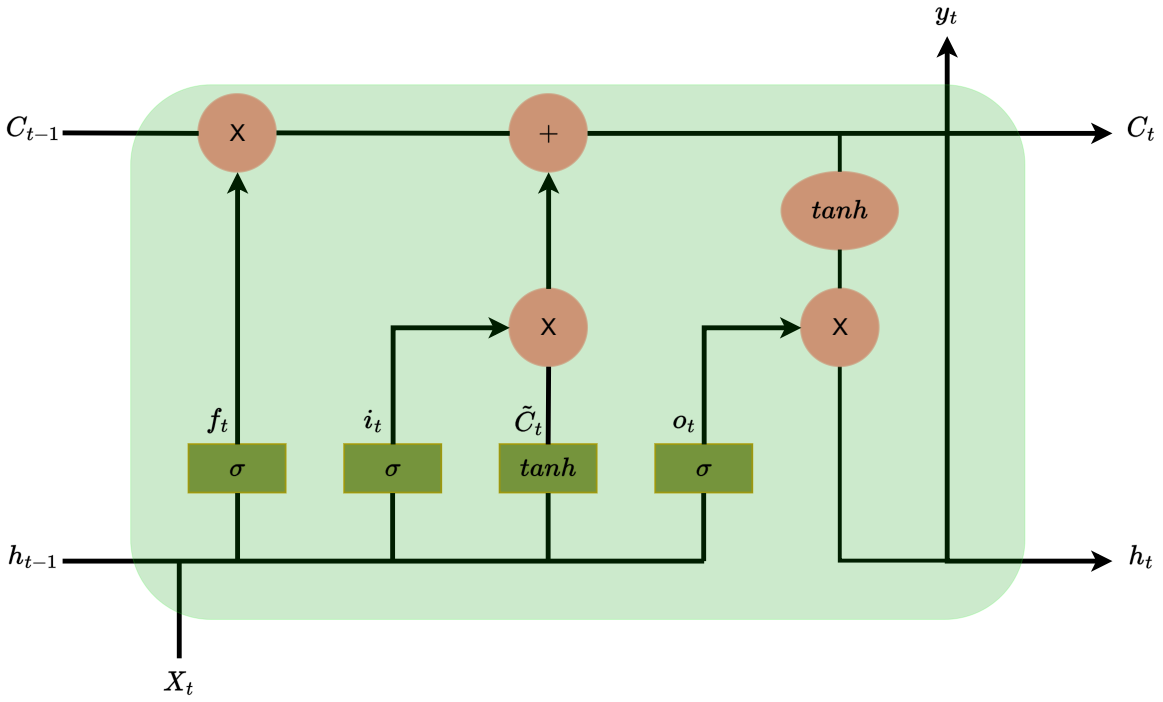


Figure 1.2 Block diagram of the LSTM cell [85]: These LSTM cells are connected in a recurrent manner. In this case, the input features are computed using a regular neuron, and its values get accumulated based on the output of the sigmoid unit. Each LSTM cell has a state unit with a linear self-loop whose weight is controlled by a forget gate. All gates have sigmoid activation function. The output gate controls the output of each cell. Note that the input unit can have any nonlinearity with varying degrees of complexity. The state unit can also be, in some cases, used as an extra input to the gating units.

works: recurrent neural networks (RNNs). In our work in Chapter 3, we use a particular type of RNN, long short-term memory [30], to predict the pose latent variable for future frames.

1.2.2 Recurrent Neural Networks

The recurrent neural network is a particular artificial neural network designed to process sequential data. Just as the architecture of a convolutional neural network is equipped to handle data with grid-like topology, the recurrent neural network helps model sequential data x_1, x_2, x_3 . . . , x_τ. In contrast to deep feed-forward neural networks, which have separate parameters for each time step, a recurrent neural network shares parameters across time. Parameter sharing enables the recurrent network to generalize to sequences with variable lengths not seen during training. Sharing parameters across different parts of the recurrent model is an essential idea of considerable practical significance, which enables a recurrent network to share statistical strength across different sequence lengths and different

positions in time. It is especially important when a specific information can occur at multiple positions within the sequence.

The Long Short-Term Memory network (LSTM) [30] is a type of recurrent neural network which is highly effective in learning the temporal dynamics of a sequence. LSTMs avoid the vanishing and exploding gradient problem primarily associated with RNNs. The LSTM cell is augmented with recurrent gates widely known as forget gates, Figure 1.2. The forget gates enable an LSTM network to handle tasks requiring memory of an event that happened thousands or millions of discrete time steps earlier. LSTMs work well in long delays between significant events and can handle signals with mixed low and high-frequency components. Unlike other widely used statistical methods for sequence modeling, like the hidden Markov models, LSTMs can learn to recognize context-sensitive information from long sequences of video frames. Equipped with the advantages associated with modeling the temporal dynamics of a video sequence by exploiting the recurrent structure of LSTMs, in our MIPAE framework, we use a recurrent pose prediction network, a standard two-layer LSTM network with 256 cells.

1.3 Deep Learning of Disentangled Representations

The autoencoder framework [43] is a quintessential example of learning effective representations from unlabelled data using artificial neural networks (ANNs). An autoencoder consists of two deep neural networks- an encoder and a decoder network. The objective of the encoder network is to learn a parametric function that converts data samples to a low-dimensional representation, and the decoder network aims to reconstruct the original data from their new representation. Many different versions with varying model architectures and optimization objectives of the plain vanilla autoencoder exist in the literature, for example, sparse[55], denoising[80], and contrastive[8]. The main objective behind the different versions of the vanilla deep autoencoder is the introduction of useful structural properties in the learned representations of data such that representation can be effectively used in solving a wide range of computer vision problems.

One of the significant objectives in designing a pipeline for learning features from unlabelled data is to *disentangle* the representation space into latent factors of data generations (factors that can explain the sources of variation within the observed data samples). Disentangled representation learning approaches provide inferred constructs that can explain the observed data. These approaches have been used to solve a wide range of computer vision problems. For example, in analyzing an image, the factors of variation can include- the semantic information present in an image, the position of various objects in the scene, direction and brightness of the illuminating source. Hence, most applications benefit from learning representation with disentangled factors of variation.

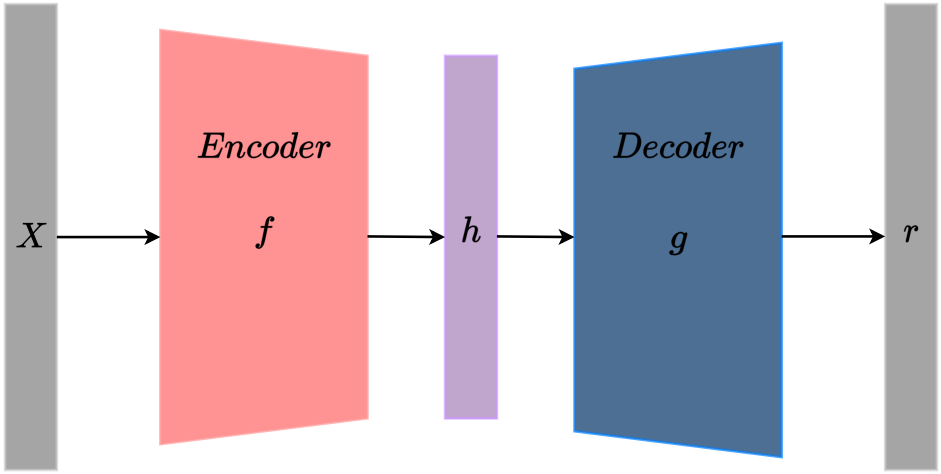


Figure 1.3 Block diagram representing the general structure of an autoencoder network [84]: The encoder function is a parametric neural network that maps an input data sample x to a lower dimensional latent representation h. Similarly, the decoder neural network reconstructs the output r. The autoencoder has two components: the encoder function f and the decoder function g.

1.4 Generative Models

This section briefly discusses a subset of prevalent deep generative modeling techniques. The concepts discussed in this section play a central role in the research works presented in Chapter 3 and in Chapter 4 of this thesis.

1.4.1 Autoencoders

An autoencoder is a neural network architecture used to learn useful data representations, especially in the unsupervised learning setting. As shown in Figure 1.3, an autoencoder network consists of an encoder function and a decoder function, both parameterized as artificial neural networks. The encoder network learns a mapping, h = f(x), to compress the input data to a lower dimensional latent representation. The decoder network produces a reconstruction by taking as input the encoder network's output such that r = g(h).

In an ideal case, an autoencoder network can be designed to learn salient features of the underlying data distribution. This can be done by limiting the model complexity of the encoder-decoder network and tuning the dimensions of the latent code. However, problems start to occur when the dimensions of the latent code are either greater than or equal to the dimensions of the input data. In these situations, even in a linear encoder and decoder network, the decoder can trivially learn to copy the input to output without learning anything useful. In practice, rather than limiting the model's capacity by choosing a shallow encoder-decoder network or by keeping latent space dimensions small, specific loss func-

tions are used to regularise the latent code. These loss functions constrain the latent code and induce desired structure to the learned representation space by enforcing nice properties. Some of the most desired properties of data representations are sparsity, robustness to missing inputs, noisy data, and small derivatives of learned representation [23].

In Chapter 3, we present a special type of predictive autoencoder to learn a disentangled representation of videos for future frame prediction. The proposed framework consists of two independent content and pose encoders. While the content encoder aims at learning slowly varying features of videos, the objective of the pose encoder is to learn the complementary time-dependent representation to achieve proper frame reconstruction. The pose encoder is trained using a novel mutual information loss term, described in Section 3.6. A standard recurrent LSTM network is trained to predict the pose representation of future frames. We train a decoder network to generate future frames by taking as input the predicted pose and content representations.

In addition to the autoencoder framework described in this section, any generative model with latent variables, equipped with a proper method to infer a latent space representation of observed data can be seen as a form of autoencoder network. In the next section, we turn our attention to an extremely popular deep generative modeling technique known as the variational autoencoders (VAEs). VAEs are directed graphical models with latent variables and are used to explicitly model the underlying data-generating distribution from a set of observed data samples.

1.4.2 Variational Autoencoders

The variational autoencoder (VAE) [37] is a deep generative modeling technique that belongs to the family of directed graphical models. VAEs are a combination of deep neural networks and probabilistic graphical models based on the variational inference technique. Variational inference is used to infer the latent code of observed data samples in a graphical model with an intractable partition function. For example, consider the directed graphical model shown in Figure 1.4 (left), where x is the observed variable and z is the hidden latent variable.

The problem of inference is computing the posterior distribution $p(z|x)$ of z given x which is given by equation below:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{1.1}$$

However, the marginal distribution $p(x) = \int p(x, z)p(z)dz$ is intractable in many cases especially when the observed data is high dimensional. This is one of the main challenges related to the optimization of probabilistic graphical models with latent variables. There are two main methods that can be used to handle the challenge associated with intractable posterior distribution. One of them is the Monte Carlo approximation, and the other one is the variational inference technique.

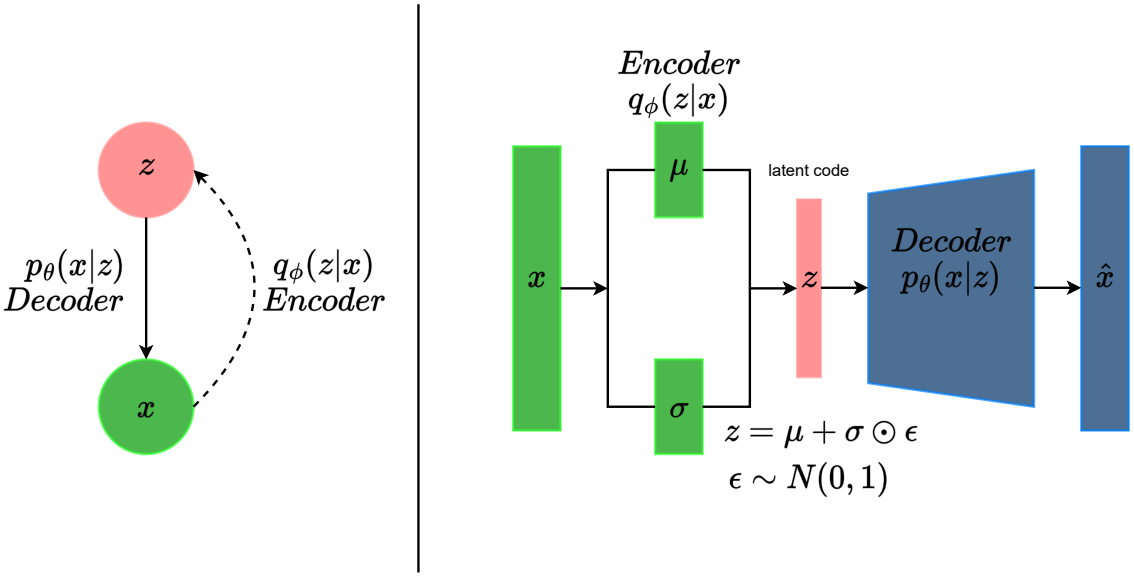


Figure 1.4 Schematic diagram of computational flow in a variational autoencoder [86]: Variational autoencoders are directed graphical models that learn a probabilistic mapping between the observed samples x and a latent variable z. The generative model learns a joint distribution $p_\theta(x, z)$ which is often factorized as $p_\theta(x, z) = p_\theta(z)\ p_\theta(x|z)$. Where $p_\theta(z)$ is the prior distribution over latent space, and $p_\theta(x|z)$ represents a stochastic decoder. The inference model, $q_\phi(z|x)$ in the above figure, is a stochastic encoder that approximates the true but intractable posterior distribution $p_\theta(z|x)$ of the generative model.

1.4.2.1 What is Variational Inference?

As discussed earlier, it is not possible to calculate the posterior distribution $p(z|x)$ because of the intractability of the integral $\int p(x,z)p(z)dz$. In variational inference, the idea is to approximate the posterior distribution $p(z|x)$ by a parametric distribution $q(z)$, such that $q(.)$ belongs to a tractable family of distributions. The objective is to optimize over the parameters of $q(.)$ such that $q(z)$ is close to $p(z|x)$. $q(z)$ can be made to be similar to $p(z|x)$ by minimizing the KL-divergence between these two distributions:

$$\min \mathcal{KL}(q(z)||p(z|x)) = -\sum q(z) \log \frac{p(z|x)}{q(z)} \tag{1.2}$$

By substituting, $p(z|x) = \frac{p(x,z)}{p(x)}$ in place of $p(z|x)$ in the optimisation objective given in Equation 1.2 above we get,

$$\begin{aligned} \mathcal{KL}(q(z)||p(z|x)) &= -\sum q(z) \log \frac{p(x,z)}{q(z)} \times \frac{1}{p(x)} \\ &= \sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} - \log p(x) \Bigg] \\ &= -\sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} \Bigg] + \sum_z q(z) \log p(x) \\ &= -\sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} \Bigg] + \log p(x) \sum_z q(z) \end{aligned} \tag{1.3}$$

$$\log p(x) = \mathcal{KL}(q(z)|p(z|x)) + \sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} \Bigg] \tag{1.4}$$

In variational inference, we deal with maximizing the quantity $q(z)\big[\log \frac{p(x,z)}{q(z)}\big]$ given in Equation 1.5, which is known as the variational lower bound. This quantity is always less than or equal to $\log(p(x))$. In conventional mean field variational inference, $q(.)$ is assumed to have a tractable form such that it can be factorized into a product of marginal distributions over each dimension.

$$\sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} \Bigg] = \log p(x) - \mathcal{KL}(q(z)|p(z|x)) \tag{1.5}$$

$$\begin{aligned} \sum q(z) \Bigg[\log \frac{p(x,z)}{q(z)} \Bigg] &= \sum q(z) \Bigg[\log \frac{p(x|z)p(z)}{q(z)} \Bigg] \\ &= \sum q(z) \Bigg[\log p(x|z) + \log \frac{p(z)}{q(z)} \Bigg] \\ &= \sum \Bigg[q(z) \Bigg[\log p(x|z) \Bigg] + \sum \Bigg[q(z) \log \frac{p(z)}{q(z)} \Bigg] \\ &= \mathbb{E}_{q(z)} \Bigg[\log p(x|z) \Bigg] - \mathcal{KL}(q(z)||p(z)) \end{aligned} \tag{1.6}$$

So, basically to make $q(z)$ similar to $p(z|x)$, one can minimize the $KL(q(z)||p(z|x))$, which is identical to maximizing the variational lower bound given by $\mathbb{E}_{q(z)}\big[\log p(x|z)\big] - \mathcal{KL}(q(z)||p(z))$. This lower bound is maximized when $q(z)$ is similar to $p(z)$ is the conditional likelihood of the data given latent variables, $\mathbb{E}_{q(z)}\big[\log p(x|z)\big]$ is maximized (z should generate the observations correctly).

1.4.2.2 Realising a Probabilistic Graphical Model as a Neural Net

Let us assume that the probability distribution $q(z|x)$ is a neural network with variational parameters ϕ and similarly the distribution $p(x|z)$ is another neural network that takes as input samples from the distribution $q(z|x)$ and outputs x Figure 1.4. The objective function to optimise the parameters of the probabilistic encoder and decoder networks is given by the evidence lower bound which is, $\mathbb{E}_{q(z)}\big[\log p(x|z)\big] - \mathcal{KL}(q(z)||p(z))$. To optimize this objective function, we need to choose a distribution $p(z)$ in order to make $q(z)$ similar to $p(z)$, for example, if we choose the distribution of $p(z)$ as a gaussian then the samples from the probabilistic encoder follow a gaussian distribution. $\mathbb{E}_{q(z)}\big[\log p(x|z)\big]$ part of the objective function is conceptually reconstruction error. This probabilistic encoder-decoder network can be trained end-to-end using mini-batch stochastic gradient methods using a simple representation trick shown in Figure 1.4.

1.4.3 Generative Adversarial Networks

Generative Adversarial Nets [24] belong to the class of implicit deep generative modeling techniques. GANs learn to generate synthetic data samples given a set of training data points from an unknown data-generating distribution. The general architecture of a generative adversarial network consists of two deep artificial neural networks- a generator, and a discriminator. These two networks act as adversaries to each other in order to generate samples similar to the ones presented in the training dataset.

The generator G and discriminator D networks are trained simultaneously in an adversarial setting. The generative network tends to capture the data-generating distribution, and the discriminator model captures the probability of whether the generated samples belong to the training data. The training objective is a minimax game; where the generator's objective is to maximize the probability of the discriminator making a mistake, whereas the discriminator's objective is to classify fake samples from real ones correctly. Formally, the discriminator D and the generator G networks are involved in a two-player minimax game associated with the value function $V(G, D)$ described below in Equation 1.7. $p_z(z)$ is the distribution from which samples of random noise are drawn and given as input to the differentiable generative network.

$$\min_{\mathbf{G}} \max_{\mathbf{D}} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{1.7}$$

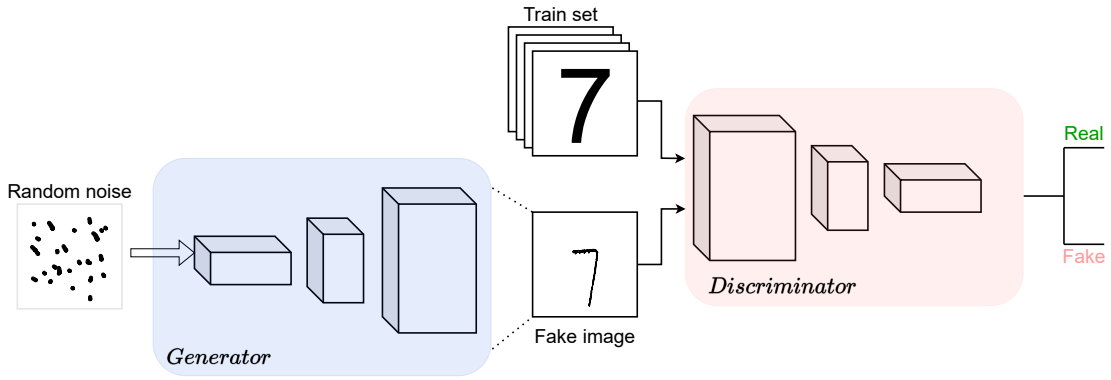


Figure 1.5 Block diagram of generative adversarial nets [23]: The generator neural network takes samples of random noise from a known distribution to generate fake samples that resemble data samples present in the dataset. The learned discriminator networks classify fake samples from real ones. The objective of the generator is to create synthetic samples to confuse the discriminator between fake samples from real ones. The generator and discriminator networks are involved in a minimax game.

1.5 Mutual Information Estimation

Mutual information is a fundamental quantity in the field of information theory that is used to quantify the dependence between two random variables (RVs). Suppose that X and Y are two RVs then the mutual information between them is defined in the equation given below:

$$I(X;Y) = \int_{\mathcal{X}\times\mathcal{Y}} log \frac{d\mathbb{P}_{XY}}{d\mathbb{P}_X \otimes \mathbb{P}_Y} d\mathbb{P}_{X,Y}$$

In the above-given equation, $\mathbb{P}_{XY}$ is the joint probability distribution between the two RVs, X and Y, with the marginal distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$, respectively. The mutual information between any two random variables varies from 0 to $+\infty$. A high value of $I(X;Y)$, implies that the RVs X and Y share a considerable amount of information and exhibit a high degree of dependency and vice-versa. Whereas it is equal to zero *iff* X and Y are mutually independent.

In the recent past, large-scale developments in computational capabilities and increase in the size of data sets have enabled reliable sample-based estimation of MI using mini-batch stochastic optimization techniques [59, 5, 62, 73, 21]. Sample-based estimation of MI refers to the problem of estimating mutual information from samples of two random variables with unknown joint and marginal distributions. The recent development of various techniques that can reliably estimate MI from data, this simple measure of dependence has found numerous applications in generative modeling [10], representation learning [9, 35, 2, 29], information bottleneck [76, 77, 1], and predictive modeling [41].

Classical approaches of estimating mutual information from samples of two random variables use non-parametric learning algorithms like binning [19], K-Nearest Neighbour based entropy estimation

[38], and kernel density estimation [54]. These methods which are based on traditional machine learning algorithms are computationally expensive and do not conform to mini-batch based optimization strategies, producing unreliable estimates of MI especially when the data dimensionality is high. To overcome the challenges associated with using non-parametric learning algorithms for reliable computation of MI, recent estimation methods [5, 62, 73] train artificial neural networks (ANNs) by using variational lower bounds of MI [56, 15]. In these more recent works, a parametric neural network known as critic is trained to approximate the likelihood density ratio between the joint and product of marginal distributions, $d\mathbb{P}_{XY}/d\mathbb{P}_X \otimes \mathbb{P}_Y$, Equation. 3.1. These approximated density ratios are used to estimate different variational lower bounds of MI. These deep learning-based estimation methods are reliable when the true MI is low, they tend to produce high variance estimates when the true MI is high as updating the critic's parameters by using gradients from the variational lower-bound formulation is unstable [61]. Hence, in our MIPAE framework, we use the GAN objective function to train the critic network in order to minimize the mutual information between pose latent variable across time, shown in Figure 3.1(left).

1.6 Contributions of this thesis

- This thesis proposes a simple yet effective predictive autoencoder framework that learns to disentangle video frame representation in an unsupervised setting. The video frame representation space is factorized into a temporally consistent content and another temporally varying, pose latent variable. We introduce a novel mutual information loss term to train the pose encoder in order to attain proper content/ pose factorization (Figure 3.1). The disentangled video representation is used for the downstream task of predicting future frames of a video sequence. We empirically validate the efficacy of the proposed MIPAE framework at learning factorized frame representation which significantly improves the quality of predicted frames. Further, factorizing the latent space representation simplifies the task of directly predicting high-dimensional video frames to that of predicting the low-dimensional pose latent vector of future frames.

- We also present a metric based on the mutual information gap [9] to quantitatively measure the degree of disentanglement between the factorized latent variables - pose and content (i.e., we quantitatively show that the learned pose and content latent factors are independent of each other). This score has been used to draw comparisons between MIPAE with other similar disentanglement-based video prediction approaches. We experimentally demonstrate that proper pose/ content factorization as measured by the MIG scores corroborates with the improvement in the quality of predicted frames and also leads to stable long-range predictions.

- Apart from the two contributions mentioned above, in Chapter 4 we present the effect of applying a novel cyclic consistency loss term to the optimization objective of two different stochastic video prediction models. In this work, it is experimentally shown that a simple architectural change of

using a content encoder leads to proper pose/ content factorization of the frame representation space learned by stochastic video prediction models.

Chapter 2

Related Work

The goal of visual prediction is to learn without explicit labels, a representation that generalizes to a range of previously unseen tasks, such as semantic labeling of various objects present in video sequences, activity classification, and video frame prediction. The task of video prediction is a well-known and long-standing problem of predicting multiple future frames by conditioning on a given set of context frames that belong to a video sequence. In this chapter, we discuss the current state of research work in video prediction and the motivation for research works presented in this thesis.

2.1 Deterministic and Transformation based Methods

The task of predicting future frames of a video sequence has lately received a lot of interest from the computer vision research community. A range of deep video prediction algorithms [65, 74, 81, 47], that assume the task of video frame prediction to be deterministic have been proposed in the literature. These deterministic video prediction models consider that there exists only one plausible future prediction. Several methods [11][58] exploit the deterministic nature of video game environments and predict future frames by conditioning on a set of action labels that are known a priori. However, models with such naive assumptions produce blurry frame predictions due to their inability to account for the stochasticity exhibited in real-world videos. The stochasticity in real-world videos can arise from a variety of different situations like occluded objects entering and exiting video frames, uncertainty in motion of various objects in the scene, as well as, from complex object interactions.

Another area of relevant research work can be represented by the transformation-based video prediction. These methods primarily focus on modeling motion rather than on reconstructing appearance. Recent works, [17, 82] predict subsequent future frames by transforming the last observed frame through a constrained geometric transformation. Finn-Goodfellow et al.[17] modeled motion by exploiting the temporal consistency in video sequences and predicted transformation kernels for masked groups of objects in an action-conditioned Conv-LSTM framework. Vondrick-Torrabla et al. [82] generated a transformation for each pixel by using an adversarial loss to constrain the set of plausible transformations that map context frames to future frames. However, learning the non-linear transformation between consecu-

tive frames of a video sequence that exhibit real-world stochastic dynamics is exceptionally challenging due to the high dimensional nature of video frames. In contrast to these transformation-based methods that directly model predictions in the image-pixel space, we propose a MIPAE framework that learns a disentangled video representation for the downstream task of video prediction. Our approach uses a standard LSTM architecture to learn a predictive model of the low dimensional pose latent variables. The predicted pose and content representations are decoded to generate high-dimensional future frames. Training a simple LSTM predictive network on the low dimensional pose latent variables is straightforward compared to modeling non-linear transformations in the high dimensional pixel space. It is also important to note that our approach to learning disentangled representations is entirely unsupervised, and the prediction model does not depend on action labels.

2.2 Disentangling Latent Space Representation

Learning disentangled representation of data that explicitly captures the salient attributes of each sample instance is a well-explored area of research in unsupervised learning [7, 39]. Information Maximizing Generative Adversarial Nets, InfoGAN [10] arguments adversarial training with a mutual information loss term to learn meaningful and interpretable data representations. In this approach, mutual information is maximized between observations and the GAN's noise latent variables to disentangle digit shapes from writing styles on the MNIST dataset, lightening of 3D synthetic images from their pose, and foreground from background digits on the SVHN dataset. Recent research has also used different variations of evidence lower bound (ELBO), proposed by Kingama [37] to force disentanglement in the latent space representation. β-VAE [28] proposed an approach based on setting a high value of the KL-divergence multiplicative factor in the evidence lower bound objective function to limit the capacity of latent information channel, thus enforcing a highly factorized latent space representation. Total correlation-based optimization objective also leads to disentanglement in β-TCVAE [9] and FactorVAE [35].

locatello et al.[45] theoretically explained that it is impossible to learn disentangled representation without exploiting a known structure of the latent generative factors of data. Some recent works that have exploited known structural relationships between the latent generative factors of data to learn disentangled representation are [42, 50, 33, 14]. In the proposed MIPAE framework (chapter 3), the factorization of video representation is based on the assumption that there exists a differentiating temporal structure between the content and pose latent factors of videos, that is, the time independence of content and time dependence of pose latent variables. Similar to our MIPAE method, [79, 78, 14, 32] also learn to factorize the latent space representation for video frame prediction.

MCNet [79] disentangled video into content and motion by explicitly modeling the flow of image difference and used the content encoding of a single frame to make future frame predictions. Similarly, Disentangled-Representation Net (DRNET) [14] factorized video representation into content and pose by applying an adversarial loss on the pose factors, preventing them from being similar from one video

to another, thus, ensuring that pose latent factors do not contain content information. Decompositional Disentangled Predictive Auto-Encoder (DDPAE) [32] is a framework that combines structured probabilistic models with deep networks to automatically decompose high-dimensional videos into components and disentangle each component to have low-dimensional temporal dynamics that are easier to predict. In contrast to these methods, we learn to factorize video into content and pose representation by adopting the DRNET video prediction architecture and penalizing the pose latent across time with a novel mutual information loss term.

2.3 Sample-based estimation of Mutual Information

Mutual Information (MI) features prominently in ICA literature [67]. However, these methods do not provide a general method for computationally estimating mutual information from samples of two random variables. [38], [70] proposed a non-parametric method to estimate mutual information between two random variables using the k-nearest neighbor approach. But these non-parametric methods do not scale well high dimensionality of data samples.

Recent works have used the variational estimation technique coupled with deep neural networks for tractable estimation of MI. Nguyen et al. [56] proposed a variational lower bound to estimate f-divergences between two probability distributions. F-GAN [57] utilized this formulation to estimate some common instances of f-divergence, like the KL-divergence and the Jensen-Shannon divergence, given samples from two different distributions. MINE [6] used Donsker-Vardhan [15] dual formulation for estimating MI which can be expressed as KL-divergence between the joint distribution and the product of marginal distributions over a couple of random variables. Methods based on variational lower bounds use a deep neural network known as a critic to estimate MI. However, updating the critic using gradients of the F-GAN formulation is unstable. Hence, [61] proposed to use GAN based objective instead of using gradients from F-GAN to train the critic. In our MIPAE framework, we use the GAN-based objective in order to train the critic network. For a comprehensive overview of variational bounds of MI, readers are requested to refer to [61].

2.4 Stochastic Video Prediction Models

As discussed earlier, deterministic video prediction models assume that the future frames are a deterministic function of the past frames and tend to predict an average of the possible futures resulting in blurry frame predictions. However, some research works take into account the inherent uncertainty in predicting future video frames and can be placed under two major categories: (i) stochastic latent variable models [3, 26], and (ii) adversarially trained generative models [24]. While stochastic latent variable models aim at capturing the predictive distribution of uncertainty in dynamics of real world videos, the aim of GAN based algorithms is the synthesis of realistic frames. The latent variable models presented in [3, 26] are VAE based approaches that disentangle video representation into determinis-

tic and stochastic components. Although standard latent variable models are expressive in terms of producing diverse predictions, these models struggle to generate natural looking future frames [13].

The other set of stochastic video prediction models explore the GAN-based formalism [24] to account for the uncertainty in real-world videos. While these models tend to produce sharp and natural looking future frames, they are susceptible to mode collapse and training instability, particularly in conditional video generation settings [91]. Stochastic Adversarial Video Prediction (SAVP) [40] combines adversarial training with latent variables to enable high-quality stochastic video prediction. In our second work, we learn to disentangle video representation into two components- stochastic pose and deterministic content components in two different adopt two different stochastic video prediction models, namely, SAVP and SVG-LP. In this work, we condition the generator of these stochastic video prediction models with a content encoder and employ a novel cyclic consistency loss term to their optimization objective in order to achieve proper factorization of the latent space.

Another critical probabilistic approach to video frame prediction explores the autoregressive technique to model the full joint distribution of pixel intensity values. In video pixel networks [34] [71] frames are generated by sampling pixels one by one in a raster scan order. Although these methods produce crisp future frames, training and inference are computationally expensive. Even highly parallelized approaches have been ruled out as impractical for high-resolution videos [66].

2.5 Summary

This chapter presents a birds-eye view of the different paradigms within the space of video prediction models. We categorize the current state of research into three different categories. The first category is comprised of the mean squared error-based video prediction models that use simple sequence to sequence modeling techniques to predict future frames [65, 74, 81]. In this category, we also place transformation-based methods that primarily focus on modeling motion in video sequences and predict a future frame by transforming the previous frame through a constrained geometric distortion [17, 82].

The second category of work is central to our work presented in Chapter 3 is the representation learning approach, where the video representation space is factorized into temporally consistent and varying components by exploiting the structure in the latent generative factors of a video [14, 32]. In our MIPAE framework, we build upon the disentangled representation learning-based video prediction framework presented in work Denton et al. [14].

Lastly, we discuss the stochastic video prediction models. These video prediction models take a probabilistic approach to modelling the predictive distribution of future frames. They uses stochastic latent variable-based generative modeling techniques to account for the inherent uncertainty in predicting future frames by conditions on a small set of context frames. In Chapter 4, we exhibit the advantages of using a novel cycle consistency loss term in stochastic video prediction models. We also present a simple technique to decompose the video representation space into time-dependent and time-independent features in two different stochastic video prediction models.

Chapter 3

Representation Learning for Video Frame Prediction

Deep learning is a subset of machine learning algorithms consisting of neural networks, hierarchical probabilistic models, and various unsupervised and supervised feature learning algorithms. Deep learning models have been extremely successful in solving complex computer vision problems such as object detection, 3D pose estimation, scene reconstruction, video tracking, object recognition, and image restoration. The ability to extract useful task-dependent representation from large amounts of unlabelled data is one of the most critical factors contributing to the large-scale prominence of deep learning models in the development of computer vision applications.

Disentangled representation learning algorithms aim to factorize the learned representation space of the observed data into latent generative factors. The disentangled representations can subsequently be used to develop an understanding of the low-dimensional manifold on which the data exists. In general, disentangled representation learning improves the robustness, explainability, and generalization capability of deep learning models. Subsequently, the learned factors of data generation can be used to develop computer vision models for a wide range of tasks, from object recognition to synthetic image generation.

Recent works such as β-VAE [27] disentangle the representation of data into latent generative factors by introducing an adjustable hyper-parameter β in the VAE [37] objective. In this approach, they increase the weight of the KL-divergence term in ELBO to put extra constraints on the implicit capacity of the latent code to learn the underlying generative factors. Similarly, FactorVAE [35] and β-TCVAE [9] approach penalize the total correlation between latent variables for learning disentangled representations. While, InfoGAN [10] modifies the GAN [24] objective by maximizing the mutual information between a subset of the GAN noise variables and the observations to learn interpretable and meaningful representations of data. However, in their work, Locatello and Bauer [46] show that it is impossible to learn disentangled representations from unlabelled data without implicit supervision or exploiting inductive biases.

This chapter presents an auto-encoder network known as Mutual information-based predictive auto-encoder, MIPAE, used to disentangle the video representation space into a set of mutually independent generative factors. These low-dimensional learned representations are then used to generate future

frames of video sequences. Disentangled representation learning approaches have been heavily explored for future frame prediction due to the high dimensionality of video data [79, 78, 14, 32].

In our approach, we exploit the temporal consistency in the videos to factorize the representation of video frames into time-independent content and time-dependent pose latent variables. The proposed MIPAE auto-encoder architecture comprises a content encoder, pose encoder, decoder, and a standard LSTM network. The content encoder in MIPAE is encouraged to learn time-independent representation using a slow feature analysis constraint. This enforces the content representation of frames belonging to the same video sequence to be similar. However, constraining the content encoder is not sufficient to ascertain mutual independence between learned pose and content latent factors. To achieve proper disentanglement, we penalize the pose encodings of different frames from the same video using a novel mutual information loss term. The mutual information loss term encourages the pose representation to learn only time-dependent information, as it is assumed that most of the mutual information across frames is time-independent. Mutual information between pose representation of different frames is estimated using the I_{JS} variational lower bound of MI.

3.1 Problem Statement

Our task is to generate the next T frames of a video sequence, $\hat{x}_{C+1:C+T} = (\hat{x}_{C+1}, \hat{x}_{C+2} \dots \hat{x}_{C+T})$ by conditioning on a sub-sequence of C context frames, $x_{1:C} = (x_1, x_2, \dots x_C)$.

3.2 Our Approach

Learning the non-linear transformation that maps a sub-sequence of known frames, $x_{1:C}$ to a sequence of future frames, $\hat{x}_{C+1:C+T}$ is challenging due to the high dimensional nature of video frames. The proposed MIPAE framework learns to factorize the video representation space into low-dimensional generative factors, which are easy to predict. The factorized video representation space consists of (i) time-independent content and (ii) time-dependent pose factors. To generate future frames, we predict the low-dimensional pose representations of future frames. The predicted pose and learned content representations are then decoded to obtain future frame predictions.

3.2.1 Assumptions behind our approach

We make some simplifying assumptions to formulate our disentangled representation learning approach. To learn proper pose content disentanglement, we assume that frames belonging to the same video sequence have similar content representations, i.e., the constituent objects in the frames remain fixed throughout the sequence (visual appearance and number remain the same). However, these objects can exhibit a complex range of motions by varying their position with time. Based on this assumption,

we consider that the true latent generative factors of a video consist of two major components, a time-independent content factor, z^c another time-dependent pose vector, ${z^p}_{1:C+T}$.

For predicting future frames, we assume that videos in the real world are generated using a two-stage generative process in which (1) the latent generative factors, content, z^c, and pose, $z^p_{1:C+T}$ are sampled from the true joint distribution, $p(z^c, z^p_{1:C+T})$; (2) and once these factors are obtained, the video is generated by sampling from the true conditional distribution, $p(x_{1:C+T}|z^c, z^p_{1:C+T})$.

3.2.2 Methodology

Locatello-Bauer et al. [45] explains that learning disentangled data representations in an unsupervised setting is impossible unless the approach exploits some inductive biases or provides implicit supervision. In our approach, we build on this crucial understanding and learn disentangled video fame representations by exploiting the temporal structure in the latent generative factors of videos: time-independence of content and time-dependence of pose latent factors.

The proposed mutual information predictive autoencoder, MIPAE, framework consists of two independent parts: an auto-encoder network and a standard LSTM network. The autoencoder network has a content encoder, E_c; a pose encoder, E_p; and a frame decoder, D. The autoencoder network is trained to factorize video frame representation into a set of mutually exclusive factors: the time-independent factor known as content, z^c, and the time-dependent factor known as pose, z^p_i. Given a video sequence $X = \{x_i\}_{i=1}^{T+C}$ of $T + C$ frames, the content encoder learns a single content representation z^c, such that $z^c = E_c(x_i)\ \forall i \in 1..T + C$. On the other hand, the pose encoder learns the time-dependent pose representation for each video frame, such that $z^p_i = E_p(x_i)$. A decoder network reconstructs the frame from the pose and content representation $\hat{x}_i = D(z^c_i, z^p_j)$.

The standard LSTM network, L, is independently trained to predict the pose representation of future frames. The predicted pose representations and time-independent content representations are used to generate future frames.

3.3 Training Procedure

In this section, we explain our two-stage training procedure. First, we train the autoencoder network in our MIPAE architecture to learn disentangled frame representation. In the second stage of our training procedure, we train the LSTM network to predict the pose representations of future frames. Our framework decodes the predicted pose and content representation to generate future frames.

3.3.1 Auto-encoder Training

In the first stage of our training procedure, we train the autoencoder network to factorize the representation of video frame into time-independent content factor, z^c, and time-dependent pose factor, z^p_i.

As explained above, given a video sequence of $T + C$ frames, the content encoder learns a single time-independent representation, $z^c = E_c(x_i)\ \forall i \in 1..T + C$. In our framework, the content encoder E_c is constrained to extract slowly moving features from videos by penalizing the similarity loss between the content representation of frames belonging to the same video sequence. The similarity loss is the MSE loss between the content encoding of two frames x_t and x_{t+k} belonging to the sequence of frames $\{x_i\}_{i=1}^{T+C}$ and separated by an offset of k time-steps.

The MSE loss in the equation below is used to train the content encoder. MSE loss encourages the content encoder to learn a time-invariant representation which is the same for all frames belonging to a video sequence.

$$\mathcal{L}_{sim} = \mathbb{E}_{P(x_t, x_{t+k})} \left[\left\| E_c\left(x_t\right), E_c\left(x_{t+k}\right) \right\|_2^2 \right] \tag{3.1}$$

However, the above-given constraint is insufficient for the proper pose-content disentanglement as the pose latent representation of frames can still encode some part of the time-invariant content representation. Our work hypothesizes that any content information in the pose representation of video frames can be modeled as the mutual information between the pose representations. Consequently, we penalize the pose representations of video frames across time using a novel mutual information loss term.

To estimate the mutual information, MI $I(z_t^p, z_{t+k}^p$ between pose encoding z_t^p and z_{t+k}^p, where $t, t + k \in 1..n$, we train a critic neural network C to discriminate between pose representations of two frames belonging to the same video sequence and different video sequences. The critic learns to approximate the log-likelihood ratio $\log(P(z_t^p, z_{t+k}^p)/P(z_t^p)P(z_{t+k}^p)$. The approximate log-likelihood ratio learned by the critic is plugged back into I_{NWJ} [61] lower bound to estimate the mutual information between pose representations. We use the variational lower bound of mutual information, I_{JS} proposed by Poole et al. [61] to estimate mutual information between the pose latent variables. I_{JS} exhibits lower variance than methods that use the monte-carlo technique to estimate MI [35].

Mutual information between pose latent representations is estimated using the equation given below:

$$\mathcal{L}_{MI} = \mathbb{E}_{P\left(z_i^p, z_{i+k}^p\right)} \left[C\left(z_i^p, z_{i+k}^p\right) \right] - \mathbb{E}_{P\left(z_j^p\right) P\left(z_{j+k}^p\right)} \left[\exp\left(C\left(z_j^p, z_{j+k}^p\right) \right) \right] + 1 \tag{3.2}$$

For the proper estimation of the mutual information between z_t^p and z_{t+k}^p, samples from the joint distribution, $(z_{p,i}^t, z_{p,i}^{t+k})$ and samples from the marginal distribution, $(z_{p,i}^t, z_{p,j}^{t+k})$ are required. Samples from the joint distribution, $(z_{p,i}^t, z_{p,i}^{t+k})$ are acquired by feeding frames from the same video sequence through the pose encoder. In contrast, samples of the marginal distribution $(z_{p,i}^t, z_{p,j}^{t+k})$ are acquired by feeding frames from different video sequences. i and j denote pose representations of frames from two different video sequences.

The critic, C, learns to approximate the log-likelihood ratio by minimizing the cross-entropy loss, which is given in the equation below:

$$\mathcal{L}_C = -\mathbb{E}_{P\left(z_i^p, z_{i+k}^p\right)} \left[\log\left(\sigma\left(C\left(z_i^p, z_{i+k}^p\right) \right) \right) \right] - \mathbb{E}_{P\left(z_i^p\right) P\left(z_{i+k}^p\right)} \left[\log\left(1 - \sigma\left(C\left(z_i^p, z_{i+k}^p\right) \right) \right) \right] \tag{3.3}$$

Where σ is the sigmoid function. The reconstruction error is:

$$\mathcal{L}_{recon} = \mathbb{E}_{P(x_i)} \left[\| D\left(E_c\left(x_i\right), E_p\left(x_i\right)\right) - x_i \|_2^2 \right] \tag{3.4}$$

Training Methodology: We use a two-step training procedure to train the pose and content encoders properly. In the first step, the critic C is trained to minimize the cross-entropy loss, and in the second step, we train E_c, E_p and D to minimize the combined training loss given in Equation 3.6.

Training object for the critic C is given by:

$$\min_{C} \mathcal{L}_C \tag{3.5}$$

The overall training objective for the content encoder E_c, pose encoder E_p and decoder D:

$$\min_{E_c, E_p, D} \mathcal{L}_{recon} + \alpha \mathcal{L}_{sim} + \beta \mathcal{L}_{MI} \tag{3.6}$$

Hyper-parameters α and β control the importance of similarity loss and MI loss, respectively.

In Section 3.5, we demonstrate the efficacy of our hypothesis that minimizing mutual information loss is crucial for proper factorization of the video representation space. This is effective because the pose encoder, E_p, is shown in Figure 3.1 is restricted from encoding any information that is not mutually exclusive between two frames of the same video sequence, which can be considered to be the time-independent content representation of the entire sequence. In the next section, we discuss the training procedure for the LSTM network. After training the auto-encoder network, a standard LSTM network is trained to predict the future frames' low-dimensional pose latent representations

3.3.2 LSTM training procedure

The LSTM network L is trained independently to predict the pose representation of future frames. To predict a future frame $\hat{x}_t$, the LSTM L network is used to first predict the low dimensional pose representation $\hat{z}_t^p$ by taking as input the pose $\tilde{z}_{t-1}^p$ and z_C^c which is the content representation of the last known frame x_C. The pose latent representation of the previous frame, $\tilde{z}_{t-1}^p$ which is taken as input to predict the pose representation of frame at time t is acquired by passing frame x_{t-1} through the pose encoder E_p if $t-1 \in [1, C]$ (i.e. if the $t-1$th frame is a context frame). Else, $\tilde{z}_{t-1}^p$ is the pose representation predicted by the LSTM for the time frame $t-1$, denoted by $\hat{z}_{t-1}^p$ in Equation 3.7.

$$\hat{z}_t^p = L(z_C^c, \tilde{z}_{t-1}^p) \; where \; \tilde{z}_t^p = \begin{cases} E_p(x_t) & t < C+1 \\ L(z_C^c, \tilde{z}_{t-1}^p) & t \geq C+1 \end{cases} \tag{3.7}$$

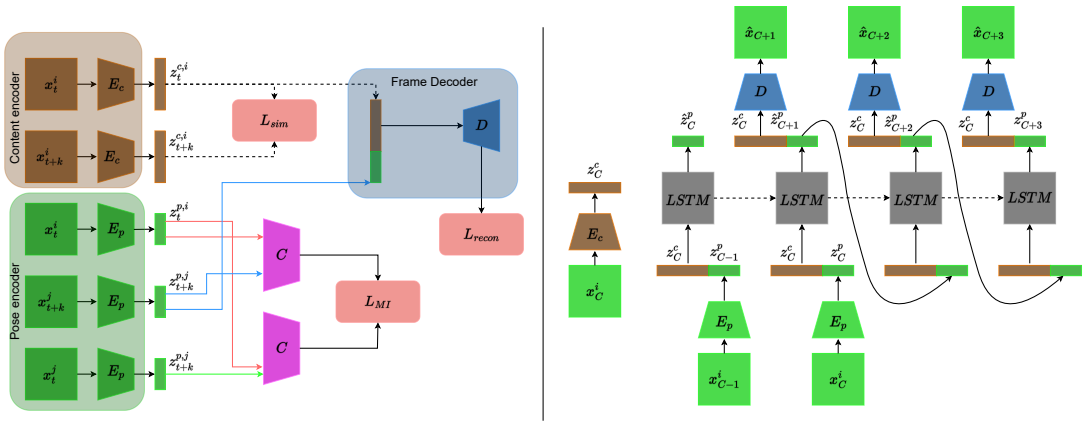


Figure 3.1 MIPAE framework: Left: Two-step training procedure for content encoder E_c, pose encoder, E_p and decoder, D along with training objectives. To calculate the mutual information between pose latent variables, pose latent variables $z_t^{p,i}$, $z_{t+k}^{p,i}$ and $z_{t+k}^{p,j}$ are taken from videos $x_{1:C+T}^i$ and $x_{1:C+T}^j$ by using the pose encoder E_p, where i and j denote that they belong to two distinct video sequences. The joint samples $(z_t^{p,i}, z_{t+k}^{p,i})$ and the marginal samples $(z_t^{p,i}, z_{t+k}^{p,j})$ are given as input to the critic C. The critic's output is used to estimate MI using Equation 3.2. Right: Recurrent generation of pose latent variables $\hat{z}_t^p$ using LSTM network. These predicted pose vectors are used to generate future frames by decoder D.

3.4 MIG metric to evaluate the degree of disentanglement

There are several popular methods to evaluate the degree of disentanglement in data representation, one of which is the latent code traversal. In this method, one dimension of the latent space representation is varied while keeping the other dimensions constant, and the corresponding visual changes are observed in an image. While the latent traversal method is useful in qualitatively depicting the effectiveness of a method's ability to disentangle the data representation into generative factors. Latent code traversal fails to provide a quantitative measure for the degree of disentanglement.

Concurrent methods [28, 9, 35] propose various metrics to quantify the effectiveness of a method in learning disentangled representation. These methods are useful for datasets for which the factors sample generation are known as apriori (i.e., synthetically generated datasets like Dsprites, MPI3D Real in our case). For example, [14, 28] trained a classifier to predict the factors of data generation by taking as input learned latent factors. In their work, the classifier's prediction accuracy was considered an indicator of the degree of disentanglement. However, it has been pointed out correctly by [9] that these methods can not be used when the learned latent representation is not axially aligned with the true factors of data generation. Moreover, their performance depended heavily on the model architecture of the classifier. H. Kim et al. [35] used a majority vote classifier to negate the dependence of classification-based evaluation metrics on model hyperparameters and deal with the case where latent factors and generative factors are not axially aligned. However, these evaluation metrics have not been used extensively due to shortcomings.

T. Q. Chen et al. [9] proposed a generic mutual information-based metric in their work. In their formulation, mutual information scores are calculated as the true factors and learned representation between each pair of dimensions. The calculated MIG score indicates the degree of disentanglement achieved by the representation learning technique. MIG scores can only be calculated for datasets with known factors of data generation. In this work, we present an adaptation of this mutual information gap (MIG) metric, which can be used to assess the degree of disentanglement in video representation quantitatively. We calculate the mutual information between the generative factors and the learned pose, content representation, instead of independently calculating them between each pair of dimensions. The proposed MIG score has been used extensively to evaluate the efficacy of the proposed MIPAE disentanglement framework. Any reference to the mutual information gap (MIG) score refers to this modified version which is given in the equation below:

$$MIG = \frac{0.5}{H(f^c)}\Big(I(f^c, z^c) - I(f^c, z^p)\Big) + \frac{0.5}{H(f^p)}\Big(I(f^p, z^p) - I(f^p, z^c)\Big) \tag{3.8}$$

3.5 Experiments and Analysis

We evaluate our MIPAE video prediction framework on multiple publicly available and widely used datasets in video prediction. In this work, we experimentally demonstrate the efficacy of our approach

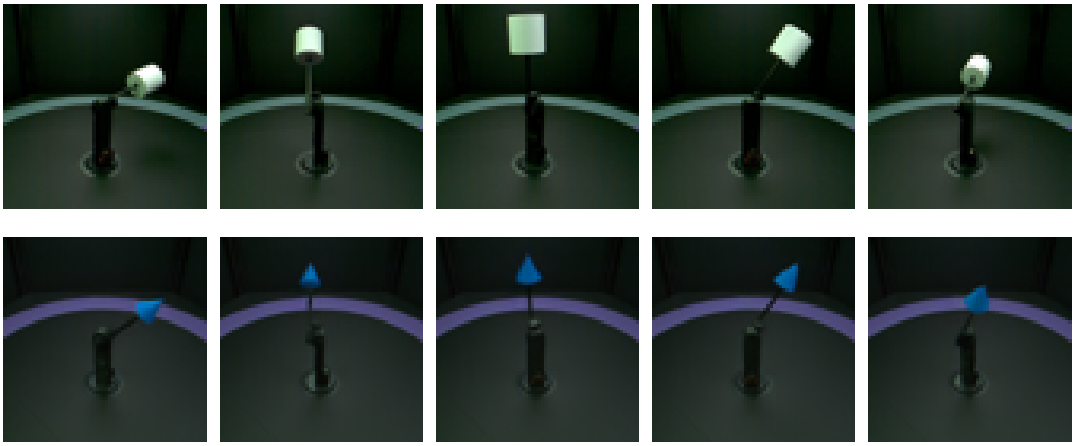


Figure 3.2 Pose, content disentanglement by MIPAE: Pose latent representation of the video sequence shown in the top row is combined with the content latent representation of the blue cone-like object to generate the video sequence below. It can be seen from the above diagram that object in the bottom generated sequence closely follows the pose of the object from the target sequence.

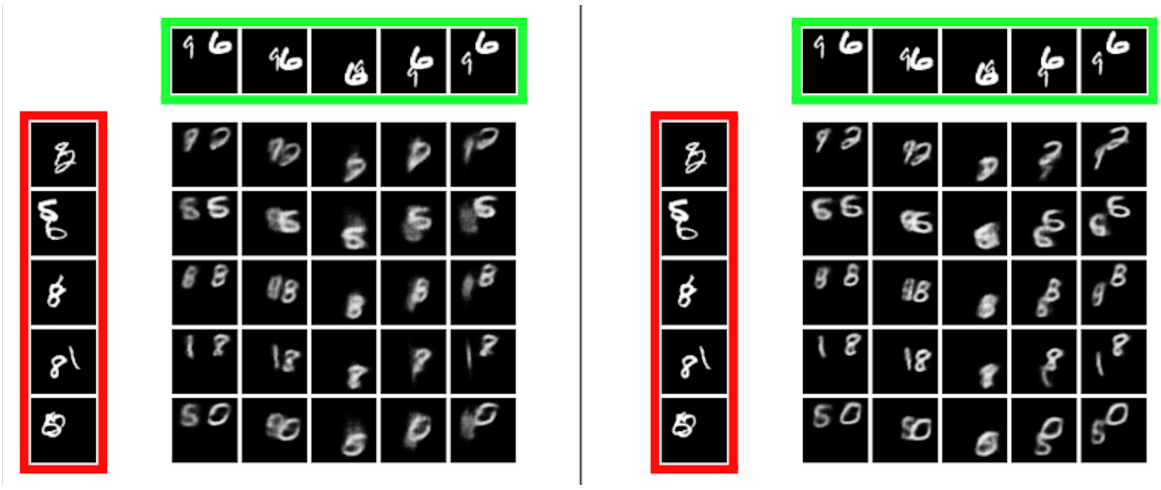
on both synthetic, moving MNIST, Dspites [51] datasets, and real-world moving MPI3D-Real [22] datasets. To measure the perceived visual quality of predicted frames, we calculated the LPIPS distance [90] between predicted and ground truth frames.

Concurrent works have also used PSNR and SSIM scores to evaluate the visual quality of predicted frames. PSNR and SSIM scores show poor correspondence with the true perceptual quality of the predicted frames [40]. However, for a comprehensive evaluation of our MIPAE framework, we calculate PSNR and SSIM scores for completeness. We use the proposed MIG scores, presented in Section 3.4 to evaluate the degree of disentanglement of the learned latent video representation by MIPAE and DRNET on the DSprites and MPI3D-Real dataset. MIPAE achieves better disentanglement than DRNET, which corresponds to higher MIG scores. Our experimental analysis shows that higher MIG scores corroborate a better quality of predicted frames. This proves the hypothesis that a better factorization of the latent representation space leads to higher quality of predicted frames over long horizons.

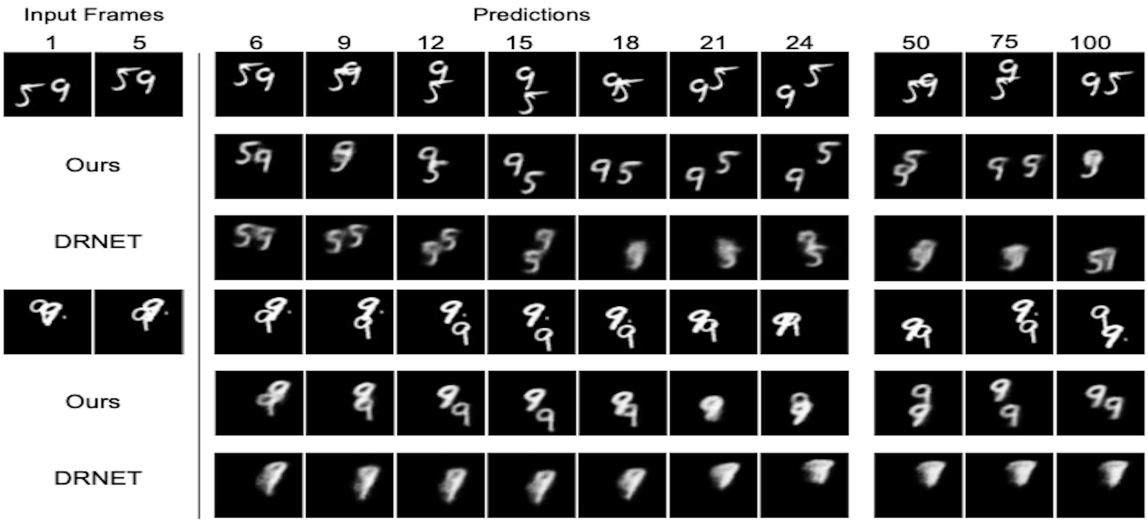
3.5.1 Synthetic MNIST

Moving MNIST dataset has been widely used by previous works to substantiate their disentanglement claims [14, 32]. It consists of video sequences with two MNIST digits bouncing independently in a frame of size 64x64. The MNIST digits move independently with a constant velocity and undergo mirror reflection upon collision with the frame boundaries.

The top left and top right parts of Figure 3.3 demonstrate the pose content disentanglement by DRNET and MIPAE, respectively. In this experiment, we swap the pose and content representation of two sequences to qualitatively access the factorization achieved by our model and the subsequent impact



(a) Disentanglement result on MNIST



(b) Long range frame prediction

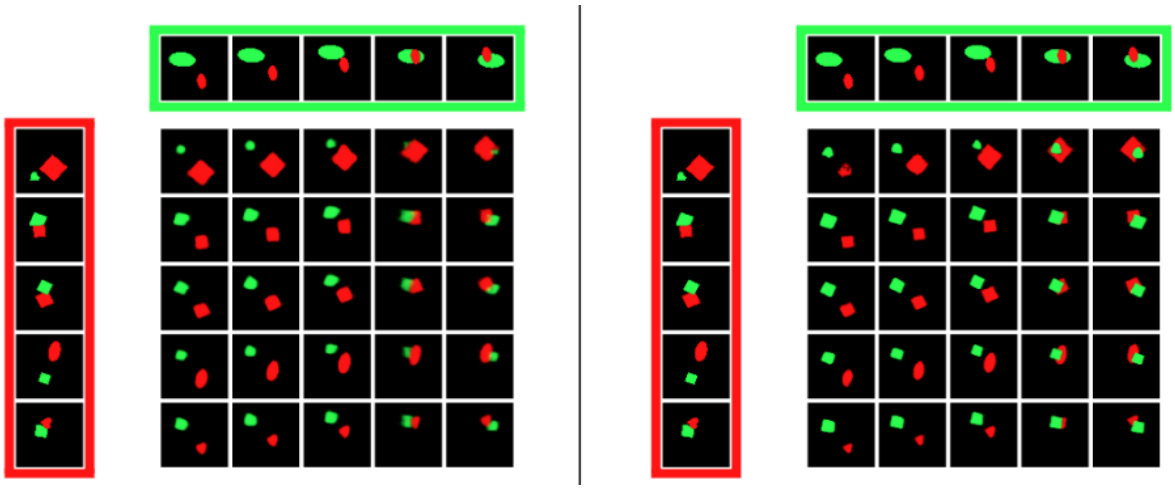
Figure 3.3 Qualitative comparison on moving MNIST dataset: (a) Demonstration of the pose-content disentanglement by DRNET and our MIPAE framework. Each image in the grid is generated by taking the pose latent variable from the sequence on the top, highlighted in green, and the latent content representation of images on the margins, highlighted in red. Our model generates sharp frames (right) in contrast to blurry predictions by DRNET (left) in frames involving complex interactions between MNIST digits due to better content/pose disentanglement. (B) Future frame prediction by our model and DRNET on two sequences. Our model produces future frames that follow the pose representation of ground truth frames over longer horizons in both sequences.

on the quality of predicted frames. In Figure 3.3(a), a new sequence of video frames is generated by combining the pose latent representation $z^p_{1:C+T}$ from the top-most sequence highlighted in green with the content representation of z^c from the images highlighted in red. Digits in the generated sequences closely follow the pose of digits in the source sequence for many different digit representations, indicating that the learned pose representation is independent of the content representations. DRNET requires additional information to learn a disentangled representation of video frames. In all of their experiments with this dataset, the digits are colored. Figure 3.3 shows that DRNET produces blurry results when trained on sequences with uncolored digits. In contrast, it can be seen that MIPAE learns to produce sharp digits even without additional color information.

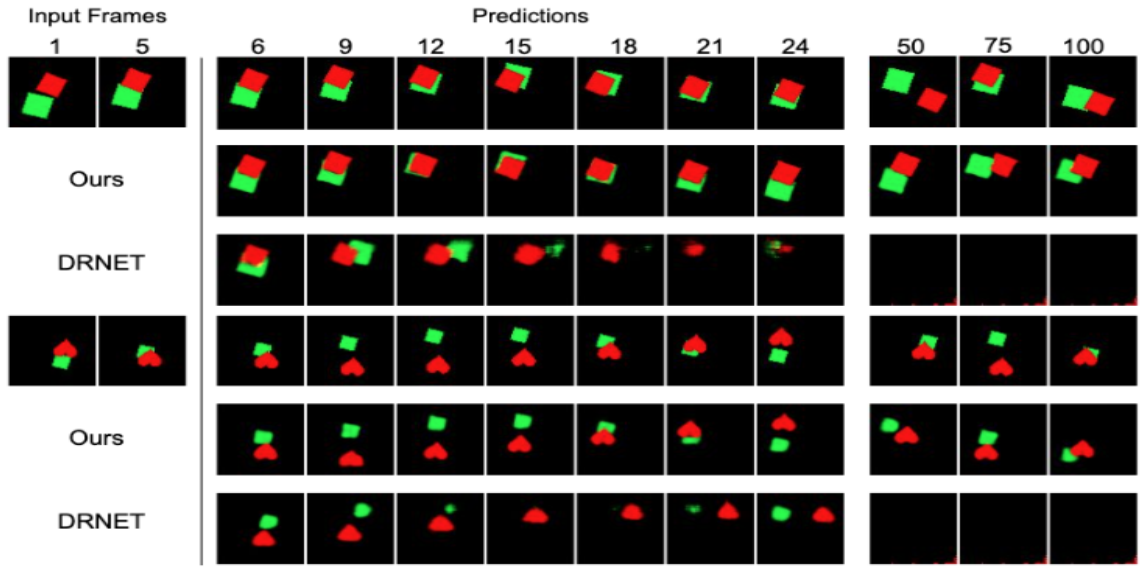
In Figure 3.3(b), we show long-range frame predictions by MIPAE and DRNET on two sequences with different content representations. The idea is to assess the efficacy of the learned disentangled video representation in predicting video frames over longer horizons. It can be seen in Figure 3.3(b) that the proposed MIPAE framework produces frame predictions that are closer to ground truth frames in comparison with DRNET for longer horizons, which indicates that learning an independent set of pose and content representations helps in sustaining frame predictions over longer horizons. This can also be verified quantitatively in Figure 3.6 that frames predicted by our model have lower LPIPS distance with ground truth frames over a longer horizon. It is important to note that in Figure 3.3(b), MIPAE generates two separate digits in the case of video sequences that contain multiple instances of the same digit. DRNET, on the other hand, fails to keep track of similar-looking digits and confuses between the multiple instances, which is partially due to the lack of color information present in our experiments. We could not evaluate the degree of disentanglement quantitatively on this dataset by using the proposed MIG metric due to partial knowledge of the generative factors for MNIST video sequences. Specifically, the true content generative factors f^c of MNIST digits are unknown.

3.5.2 Synthetic Moving Dsprites

The top left and top right parts of Figure 3.4(a) demonstrate the pose content disentanglement by DRNET and MIPAE respectively. It can be seen from this figure that MIPAE produces a sharp and accurate reconstruction of the shapes in these sequences. DRNET fails to reconstruct these shapes accurately. MIPAE captures the content representation z^c of video sequences better due to effective pose content disentanglement. We validate our claim that MIPAE learns better pose content disentanglement than DRNET on the proposed MIG scores for this dataset. MIG metric scores of our method and DRNET can be found in Table.3.1. We also estimated the MI between generative factors and learned representations. Our method achieves a higher mutual information gap between the learned content and poses representations when compared with DRNET, indicating better pose content disentanglement. Visual comparison of generated future frames by both methods further supports our disentanglement claims.



(a) Disentanglement result on moving Dsprites



(b) Long range frame prediction

Figure 3.4 Qualitative comparison of disentanglement on moving DSprites: (a) Demonstration of pose-content disentanglement by DRNET and MIPAE. Images in the grid are generated by taking the pose latent representation of sequence on the top, highlighted in green, and the latent content representation of the images on the margins, highlighted in red. It can be seen that DRNET fails to produce accurate shapes for many objects, whereas our MIPAE model produces sharp reconstruction by capturing true object shapes. This difference in results is due to proper pose content factorization by the proposed framework. (b) Future frames generated by our method and DRNET on moving DSprites dataset. It can be seen that in comparison to DRNET, MIPAE generates coherent and stable long-range predictions.

Table 3.1 MIG score

Dataset	Experiment	$I(f^c, z^c)$	$I(f^c, z^p)$	$I(f^p, z^c)$	$I(f^p, z^p)$	MIG
Dsprites	DRNET[14]	5.6476	0.7483	0.0748	6.3434	0.8574
	Ours	5.6992	0.4660	0.725	6.4977	**0.8975**
MPI3D Real	DRNET[14]	8.1353	0.0376	0.0448	6.2029	0.5658
	Ours	8.3866	0.0461	0.0080	7.1034	**0.6126**

In Figure 3.4(b), it can be seen that DRNET is unable to sustain frame prediction over longer horizons. MIPAE also outperforms DRNET on PSNR, SSIM, and LPIPS metrics, as can be seen in Figure 3.6.

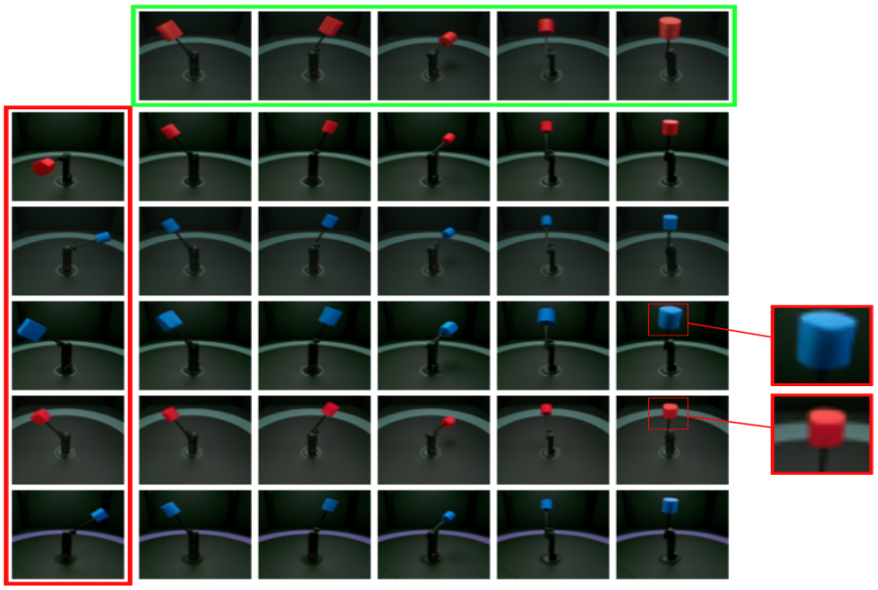
3.5.3 Real-world MPI-3D

MPI3D-Real [22] contains video sequences of mounted objects rotating on a robotic arm at different angular positions. Video sequences in this dataset are generated by changing the angular velocity of the robotic arm, which undergoes mirror reflection upon collision with the ground pixels in the frame.

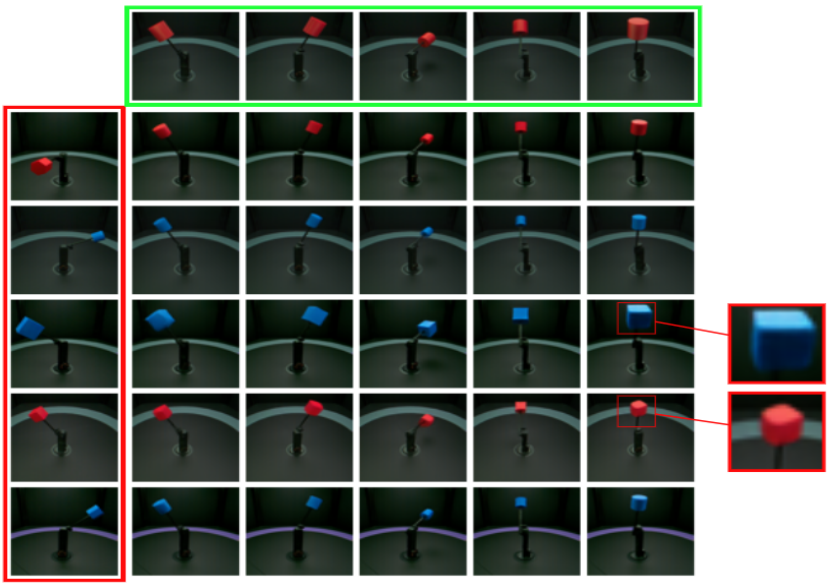
The top and bottom parts of the Figure 3.5 shows the qualitative disentanglement results of DRNET and MIPAE, respectively. It can be seen that DRNET cannot learn accurate content representations of video sequences and does not construct the appearance of object shapes accurately. Specifically, we highlight this difference in the third and fourth rows of this diagram, where it can be seen that DRNET reconstructs a cube as a cylinder. In contrast, MIPAE does not confuse these shapes and constructs them accurately. A quantitative comparison of future frames predicted can be found in Figure 3.6. MIG scores in Table 3.1 also indicate that our method attains better disentanglement than DRNET.

3.6 Model architectures

The proposed MIPAE video prediction framework and the deep learning architectures of the encoder, decoder, and LSTM network are similar to DRNET [14]. For a complete understanding of the model architectures, refer to the next Section 3.6.1. Apart from the model architectures, MIPAE is related to DRNET in some other aspects, such as the objective functions used to train the pose predicting LSTM network and the content encoder being the same. However, our MIPAE framework uses a novel mutual information loss term instead of an adversarial loss to train the pose encoder.



(a) DRNET



(b) MIPAE

Figure 3.5 Qualitative comparison of disentanglement on MPI 3D Real: Demonstration of pose content disentanglement by DRNET and MIPAE. Images in the grid are generated by taking the pose latent variable from the sequence on the top, highlighted in green, and the latent content representation of images highlighted in red. It can be seen that DRNET reconstructs the cube as a cylinder (magnified), whereas our method accurately reconstructs all shapes.

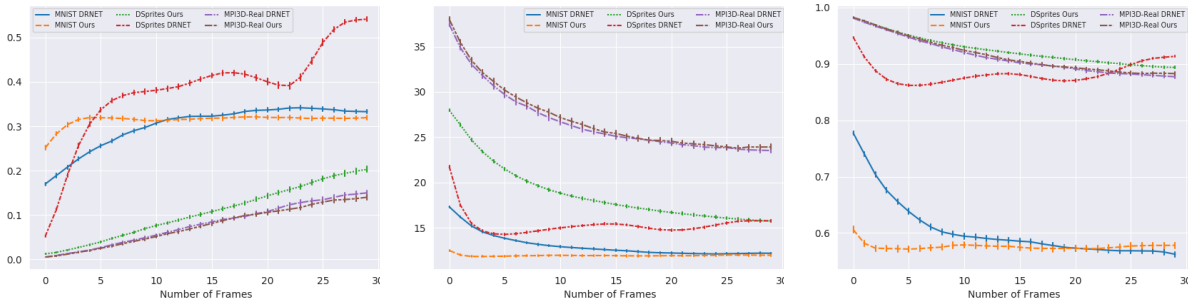


Figure 3.6 Quantitative comparison on SM-MNIST, Dsprites, and MPI3D-Real datasets of future frame prediction over a long-range. The left graph shows the LPIPS distance (lower is better) between ground truth frames and predicted frames. The middle and right graphs show the PSNR and SSIM metric (higher is better) between ground truth and predicted frames.

3.6.1 Training Details

For moving MNIST and DSprites datasets, E_c, and E_p all use DCGAN [63] architecture with $\|z^c\| = 128$ and $\|z_t^p\| = 5$. D is the mirrored version of the encoder where sub-sampling convolutional layers are replaced with deconvolutional layers.

For moving MPI3D-Real, E_p is ResNet-18 [25] architecture and E_c and D are VGG-16 [72] architecture with $\|z^c\| = 128$ and $\|z_t^p\| = 10$. Decoder D is the mirrored version of the content encoder E_c. In the decoder network, spatial up-sampling layers are used instead of pooling layers. We also provide a skip connection from the content encoder to the decoder in U-Net [69] style architecture.

In all experiments, the critic C is a multilayer perceptron with two hidden layers of 512 units each and RELU activation function. We use Adam optimizer [36] with a learning rate 0.002 and $\beta_1 = 0.5$. We choose $\alpha = 1$ and $\beta = 0.0001$ for our training objective as described in Equation 3.6.

Recurrent pose prediction network L is a two-layer LSTM network with 256 cells each, with linear input and output embedding layers. The proposed MIPAE network is trained to observe five context frames to predict ten future frames. For a fair comparison, MIPAE and DRNET are trained for the same number of video frames and hyperparameters.

3.7 Summary

In this chapter, we present a novel approach to disentangling the latent representation space of videos using self-supervised learning. The proposed MIPAE method factorizes the learned representation of each video frame into a set of two mutually independent generative factors, namely pose (time-dependent) and content (time-independent). We experimentally demonstrate the benefits of learning such representation for video prediction, where predicting high dimensional future frames is reduced to the much more straightforward task of predicting low dimensional pose latent vector of the future

frames. In the proposed MIPAE framework, we introduce a novel mutual information loss term to penalize the pose representation of frames from encoding any content like information, which leads to the proper disentanglement of the video representation space. We adopt a Mutual Information Gap (MIG) metric to quantitatively demonstrate the efficacy of our disentanglement approach, which indicates that our method learns substantially better disentangled latent representations, which in turn leads to visually sharp and realistic frame prediction in comparison with another similar disentanglement based video prediction method, the DRNET. The significant improvements over DRNET are achieved by simply replacing their adversarial loss with our mutual information loss term without substantially changing the model architecture or training methodology. The simplicity of this mutual information loss term lends itself to easy integration with other video prediction models.

Chapter 4

Cycle Consistency in Stochastic Video Prediction

In the previous chapter, we discussed a method to learn disentangled representation of video frames by exploiting the temporal consistency in videos. The learned frame representations were used to predict the future frames of a video sequence. Learning factorized representation of video frames reduces the task of predicting future frames in a high-dimensional pixel space to a problem of predicting a set of low-dimensional latent variables which can be decoded to generate high-quality video frames.

This chapter focuses on another crucial challenge in predicting future frames arising from the inherent stochasticity exhibited in real-world videos. Many existing video prediction models assume that given a sequence of context frames, there is only one possible sequence of future frames that can be predicted, i.e., the sequence of future frames is a deterministic function of the previous frames. These models learn to combine all plausible future frame sequences into a single sequence, leading to blurry frame predictions. On the other hand, stochastic video prediction methods use the latent variables in VAEs and GANs to overcome this assumption and capture the inherent uncertainty in predicting future frames. Our approach learns to factorize the latent representation space of videos into a stochastic temporally varying component and a deterministic temporally invariant component. We achieve this disentanglement by employing simple architectural changes in two stochastic video prediction models. We also present a simple but effective cyclic consistency loss term to refine the latent space representation learned by the stochastic video prediction models. We empirically show that the cycle consistency loss term is extremely helpful in generating diverse, sharp, and realistic-looking future frames.

4.1 Stochastic Video Prediction

Stochastic video prediction models [3, 26] use the latent variables in variational autoencoders and generative adversarial networks to capture the mode of uncertainty in predicting future video frames. These methods learn to disentangle the representation space of videos into deterministic and stochastic components. Stochastic Video Generation using Learning Prior, SVG-LP is a model which exploits the variational autoencoder framework [37] for future frame prediction. While methods such as SVG-LP based on the VAE formalism produce diverse predictions, they tend to generate blurry frames. On the

other hand, GANs [24] based prediction models tend to produce sharper frames but are susceptible to mode collapse and training instability, particularly in conditional settings [91]. Stochastic Adversarial Video Prediction, SAVP [40] combines adversarial losses with the variational autoencoder framework to exploit the advantage of both VAE and GAN framework to predict sharp and varied frames.

Our approach builds on top of the SAVP and SVG-LP video prediction frameworks. We learn to decompose video representation into pose and content latent representations by conditioning the prediction model of the SVG-LP with deterministic time-invariant encoding. However, we do not make any such architectural changes to the SAVP model architecture.

4.2 Problem Statement

We aim to generate the next $T - C$ future frames by conditioning on C context frames, where a total of T frames are in a video sequence. Formally, given context frames, $x_{1:C}$, the task is to predict future frames, $\hat{x}_{C+1:T}$. Predicting future frames of a video sequence is challenging due to the high dimensional and stochastic nature of video data. To overcome these challenges, we assume that the frames of a video sequence can be described using two independent sets of information: (i) the appearance of various objects present in the video frames (content information) and (ii) the position of the constituent objects at any time step (pose information). Further, we also assume that the content information is the same for all frames that belong to the same video sequence, whereas the pose information is stochastic and temporally varying.

To model the predictive distribution of future frames, we consider that each frame within a video sequence is generated from an underlying directed graphical model. As shown in Figure 4.1, a frame x_i is generated from two disjoint random variables: content, z_c, and pose, $z_{i,p}$ latent generative factors. Therefore, to predict a frame at time t, we need the content of the frame z_c and pose z_t^p representation. As the latent content variable, z_c is assumed to be the same for all frames, predicting future frames gets reduced to a much simpler task of predicting the low-dimensional future stochastic pose representation, $z_{t,p}$.

The following sections briefly explain the SVG-LP and the SAVP video prediction models.

The stochastic video generation (SVG) model comprises two components. The first component is a prediction model p_θ that predicts the next frame $\hat{x}_t$ by taking as input a latent variable z_t and the previous sequence of frames $x_{1:t-1}$. The second component is the parameterized prior distribution $p(z)$ from which the time-dependent latent variable z_t is sampled at each time step for the next frame prediction. The stochastic latent code z_t contains information regarding all the uncertainty the deterministic model can not capture. The SVG model has two different variants: (i) one in which the latent code is sampled from a fixed prior Gaussian distribution (SVG-FP), and (ii) in which the latent code is sampled from a prior which is a learned distribution (SVG-LP). The SVG-LP model can pass the generated frames and samples from the learned prior back as input to the prediction model after conditioning on a short series of frames.

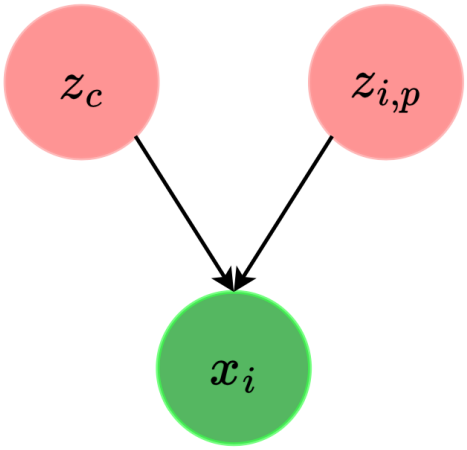


Figure 4.1 Directed graphical model depicting the underlying generative process of generating a video frame. Each frame x_i is generated from samples of two disjoint latent variables, which are the content, z_c, and pose, $z_{i,p}$ representation of that frame

4.3 Stochastic Video Generation with Learned Prior

SVG-LP model training: The true distribution over latent variables z_t is intractable; hence, a time-dependent inference network $q_\phi(z_{1:t}|x_{1:t})$ is used to approximate the intractable posterior distribution with a conditional gaussian distribution, $N(\mu_\phi(x_{1:t}), \sigma_\phi(x_{1:t}))$. This inference network is used to learn a prior, which varies with time. The learned prior is a function of all past frames excluding the frame which is being predicted, $p_\psi(z_t|x_{1:t-1})$. At time step t the prior network observes frames $x_{1:t-1}$ to output the parameters of a conditional Gaussian distribution $N(\mu_\psi(x_{1:t-1}), \sigma_\psi(x_{1:t-1}))$. The parameters of the neural network used to learn the prior distribution are trained jointly with the rest of the model. Figure 4.3(Left) shows the SVG-LP inference procedure, whereas the SVG-LP generation procedure is shown in Figure 4.3(Right).

The objective function used to train the SVG-LP model is given in the equation below:

$$\mathcal{L}_{\theta,\phi}(x_{1:t}) = \sum_{t=1}^{T} \left[\mathbb{E}_{q_\phi(z_{1:t}|x_{1:t})} \log p_\theta(x_t|x_{1:t-1}, z_{1:t}) - \beta D_{KL}(q_\phi(z_t|x_{1:t})||p_\psi(z_t|x_{1:t-1})) \right]$$

Likelihood estimation of the model parameters leads to an l_2 loss between the predicted frame $\hat{x}_t$ and ground truth frame x_t. The model is trained using the re-parameterization trick proposed by (Kingma and Welling, 2014) [37]. The KL divergence multiplier β in the above equation controls the trade-off between fitting the prior and minimizing the frame prediction error. In SAV-LP, frame at time t is predicted by using a recurrent frame predictor p_θ which takes as input the stochastic latent variable z_t which is sampled from the learned prior distribution, $z_t \sim p_\psi(z_t|x_{1:t-1})$. After conditioning on a

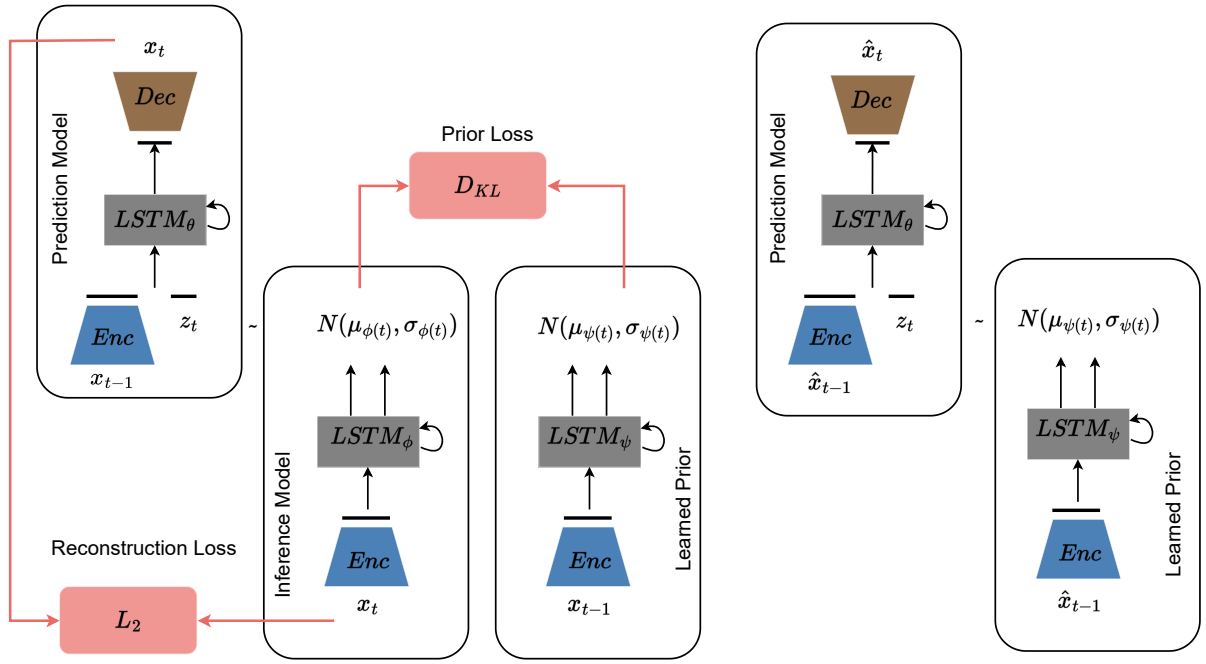


Figure 4.2 Schematic diagram of SVG-LP model [13]: Left: Training with learned prior; Right: Video frame prediction with the SVG-LP model. The orange boxes show the loss functions used to train the SVG-LP model

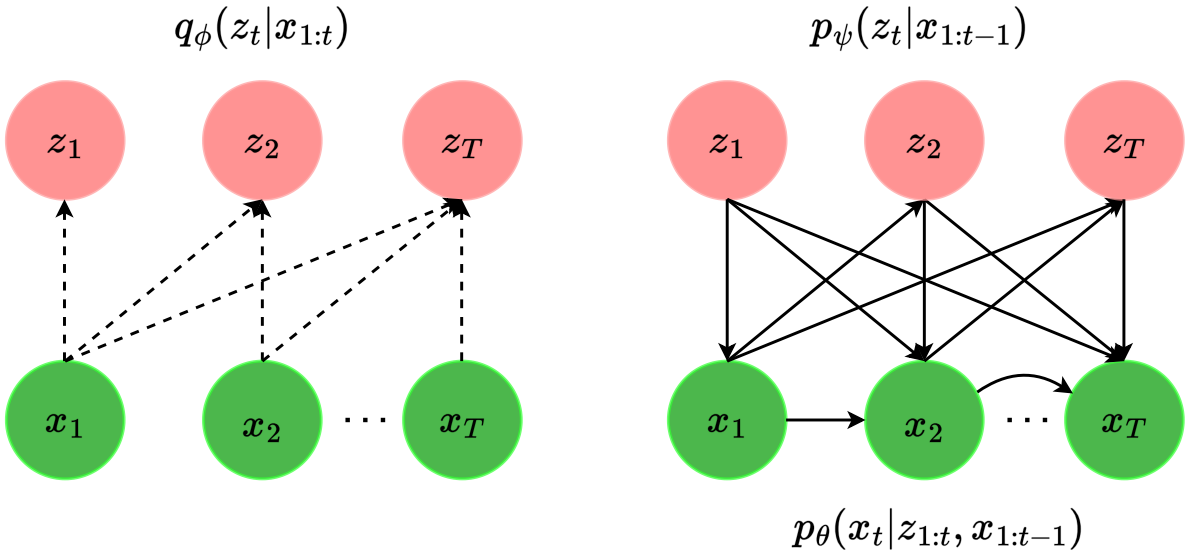


Figure 4.3 Left: Inference and Right: generation in SVG-LP model[13]

short series of ground truth frames, the SVG-LP model can pass the generated frames $\hat{x}_t$ back into the prediction model. The sampling procedure for SVG-LP is shown in Figure 4.2(Right) [13].

4.3.1 SVG-LP: Architecture Details

The frame predictor p_θ, inference network q_ϕ and the learned prior p_ψ are all convolutional-LSTM networks, where the convolutional block is shared across the thee recurrent neural networks. A convolutional frame decoder maps the recurrent frame predictor's output back to the image-pixel space for frame generation.

The generation of a frame at time step t is explained in the equations given below:

$$\begin{aligned} \mu_\phi(t), \sigma_\phi(t) &= LSTM_\phi(h_t) & h_t &= Enc(x_t) \\ z_t &\sim \mathcal{N}(\mu_\phi(t), \sigma_\phi(t)) & & \\ g_t &= LSTM_\theta(h_{t-1}, z_t) & h_{t-1} &= Enc(x_{t-1}) \\ \mu_\theta &= Dec(g_t) & & \end{aligned}$$

Prior distribution parameters at time t are generated as follows:

$$\begin{aligned} h_{t-1} &= Enc(x_{t-1}) \\ \mu_\psi(t), \sigma_\psi(t) &= LSTM_\psi(h_{t-1}) \end{aligned}$$

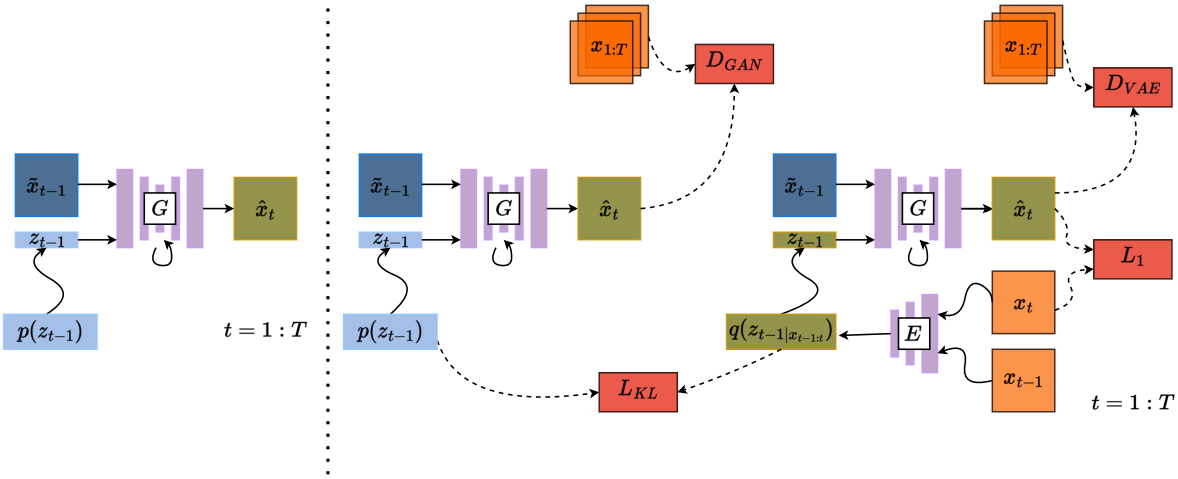


Figure 4.4 Schematic diagram of SAVP model [40]: Left: Test time usage of SAVP Right: Schematic diagram depicting the procedure of training the SAVP video prediction model

4.4 Stochastic Adversarial Video Prediction

The stochastic adversarial video prediction model consists of a deterministic recurrent frame generator G, which takes as input an initial image (either ground truth frame or predicted frame) and a sequence of random latent codes $z_{0:t-1}$ to generate a sequence of future frames $\hat{x}_{1:t}$. Like the SVG-LP video prediction model, the random latent variables encode any uncertainty in predicting the sequence of future frames. At test time, videos are sampled by first sampling the latent codes from a prior distribution $p(z_t)$, which is a fixed standard normal distribution $N(0, 1)$. The sampled latent code and previous fame are passed to the recurrent frame generator for the next frame prediction.

SAVP Training Procedure: The SAVP video prediction framework is based on the VAE-GAN formalism, and its training procedure includes aspects from both variational inference and GANs. The generator of the SAVP model specifies the conditional distribution $p(x_t|x_{0:t-1}, z_{0:t-1})$. It is parameterized as a fixed-variance laplacian distribution with mean $\hat{x}_t = G(x_0, z_{0:t-1})$. The maximum likelihood estimation of data $p(x_{1:t}|x_0)$ is intractable since it involves marginalization over the latent variables. Thus, a variational lower bound of the log-likelihood is maximized. The posterior is approximated using a recognition model $q(z_t|x_{t:t+1})$ which is a gaussian distribution $N(\mu_{z_t}, \sigma^2_{z_t})$ parametarized as a deep neural network $E(x_{t:t+1})$. The encoder network is conditioned on adjacent frames so that z_t can encode any uncertainty in transition between frames x_t and x_{t+1}.

During training, the latent variable z_t is sampled from the recognition model $q(z_t|x_{t:t+1})$. In SVAP, predicting a future frame can be seen as the reconstruction of frame $\hat{x}_{t+1}$ from the information encoded into the stochastic latent variable z_t. Since the latent code is sampled from the recognition model at training time, it has ground truth information about the frame, which has to be reconstructed. SAVP framework uses the conditional VAE formalism in which the encoder and decoder networks are con-

ditioned on the last known frame, either the last predicted frame $\hat{x}_t$ or a ground truth frame x_t when available.

The reconstruction loss is given in the equation below:

$$\mathcal{L}_1(G, E) = \mathbb{E}_{x_{0:T}, z_t \sim E(x_{t:t+1})|_{t=0}^{T-1}} \left[\sum_{t=1}^{T} ||x_t - G(x_0, z_{0:t-1})||_1 \right] \tag{4.1}$$

In their training, a KL divergence term encourages the approximate posterior to be close to the prior distribution. This KL divergence term given below enables sampling from the prior distribution at test time,

$$\mathcal{L}_{KL}(E) = \mathbb{E}_{x_{0:T}} \left[\sum_{t=1}^{T} D_{KL}(E(x_{t-1:t})||p(z_{t-1})) \right] \tag{4.2}$$

The final VAE objective function is given by:

$$G^*, E^* = \underset{G,E}{\operatorname{argmin}} \lambda_1 \mathcal{L}_1(G, E) + \lambda_{KL} \mathcal{L}_{KL}(E) \tag{4.3}$$

To produce shape and clean frame prediction, SAVP uses a classifier D which is capable of distinguishing between generated videos $\hat{x}_{1:T}$ and real videos $x_{1:T}$. The generator is trained by using the binary cross-entropy loss to match the real data distribution:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x_{1:T}} \big[\log D(x_{0:T-1}) \big] + \mathbb{E}_{x_{1:T}, z_t \sim p(z_t)|_{t=0}^{T-1}} \big[\log(1 - D(G(x_0, z_{0:T-1}))) \big] \tag{4.4}$$

The classifier network is trained adversarially along with the generator network,

$$G^* = arg \min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}_{GAN}(G, D) \tag{4.5}$$

The SAVP video prediction model exploits the advantages of both the VAE and GANs to generate sharp and varied frame predictions. GANs can learn to generate high-quality video frames using adversarial training, but they tend to suffer from training instability and model collapse, especially in conditional settings [91]. On the other hand, VAEs explicitly learn the latent representation from observed data which are expressive and meaningful, since the learned encoder produces codes useful for making accurate predictions at training time. However, VAE-based video prediction models fail to produce sharp frame predictions. SAVP model proposes a VAE-GAN model for the task of video frame prediction.

4.4.1 SAVP: Model Architecture

The generator of the SAVP model is a convolutional LSTM with skip connections to the first frame, as done in SNA [16]. The recurrent frame generator predicts the pixel-space transformations between

a set of context frames to the next frame. At every time step, the generator is conditioned on the last known frame (ground truth or predicted) and the stochastic latent codes. The encoder is a feed-forward convolutional network that encodes pairs of images at every time step. The video discriminator is a feed-forward convolutional network with 3D filters based on SNGAN [53].

4.4.2 SAVP - VAE-GAN model

In this section, we introduce the learning objective of the VAE-GAN-based stochastic adversarial video prediction model [40]. The learning objective, $L_{VAE-GAN}$, is given in Equation. 4.6. $L_{VAE-GAN}$ comprises three main loss terms: (i) reconstruction loss L_{Recon}, (ii) KL Divergence L_{KL}, and (ii) two GAN loss terms, L_{GAN} and L_{GAN}^{VAE}. We briefly explain the motivation behind each of these loss terms below,

$$L_{VAE-GAN} = \lambda_{Recon} L_{Recon} + \beta L_{KL} + \alpha_1 L_{GAN} + \alpha_2 L_{GAN}^{VAE} \tag{4.6}$$

Reconstruction Loss: Generation of future frames can be seen as the task of reconstructing frame $\hat{x}_{t+1}$. Minimization of l_2 loss leads to blurry frame predictions [49]. We use l_1 loss for the real-world BAIR dataset and l_2 loss for the synthetic SMNIST dataset.

$$\mathcal{L}_{Recon}(G, E) = \mathbb{E}_{x_{0:T}, z_t \sim E(x_{t:t+1})|_{t=0}^{T-1}} \left[\sum_{t=1}^{T} ||x_t - G(x_0, z_{0:t-1})||_1 \right] \tag{4.7}$$

KL Divergence: A KL divergence term L_{KL} is used in SAVP to enforce that the learned prior distribution $p(z_{t,p}|x_{1:t-1})$ is a strong approximation of the posterior distribution $q(z_{t,p}|x_{2:t})$. Pose latent variables are sampled from the posterior distribution $q(z_{t,p}|x_{2:t})$ during training. At test time, future frames are generated sampling from the learned prior $p(z_{t,p}|x_{1:t-1})$ distribution.

$$\mathcal{L}_{KL}(E) = \mathbb{E}_{x_{0:T}} \left[\sum_{t=1}^{T} D_{KL}(E(x_{t-1:t})||p(z_{t-1})) \right] \tag{4.8}$$

GAN Loss: In SAVP, two different discriminators D and D^{VAE} are used to operate on generated videos based on the distribution used to sample the latent code. The latent codes are sampled from two distributions: (i) the prior distribution and (ii) a posterior distribution. A discriminator D is used on the frames generated by latent code sampled from the posterior distribution. A second discriminator D^{VAE} is applied to the frames generated by sampling from the learned prior distribution. The two GAN loss terms, L_{GAN} and L_{GAN}^{VAE}, enforce that the video generated by sampling from the learned prior and the posterior distribution follows the data generating distribution. In Equation. 4.9, we mention the L_{GAN} loss term. The other adversarial loss L_{GAN}^{VAE} is analogous to L_{GAN}. Both the discriminators have the same architecture but do not share weights.

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x_{1:T}} \big[\log D(x_{0:T-1}) \big] + \mathbb{E}_{x_{1:T}, z_t \sim p(z_t)|_{t=0}^{T-1}} \big[\log(1 - D(G(x_0, z_{0:T-1}))) \big] \tag{4.9}$$

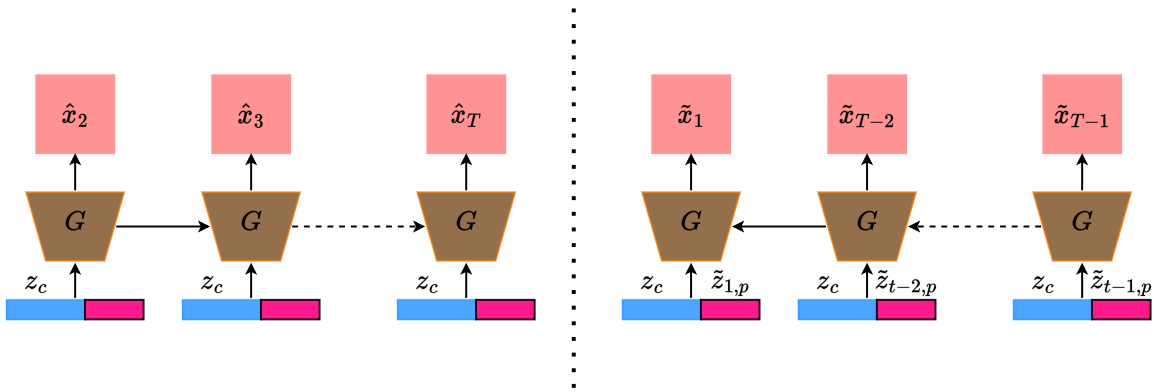


Figure 4.5 Implementation of cycle consistency: Left: Forward prediction of $\hat{x}_{2:T}$. Right: Reverse prediction of $\tilde{x}_{(T-1):1}$

The final optimization objective for the SAVP frame prediction model is:

$$G^*, E^* = arg\min_{\mathbf{G,E}} \max_{\mathbf{D,D^{VAE}}} \mathcal{L}_{VAE-GAN} \tag{4.10}$$

4.5 Cycle consistency loss

This section introduces our *Cycle Consistency* loss term. The problem of video frame prediction can be viewed as learning the transformation that maps a set of context frames to a set of possible future frames. While stochastic video prediction models successfully learn the stochastic transformation from a sequence of context frames to a sequence of future frames, they fail to factorize the latent representation space into generative factors. This limits the ability of stochastic video prediction models to generate long-range sharp frame predictions. To disentangle the video representation space learned by stochastic video prediction models, we propose a simple architecture change of conditioning the prediction model of SVG-LP [13] and SAVP [40] prediction frameworks with time-independent content representation which is same for all frames belonging to the video sequence. In our experiments, we use a recurrent content encoder, E_{con}, which produces a single content encoding z_c consistent across all frames belonging to the same video sequence. Conditioning the prediction model with content-encoding enables the model to learn the complementary pose-like information to predict future frames correctly.

In our experiments, we incorporate a cycle consistency loss term which enforces transitivity of the learned transformation. That is, once the future frames are predicted from the context frames in the forward direction, then by transforming the predicted future frames in reverse order, we try to recover the context frames. Formally, Cycle consistency ensures that if $S : x_{1:C} \rightarrow x_{C+1:T}$ is the forward transformation, mapping a sequence of context frames to a sequence of the future frames, then the same transform must also be able to predict the context frames from the predicted frames, i.e., $S(x_{T:C+1}) = x_{C:1}$. Effectively, this leads to a regularization effect on the number of possible transformations mapping

context frames to future frames. We empirically demonstrate that cycle consistency constraint helps predict sharp and varied frames over a longer horizon.

To compute the cycle consistency loss, L_{CC}, first $\hat{x}_{2:T}$ are predicted in forward prediction, then as shown in Figure 4.5, reverse prediction $\tilde{x}_{T-1:1}$ are made in the reverse order. In the general formulation of this loss function, the distance between $\hat{x}_{2:T-1}$ and $\tilde{x}_{T-1:2}$ is minimized using n-norm, where both the forward predicted frames $\hat{x}_{2:T-1}$ and reverse frames $\tilde{x}_{T-1:2}$ are generated using the same pose $z_{2:T,p}$ encodings. Specifically, our experiments use the l_2 norm for SM-MNIST and the l_1 norm for the BAIR dataset.

$$L_{CC} = \mathbb{E}_{\hat{x}_{1:T}, z_{t,p} \sim q(\tilde{z}_{t,p}|x_{2:t})|_2^T} \Big[\frac{1}{T-2} \sum_{T-1}^{2} ||\hat{x}_t - G(z_c, \tilde{z}_{t,p})||_n \Big] \tag{4.11}$$

So the total loss L_{Total} becomes as in Equation 4.12

$$L_{Total} = L_{VAE-GAN} + \lambda_{CC} L_{CC} \tag{4.12}$$

Let $\mathbb{E}$ be the set of all encoder, E_{pos}, E_{prior} and E_{con} and let $\mathbb{D}$ be the set of all discriminators, D and D^{VAE}. Then the training objective is given in equation 4.13.

$$G^*, \mathbb{E}^*, \mathbb{D}^* = arg \min_{G,\mathbb{E}} \max_{\mathbb{D}} L_{Total} \tag{4.13}$$

So the total loss L_{Total} becomes as in Equation 4.14

$$L_{Total} = L_{VAE-GAN} + \lambda_{CC} L_{CC} \tag{4.14}$$

4.6 Architecture Details

We test the disengagement efficacy of our approach on BAIR and SMNIST datasets. While we adopt the VAE-GAN based SAVP [40] video prediction framework for the real-world BAIR dataset. In the case of the synthetic dataset, SMINST, VAE-based SVG-LP [13] model has been used in our experiments.

4.6.1 Architecture details for SMNIST

We condition the prediction model of SVG-LP with time-independent content information by using the recurrent DCGAN [64] content encoder E_{con} for to achieve pose content disentanglement. The content encoder infers only one content encoding for all context frames [32]. We build on this by incorporating cycle consistency loss to get our cycle consistent SVG-LP model, **CC-SVGLP**. Weights for the gan losses L_{GAN} and L_{GAN}^{VAE}, α_1 and α_2 in equation 4.6, are set to 0. For CC-SVGLP, we set $\lambda_{Recon} = \lambda_{CC} = 0.5$ and for the baseline mentioned above we set $\lambda_{Recon} = 1$ & $\lambda_{CC} = 0$.

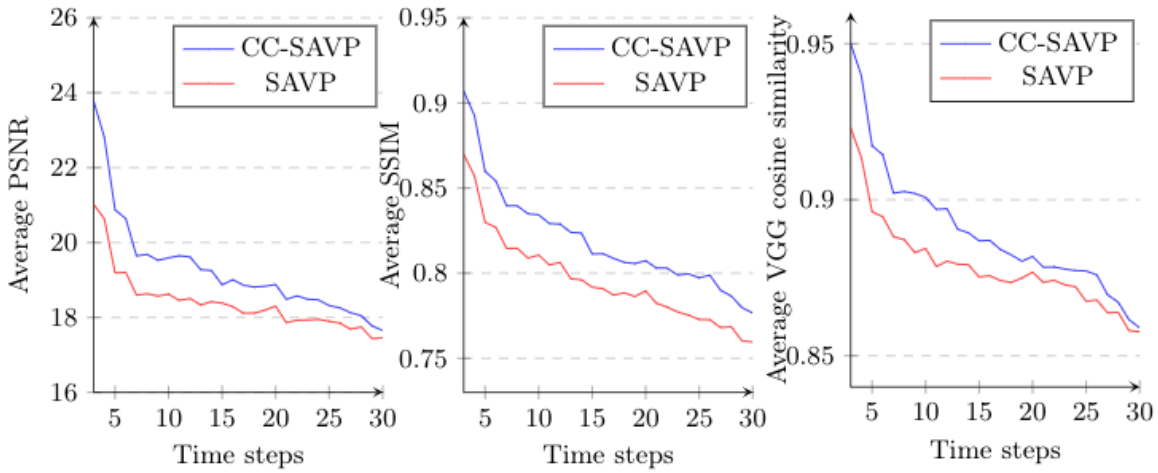


Figure 4.6 The above three plots depict the average similarity between ground truth and the best-predicted frames. We pick the best-predicted sequence out of 100 predicted sequences for each test sequence and average the similarity score overall for test samples. Although PSNR and SSIM [83] correspond poorly to human perception [89], we include this metric for completeness. VGG cosine similarity scores match better with human perception [89]. It can be seen that our model CC-SAVP significantly outperforms SAVP on all three evaluation metrics.

4.6.2 Architecture details for BAIR

For the real-world BAIR dataset, we adopt a VAE-GAN-based SAVP model. SAVP is a conditional video prediction model in which the content information is passed to the generator by feeding the previous known frame to make future predictions. We do not condition the generator of SAVP with a content encoder as feeding the generator on the previous frame can be approximately considered as providing content information. However, CC-SAVP differs from the SAVP baseline model in three significant ways. First, the latent variable LSTM of the generator network is removed, and we pass $z_{t-1,p}$ along with $z_{t,p}$ to all convolutional layers by tiling and concatenating along channel dimensions. Second, instead of passing two frames $x_{t-1:t}$, we pass only one frame x_t to the posterior encoder E_{pos} and make the encoder recurrent by adding a fully connected LSTM layer. Finally, for the sake of consistency, at inference time, we sample from the learned prior $p(z_{t,p}|x_{1:t-1})$ [13] rather than sampling from the standard normal distribution. Prior pose encoder E_{prior} has the same architecture as the posterior encoder E_{pos}. We set $\lambda_{Recon} = \lambda_{CC} = 0.5$ and $\alpha_1 = \alpha_2 = 0.1$.

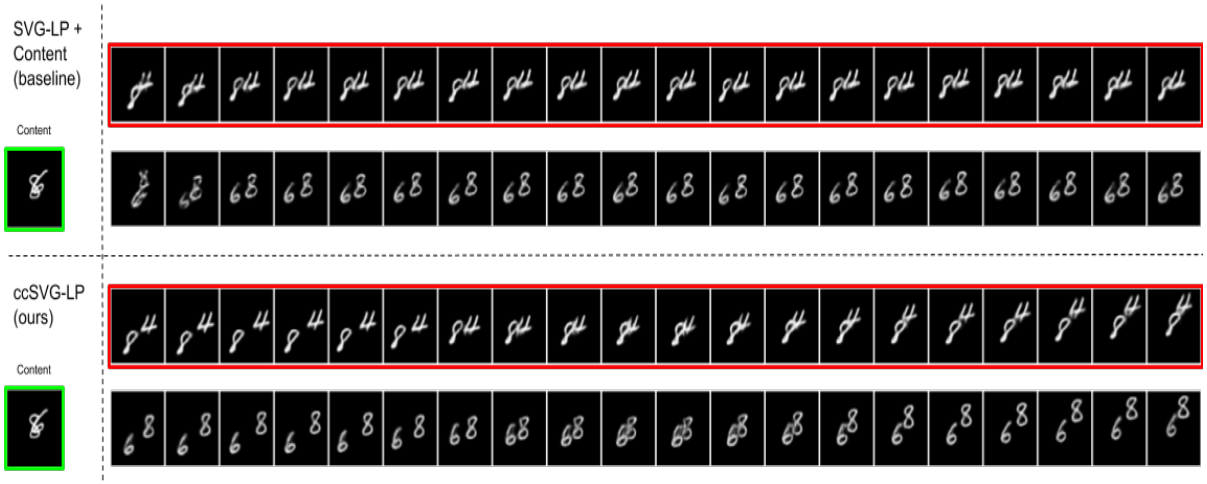
4.7 Experiments

We evaluate the competence of our approach in learning disentangled representation of video and its subsequent effect on long-range video prediction on synthetic, SMNIST, and real-world BAIR datasets. For BAIR, we quantitatively compare the quality of generated frames between SAVP and CC-SAVP by computing the structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR). We also assess the perceptual quality of the generated frames by computing the cosine similarity between VGG features [40].

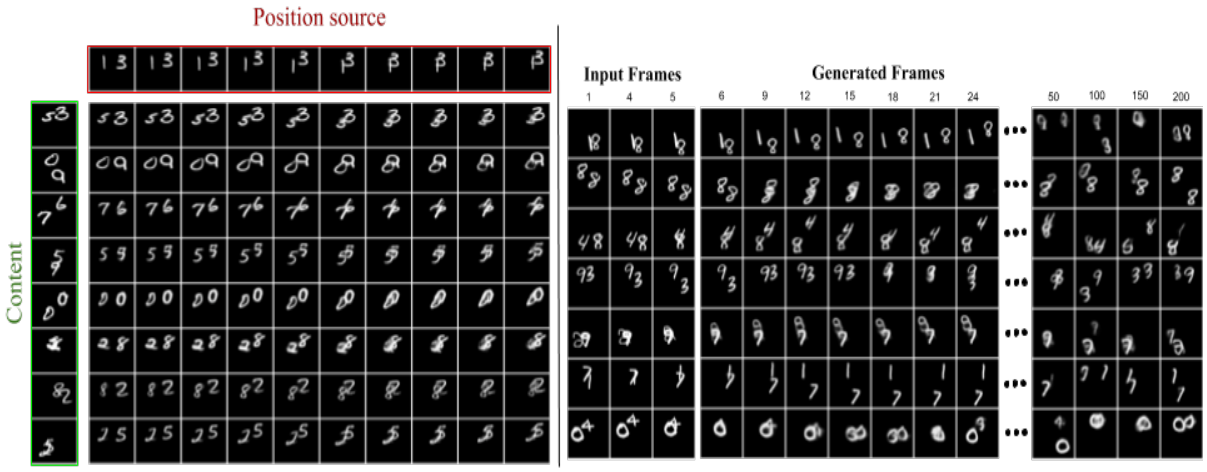
4.7.1 Stochastic Moving MNIST

Stochastic moving MNIST dataset consists of video sequences with one or more MNIST [74] digits bouncing around in a frame of size 64×64. MNIST digits move with constant velocity and change velocity and direction randomly upon collision with the frame boundaries.

We consider SVGLP with content encoder as the baseline architecture for this dataset. Figure 4.7(a) demonstrates the baseline pose/ content disentanglement results and the proposed CC-SVGLP architecture. In this experiment, we combine pose representation $z_{2:T,p}$ of the predicted sequence at the top, highlighted in red, and the content representation of the image from the margin, highlighted in green, to generate 20 future frames. The digits in the generated sequence closely follow the position of the digits of the target sequence. This demonstrates that conditioning the SVG-LP architecture with time-independent content-encoding leads to the video representation space factorization. In sequence generated by the baseline, architecture digits appear to be hovering around the center of the frame. However, in the sequence generated by our model CC-SVGLP which enforces cyclic consistency of predicted



(a) Disentanglment results on SMNIST



(b) Qualitative comparison of disentanglement

Figure 4.7 (a) Comparison between our baseline and CC-SVGLP in learning disentangled video representation. The pose latent variable of the topmost sequence, highlighted in red, is combined with the content representation of the image in the green box to generate new sequences. The digits in the generated sequence follow the pose of digits from the target sequence (b) Bottom-left: The experiment shown above is repeated on a large variety of images to test the efficacy of pose/content factorization by our approach (b) Bottom-right: Each row is a long-range prediction of 200 frames by CC-SVGLP, given five context frames.

frames, the digits demonstrate an expressive display of movement. Thus, the cycle consistency loss improves the disentangled representation space.

In the bottom left side of Figure 4.7, we show the pose/ content disentanglement results of our approach, CC-SVGLP. The sequence on the top, highlighted in red, is the source sequence from which the pose representation is combined with the content of images from the right to generate a sequence of video frames. MNIST digits in all generated sequences follow the pose of digits from the source sequence, irrespective of content. In the bottom right side of Figure 4.7, we test the long-range prediction efficacy of our CC-SVGLP approach by generating sharp, varied video predictions up to 200 frames into the future. Note that all predictions are conditioned on an initial set of five ground truth frames.

4.7.2 BAIR robot pushing dataset

BAIR robot pushing dataset [16] is a challenging video prediction dataset. It consists of video sequences of a robotic arm interacting with various objects in a cluttered environment. The robotic arm performs stochastic movements exhibiting complex real-world dynamics. The spatial resolution of video frames in the BAIR robotic pushing dataset is 64×64.

We show the pose/content disentanglement results in Figure 4.8(Top). Latent pose variables from the top predicted sequence, highlighted in red, are combined with content information from frames in green to generate sequences shown in the grid. The robot's arm in the generated sequences follows the trajectory of the arm in the source sequence. These results demonstrate that our CC-SAVP model effectively learns to disentangle pose from content. CC-SAVP is the first stochastic video prediction model to demonstrate clean pose/ content factorization on real-world videos.

Figure 4.8(Bottom) depicts long-range stochastic video predictions generated by our model. We generate a different possible future sequence for each input sequence by sampling from the prior. CC-SAVP generates sharp and varied long-range predictions. We display 150 future frame predictions here, but our model can predict up to and beyond 500 future frames with graceful degradation over time. Cycle consistency loss supports long-range predictions by forcing the model to learn proper pose/ content disentanglement.

Figure 4.6 shows the quantitative comparison between CC-SAVP and SAVP. These plots show the average similarity between the ground truth sequence and the closest predicted sequence. The left and middle plots are average PSNR and SSIM metric scores; these are not designed for video prediction and are known to correspond to human judgment poorly. The third plot is average VGG cosine similarity, which has been shown to match human perception better. Our model CC-SAVP outperforms SAVP on all three metrics. We have used publicly available code to produce SAVP predictions.

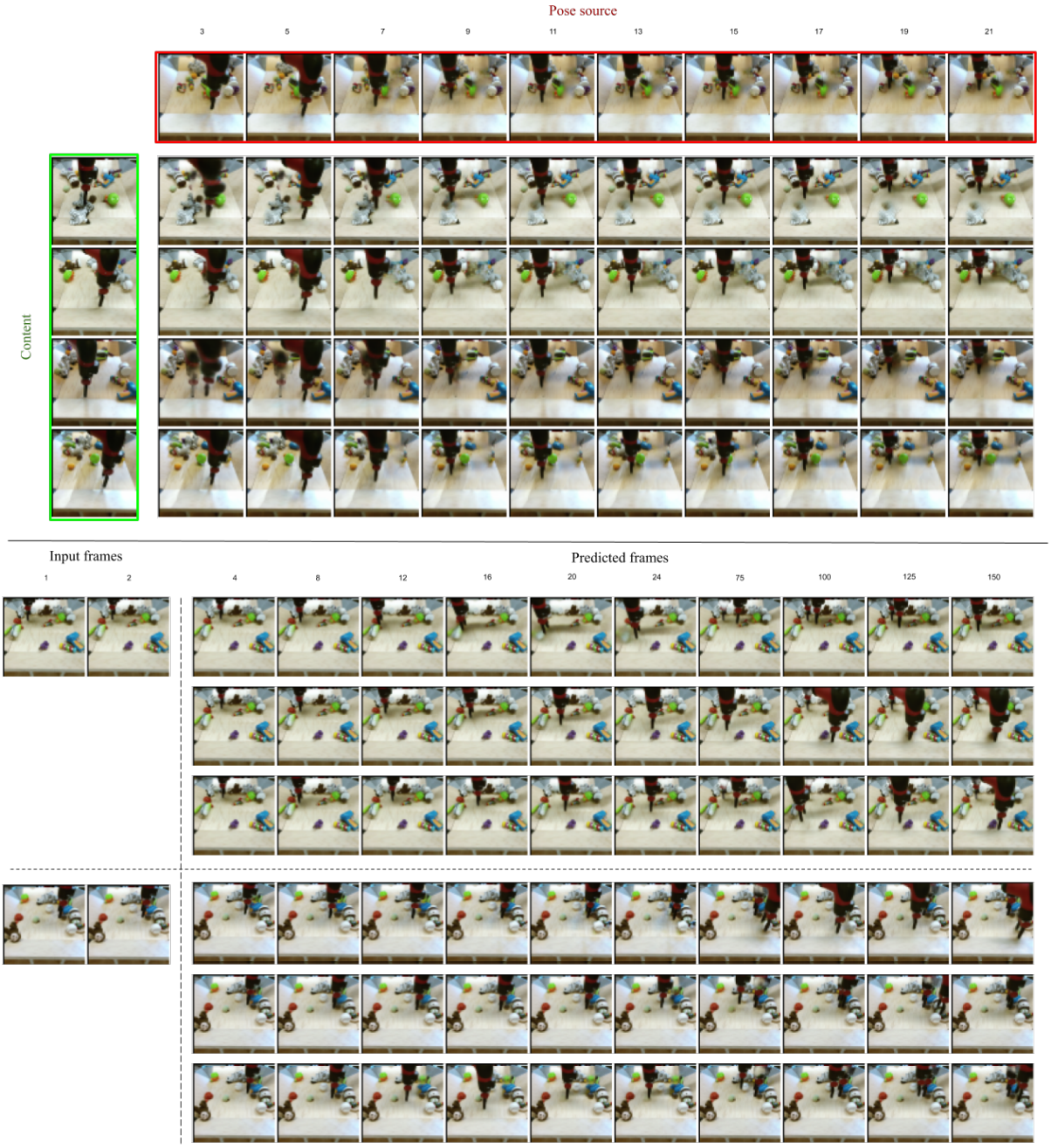


Figure 4.8 Top: Pose/ content disentanglement results on the real world BAIR robot pushing dataset by our CC-SAVP model. The pose information of the predicted sequence in the top row, highlighted in red, is combined with the different content representations of frames in green to generate video sequences. The robotic arm in generated sequences accurately follows the target sequence's motion. Bottom: Input frames on the left generate multiple long-range stochastic video predictions. It can be seen that the system generates varied, sharp predictions up to 150 time steps. Note that the generator has been conditioned on only two context frames.

4.8 Summary

We experimentally demonstrate the efficacy of our approach in learning pose/ content disentanglement on two different stochastic video predictions by simply conditioning the prediction model with time-invariant context information using a content encoder. Further, we formulate an easy-to-interpret and train cycle consistency loss term, which helps learn a refined predictive model of the time-variant stochastic latent variable pose. Our ablation results establish the qualitative and quantitative advantages of using cycle consistency loss and our approach to learning factorized pose and content representation of long-range video prediction. Applying cycle consistency on two different stochastic video prediction models shows that this approach is sufficiently general and can be easily incorporated in other similar latent variable video frame prediction models. Cycle consistency enables the model to learn the complex underlying generative distribution of the pose latent variable.

Chapter 5

Conclusion and Future Work

Visual prediction is an active area of research in computer vision that has been studied in several contexts, such as trajectory forecasting, early recognition, human pose estimation, and future frame prediction. Specifically, video frame prediction models can hallucinate the visual appearance of future frames and have found large-scale applications in reinforcement learning, robotics, and healthcare.

Despite significant improvements in deep generative modelling techniques, learning the non-linear mapping between frames is challenging due to video data's stochastic and high dimensional nature. In this thesis, we presented a self-supervised learning approach known as MIPAE, which disentangled the latent representation of video frames by leveraging the temporal consistency present in videos. In the proposed work, the low-dimensional disentangled representation space consists of two components - temporally varying pose and temporally consistent content factors. Using a novel mutual information loss term, we penalized the pose representation of frames belonging to the same video sequence from encoding dependent information. Further, we attained proper pose content disentanglement by penalising the content encoding an MSE loss term, forcing them to extract slowly varying features from the video.

We empirically demonstrated the efficacy of our approach in learning disentangled representations of videos using latent traversal. We also adopted a mutual information-based metric, MIG, to quantitatively evaluate the degree of disentanglement achieved by our method. MIG scores show that our MIPAE framework is a superior disentanglement approach compared to similar models. It was shown in qualitative results that MIPAE produces sharper frames because of its better pose content factorization.

Other important sets of video prediction methods use stochastic latent variable [13, 40] in generative models like variational autoencoders [37] and generative adversarial networks [24] to model the inherent uncertainty in making futures fame prediction by simply looking at a small number of context frames from a video sequence. These models decompose the latent space representation into deterministic and stochastic components. In Chapter 4, we build upon the work of two such stochastic video prediction models by conditioning their prediction model with time-independent content encoding to disentangle the representation space into temporally varying stochastic pose and temporally consistent deterministic content representations. Our approach is based on the intuition that the major component of stochasticity

in real-world video comes from the uncertainty in the motion of various objects in the scene. This is based on the assumption that the content in the videos does not change with time. This work also presents the application of a cycle consistency loss term which significantly improves the quality of the predicted frame.

MIPAE framework assumes a single content representation for all frames in the same video sequence. This assumption does not account for the cases where objects can freely enter or exit a video frame or appear abruptly from behind objects already in videos. A significant extension of the video prediction models presented in this thesis could be in the direction of easing out this assumption.

Related Publications

1. Ujjwal Tiwari, Aditya P. Sreekar, Anoop M. Namboodiri, “Cycle Consistency based Method for Learning Disentangled Representation for Stochastic Video Prediction”, 21st International Conference on Image Analysis and Processing, ICIAP 2021

2. Aditya P. Sreekar, Ujjwal Tiwari, Anoop M. Namboodiri, “Mutual Information based Method for Unsupervised Disentanglement of Video Representation”, International Conference on Pattern Recognition, ICPR 2020

3. Aditya P. Sreekar, Ujjwal Tiwari, Anoop M. Namboodiri, “Reducing the Variance of Variational Estimates of Mutual Information by Limiting the Critic’s Hypothesis Space to RKHS”, International Conference on Pattern Recognition, ICPR 2020

Bibliography

[1] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

[3] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006.

[5] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540, 2018.

[6] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[7] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[8] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[9] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[10] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[11] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed. Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*, 2017.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[13] E. Denton and R. Fergus. Stochastic video generation wi?th a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[14] E. L. Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017.

[15] M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

[16] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *CoRL*, 12:16, 2017.

[17] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

[18] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[19] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[20] J. H. Friedman, F. Baskett, and L. J. Shustek. An algorithm for finding nearest neighbors. *IEEE Transactions on computers*, 100(10):1000–1006, 1975.

[21] S. Ghimire, P. K. Gyawali, and L. Wang. Reliable estimation of kullback-leibler divergence by controlling discriminator complexity in the reproducing kernel hilbert space. *arXiv preprint arXiv:2002.11187*, 2020.

[22] M. W. Gondal, M. Wüthrich, . Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *arXiv preprint arXiv:1906.03292*, 2019.

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] M. Henaff, J. Zhao, and Y. LeCun. Prediction under uncertainty with error-encoding networks. *arXiv preprint arXiv:1711.04994*, 2017.

[27] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[28] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[29] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

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

[31] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[32] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018.

[33] W.-N. Hsu, Y. Zhang, and J. Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in neural information processing systems*, pages 1878–1889, 2017.

[34] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1771–1779. JMLR. org, 2017.

[35] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[38] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[39] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[40] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[41] Y. Li. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] Y. Li and S. Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

[43] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[44] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[45] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

[46] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[47] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[48] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[49] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[50] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.

[51] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[52] J. R. Medel and A. Savakis. Anomaly det0ection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[53] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[54] Y.-I. Moon, B. Rajagopalan, and U. Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318, 1995.

[55] A. Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[56] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[57] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[58] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.

[59] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[60] C. Paxton, Y. Barnoy, K. Katyal, R. Arora, and G. D. Hager. Visual robot task planning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8832–8838. IEEE, 2019.

[61] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[62] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker. On variational lower bounds of mutual information. In *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

[63] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 11 2015.

[64] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[65] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[66] S. Reed, A. v. d. Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, D. Belov, and N. de Freitas. Parallel multiscale autoregressive density estimation. *arXiv preprint arXiv:1703.03664*, 2017.

[67] S. Roberts and R. Everson. *Independent component analysis: principles and practice*. Cambridge University Press, 2001.

[68] L. Rokach and O. Maimon. Decision trees. *Data mining and knowledge discovery handbook*, pages 165–192, 2005.

[69] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[70] B. C. Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357, 2014.

[71] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[72] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[73] J. Song and S. Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

[74] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[75] S. Suthaharan and S. Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[76] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[77] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[78] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[79] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[80] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[81] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[82] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1020–1028, 2017.

[83] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[84] Wikipedia. Autoencoder — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Autoencoderoldid=1166479437, 2023. [Online; accessed 30-July-2023].

[85] Wikipedia. Long short-term memory — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Long%20short-term%20memoryoldid=1166629581, 2023. [Online; accessed 30-July-2023].

[86] Wikipedia. Variational autoencoder — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Variational%20autoencoderoldid=1167554236, 2023. [Online; accessed 30-July-2023].

[87] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[88] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206, 2007.

[89] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[90] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[91] Y. Zhou and T. L. Berg. Learning temporal transformations from time-lapse videos. In *European Conference on Computer Vision*, pages 262–277. Springer, 2016.