Disentangling Multiple Features in Video Sequences using Gaussian Processes in Variational Autoencoders

Sarthak Bhagat[1], Shagun Uppal[1][*], Zhuyun Yin[2], and Nengli Lim[3]

[1] IIIT Delhi
{sarthak16189, shagun16088}@iiitd.ac.in
[2] Bioinformatics Institute, A*STAR, Singapore
yinzhuyun@gmail.com
[3] Singapore University of Technology and Design
nengli_lim@sutd.edu.sg

Abstract. We introduce MGP-VAE (Multi-disentangled-features Gaussian Processes Variational AutoEncoder), a variational autoencoder which uses Gaussian processes (GP) to model the latent space for the unsupervised learning of disentangled representations in video sequences. We improve upon previous work by establishing a framework by which multiple features, static or dynamic, can be disentangled. Specifically we use fractional Brownian motions (fBM) and Brownian bridges (BB) to enforce an inter-frame correlation structure in each independent channel, and show that varying this structure enables one to capture different factors of variation in the data. We demonstrate the quality of our representations with experiments on three publicly available datasets, and also quantify the improvement using a video prediction task. Moreover, we introduce a novel geodesic loss function which takes into account the curvature of the data manifold to improve learning. Our experiments show that the combination of the improved representations with the novel loss function enable MGP-VAE to outperform the baselines in video prediction.

1 Introduction

Finding good representations for data is one of the main goals of unsupervised machine learning [3]. Ideally, these representations reduce the dimensionality of the data, and are structured such that the different factors of variation in the data get distilled into different channels. This process of disentanglement in generative models is useful as in addition to making the data interpretable, the disentangled representations can also be used to improve downstream tasks such as prediction.

In prior work on the unsupervised learning of video sequences, a fair amount of effort has been devoted to separating motion, or dynamic information from static content [7,11,14,22,31]. To achieve this goal, typically the model is structured to consist of dual pathways, e.g. using two separate networks to separately capture motion and semantic content [7,31].

Such frameworks may be restrictive as it is not immediately clear how to extend them to extract multiple static and dynamic features. Furthermore, in complex videos, there usually is not a clear dichotomy between motion and content, e.g. in videos containing dynamic information ranging over different time-scales.

In this paper, we address this challenge by proposing a new variational autoencoder, MGP-VAE (Multi-disentangled-features Gaussian Processes Variational AutoEncoder), for the unsupervised learning of video sequences. It utilizes a latent prior distribution that consists of multiple channels of fractional Brownian motions and Brownian bridges. By varying the correlation structure along the time dimension in each channel to pick up different static or dynamic features, while maintaining independence between channels, MGP-VAE is able to learn multiple disentangled factors.

We then demonstrate quantitatively the quality of our disentanglement representations using a frame prediction task. To improve prediction quality, we also employ a novel geodesic loss function which incorporates the manifold structure of the data to enhance the learning process.
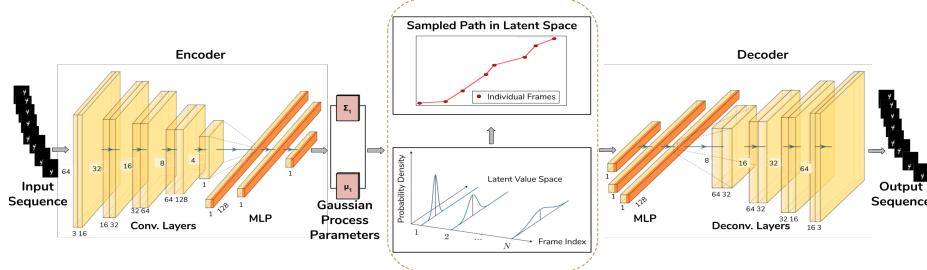

Fig. 1. Network illustration of MGP-VAE: The network takes in a video sequence, an array of images, and encodes a Gaussian process latent space representation. The output of the encoder is the mean and covariance matrix of the Gaussian process, after which a sequence of points in $\mathbb{R}^d$ is sampled where each point represents one frame.

Our main contributions can be summarized as follows:

– We use Gaussian processes as the latent prior distribution in our model MGP-VAE to obtain disentangled representations for video sequences. Specifically, we structure the latent space by varying the correlation between video frame distributions so as to extract multiple factors of variation from the data.
– We introduce a novel loss function which utilizes the structure of the data manifold to improve prediction. In particular, the actual geodesic distance

between the predicted point and its target on the manifold is used instead of squared-Euclidean distance in the latent space.
– We test MGP-VAE against various other state-of-the-art models in video sequence disentanglement. We conduct our experiments on three datasets and use a video prediction task to demonstrate quantitatively that our model outperforms the competition.

2   Related Work

2.1   Disentangled Representation Learning for Video Sequences

There are several methods for improving the disentanglement of latent representations in generative models. InfoGAN [6] augments generative adversarial networks [10] by additionally maximizing the mutual information between a subset of the latent variables and the recognition network output. beta-VAE [13] adds a simple coefficient (β) to the KL divergence term in the evidence lower bound of a VAE. It has been demonstrated that increasing β beyond unity improves disentanglement, but also comes with the price of increased reconstruction loss [18]. To counteract this trade-off, both FactorVAE [18] and β-TCVAE [5] further decompose the KL divergence term, and identify a total correlation term which when penalized directly encourages factorization in the latent distribution.

With regard to the unsupervised learning of sequences, there have been several attempts to separate dynamic information from static content [7,11,14,22,31]. In [22], one latent variable is set aside to represent content, separate from another set of variables used to encode dynamic information, and they employ this graphical model for the generation of new video and audio sequences.

[31] proposes MCnet, which uses a convolutional LSTM for encoding motion and a separate CNN to encode static content. The network is trained using standard l_2 loss plus a GAN term to generate sharper frames. DRNet [7] adopts a similar architecture, but uses a novel adversarial loss which penalizes semantic content in the dynamic pathway to learn pose features.

[14] proposes DDPAE, a model with a VAE structure that performs decomposition on video sequences with multiple objects in addition to disentanglement. In their experiments, they show quantitatively that DDPAE outperforms MCnet and DRNet in video prediction on the Moving MNIST dataset.

Finally, it has been shown that disentangled representation learning can be placed in the framework of nonlinear ICA [17], particularly in the context of time-varying data [15].

2.2   VAEs and Gaussian Process Priors

In [11], a variational auto-encoder which structures its latent space distribution into two components is used for video sequence learning. The "slow" channel extracts static features from the video, and the "fast" channel captures dynamic motion. Our approach is inspired by this method, and we go further by giving

a principled way to shape the latent space prior so as to disentangle multiple features.

Outside of video analysis, VAEs with a Gaussian process prior have also been explored. In [4], they propose GPPVAE and train it on image datasets of different objects in various views. The latent representation is a function of an object vector and a view vector, and has a Gaussian prior imposed on it. They also introduce an efficient method to speed up computation of the covariance matrices.

In [8], a deep VAE architecture is used in conjunction with a Gaussian process to model correlations in multivariate time series such that inference can be performed on missing data-points.

Bayes-Factor VAE [19] uses a hierarchical Bayesian model to extend the VAE. As with our work, they recognize the limitations of restricting the latent prior distribution to standard normal, but they adopt heavy-tailed distributions as an alternative rather than Gaussian processes.

2.3 Data Manifold Learning

Recent work has shown that distances in latent space are not representative of the true distance between data-points [1,21,27]. Rather, deep generative models learn a mapping from the latent space to the data manifold, a smoothly varying lower-dimensional subset of the original data space.

In [23], closed curves are abstractly represented as points on a shape manifold which incorporates the constraints of scale, rotational and translational invariance. The geodesic distance between points on this manifold is then used to give an improved measure of dissimilarity. In [28], several metrics are proposed to quantify the curvature of data manifolds arising from VAEs and GANs.

3   Method

In this section, we review the preliminaries on VAEs and Gaussian processes, and describe our model MGP-VAE in detail.

3.1 VAEs

Variational autoencoders [20] are powerful generative models which reformulate autoencoders in the framework of variational inference. Given latent variables $z \in \mathbb{R}^M$, the decoder, typically a neural network, models the generative distribution $p_\theta(x \mid z)$, where $x \in \mathbb{R}^N$ denotes the data. Due to the intractability of computing the posterior distribution $p(z \mid x)$, an approximation $q_\phi(z \mid x)$, again parameterized by another neural network called the encoder, is used. Maximizing the log-likelihood of the data can be achieved by maximizing the evidence lower bound

$$\mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right], \tag{1}$$

which is equal to

$$\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left[q_\phi\left(z|x\right)\Big| p(z)\right], \tag{2}$$

with $p(z)$ denoting the prior distribution of the latent variables.

The negative of the first term in (2) is the reconstruction loss, and can be approximated by

$$\frac{1}{L}\sum_{l=1}^{L} -\log p_\theta\left(x\,\big|\,z^{(l)}\right), \tag{3}$$

where $z^{(l)}$ is drawn (L times) from the latent distribution, although typically only one sample is required in each pass as long as the batch size is sufficiently large [20]. If $p_\theta\left(x\,\big|\,z\right)$ is modeled to be Gaussian, then this is simply mean-squared error.

3.2   Gaussian Processes

Given an index set T, $\{X_t; t \in T\}$ is a Gaussian process [12,32] if for any finite set of indices $\{t_1, ..., t_n\}$ of T, $(X_{t_1}, ..., X_{t_n})$ is a multivariate normal random variable. In this paper, we are concerned primarily in the case where T indexes time, i.e. $T = \mathbb{R}^+$ or $\mathbb{Z}^+$, in which case $\{X_t; t \in T\}$ can be uniquely characterized by its mean and covariance functions

$$\mu(t) := \mathbb{E}\left[X_t\right], \tag{4}$$
$$R(s,t) := \mathbb{E}\left[X_t X_s\right], \quad \forall\, s, t \in T. \tag{5}$$

The following Gaussian processes are frequently encountered in stochastic models, e.g. in financial modeling [2,9], and the prior distributions employed in MGP-VAE will be the appropriately discretized versions of these processes.

Fractional Brownian Motion (fBM).   Fractional Brownian motion [24] $\{B_t^H; t \in T\}$ is a Gaussian process parameterized by a Hurst parameter $H \in (0, 1)$, with mean and covariance functions given by

$$\mu(t) = 0, \tag{6}$$
$$R(s,t) = \frac{1}{2}\left(s^{2H} + t^{2H} - |t-s|^{2H}\right), \quad \forall\, s, t \in T. \tag{7}$$

When $H = \frac{1}{2}$, $W_t := B_t^{\frac{1}{2}}$ is standard Brownian motion [12] with independent increments, i.e. the discrete sequence $(W_0, W_1, W_2, \ldots)$ is a simple symmetric random walk where $W_{n+1} \sim \mathcal{N}(W_n, 1)$.

Most notably, when $H \neq \frac{1}{2}$, the process is not Markovian. When $H > \frac{1}{2}$, the disjoint increments of the process are positively correlated, whereas when $H < \frac{1}{2}$, they are negatively correlated. We will demonstrate in our experiments how tuning H effects the clustering of the latent code.
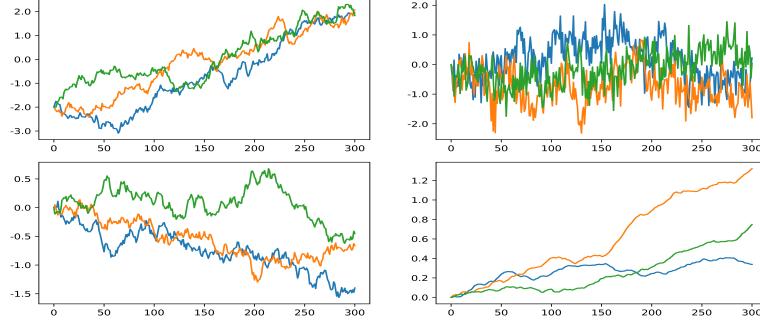
Fig. 2. Sample paths for various Gaussian processes. Top-left: Brownian bridge from -2 to 2; top-right: fBM with H = 0.1; bottom-left: standard Brownian motion; bottom-right: fBM with H = 0.9

Brownian Bridge (BB). The Brownian bridge [9,16] from $a \in \mathbb{R}$ to $b \in \mathbb{R}$ on the domain $[0, T]$ is the Gaussian process defined as

$$X_t = a\left(1 - \frac{t}{T}\right) + b\left(\frac{t}{T}\right) + W_t + \frac{t}{T}W_T. \tag{8}$$

Its mean function is identically zero and its covariance function is given by

$$R(s,t) = \min(s,t) - \frac{st}{T}, \quad \forall s, t \in T. \tag{9}$$

It can be also represented as the solution to the stochastic differential equation [16]

$$\mathrm{d}X_t = \frac{b - X_t}{T - t}\,\mathrm{d}t + \mathrm{d}W_t, \quad X_0 = a, \tag{10}$$

with solution

$$X_t = a\left(1 - \frac{t}{T}\right) + b\left(\frac{t}{T}\right) + (T - t)\int_0^t \frac{1}{T - s}\,\mathrm{d}W_s. \tag{11}$$

From (8), its defining characteristic is that it is pinned at the start and the end such that $X_0 = a$ and $X_T = b$ almost surely.

3.3   MGP-VAE

For VAEs in the unsupervised learning of static images, the latent distribution $p(z)$ is typically a simple Gaussian distribution, i.e. $z \sim \mathcal{N}(0, \sigma^2 \mathcal{I}_d)$. For a video sequence input $(x_1, \ldots x_n)$ with n frames, we model the corresponding latent code as

$$z = (z_1, z_2, \ldots, z_n) \sim \mathcal{N}(\mu_0, \Sigma_0), \quad z_i \in \mathbb{R}^d, \tag{12}$$

$$\mu_0 = \left[\mu_0^{(1)}, \ldots, \mu_0^{(d)}\right] \in \mathbb{R}^{n \times d}, \tag{13}$$

$$\Sigma_0 = \left[\Sigma_0^{(1)}, \ldots, \Sigma_0^{(d)}\right] \in \mathbb{R}^{n \times n \times d}. \tag{14}$$

Here d denotes the number of channels, where one channel corresponds to one sampled Gaussian path, and for each channel, $\left\{ \mu_0^{(i)}, \Sigma_0^{(i)} \right\}$ are the mean and covariance of

$$V + \sigma B_t^H, \quad t = \{1, \ldots, n\}, \tag{15}$$

in the case of fBM or

$$A\left(1 - \frac{t}{n}\right) + B\left(\frac{t}{n}\right) + \sigma\left(W_t + \frac{t}{n}W_n\right) \tag{16}$$

in the case of Brownian bridge. V, A are initial distributions, and B is the terminal distribution for Brownian bridge. They are set to be standard normal, and we experiment with different values for σ. The covariances can be computed using (6) and (9) and are not necessarily diagonal, which enables us to model more complex inter-frame correlations.

Rewriting z as $\left(z^{(1)}, \ldots, z^{(d)}\right)$, for each channel $i = 1, \ldots, d$, we sample $z^{(i)} \in \mathbb{R}^n \sim \mathcal{N}\left(\mu_0^{(i)}, \Sigma_0^{(i)}\right)$ by sampling from a standard normal ξ and computing

$$z^{(i)} = \mu_0^{(i)} + L^{(i)}\xi, \tag{17}$$

where $L^{(i)}$ is the lower-triangular Cholesky factor of $\Sigma_0^{(i)}$.

The output of the encoder is a mean vector μ_1 and a symmetric positive-definite matrix Σ_1, i.e.

$$q(z \mid x) \sim \mathcal{N}(\mu_1, \Sigma_1), \tag{18}$$

and to compute the KL divergence term in (2), we use the formula

$$D_{KL}\left[q \mid p\right] = \frac{1}{2}\left[\text{tr}\left(\Sigma_0^{-1}\Sigma_1\right) + \left\langle \mu_1 - \mu_0, \Sigma_0^{-1}(\mu_1 - \mu_0)\right\rangle - k + \log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right]. \tag{19}$$

Following [13], we add a β factor to the KL divergence term to improve disentanglement. We will describe the details of the network architecture of MGP-VAE in Section 4.

3.4 Video Prediction Network and Geodesic Loss Function

For video prediction, we predict the last k frames of a sequence given the first $n - k$ frames as input. To do so, we employ a simple three-layer MLP (16 units per layer) with ReLU activation which operates in latent space rather than on the actual frame data so as to best utilize the disentangled representations. The first $n - k$ frames are first encoded by a pre-trained MGP-VAE into a sequence of points in latent space. These points are then used as input to the three-layer MLP to predict the next point, which is then passed through MGP-VAE's decoder to generate the frame. This process is then repeated $k - 1$ more times.

Given an output z_0 and a target z_T, we use the geodesic distance between $g(z_0)$ and $g(z_T)$ as the loss function instead of the usual squared-distance $\|z_0 - z_T\|^2$. Here, $g : \mathbb{R}^{n \times d} \to M \subset \mathbb{R}^N$ is the differentiable map from the latent space to the data manifold M which represents the action of the decoder. We use the following algorithm from [27] to compute the geodesic distance.

Algorithm 1: Geodesic Interpolation

Input: Two points, $z_0, z_T \in Z$;
α, the learning rate
Output: Discrete geodesic path, $z_0, z_1, ..., z_T \in Z$
Initialize z_i as the linear interpolation between z_0 and z_T
while $\Delta E_{z_t} > \epsilon$ **do**
    for $i \in \{1, 2, ..., T-1\}$ **do**
        Compute gradient using (21)
        $z_i \leftarrow z_i - \alpha \nabla_{z_t} E_{z_t}$
    end for
end while

This algorithm finds the minimum of the energy of the path (and thus the geodesic)

$$E_{z_t} = \frac{1}{2} \sum_{i=0}^{T} \frac{1}{\delta t} \|g(z_{i+1}) - g(z_i)\|^2 \tag{20}$$

by computing its gradient

$$\nabla_{z_t} E_{z_t} = - \left(\nabla g(z_i)\right)^T \left[g(z_{i+1}) - 2g(z_i) + g(z_{i-1})\right]. \tag{21}$$

Algorithm 1 initializes $\{z_i\}$ to be uniformly-spaced points along the line between z_0 and z_T and gradually modifies them until the change in energy falls below a predetermined threshold. At this point, we use z_1 as the target instead of z_T as $z_1 - z_0$ is more representative of the vector in which to update the prediction z_0 such that the geodesic distance is minimized; see Figure 3 for an illustration.
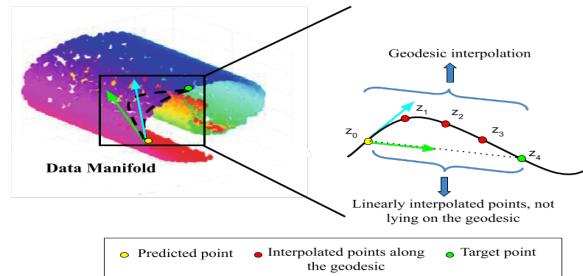

Fig. 3. Using the geodesic loss function as compared to squared-distance loss for prediction. By setting the target as z_1 instead of z_4, the model learns more efficiently to predict the next point.

4   Experiments

In this section, we present experiments which demonstrate MGP-VAE's ability to disentangle multiple factors of variation in video sequences.

4.1   Datasets

Moving MNIST[4] [29] comprises of moving gray-scale hand-written digits. We generate 60,000 sequences for training, each with a single digit moving in a random direction across frames and bouncing off edges.
Coloured dSprites is a modification of the dSprites[5] [13] dataset. It consists of 2D shapes (square, ellipse, heart) with 6 values for scale and 40 values for orientation. We modify the dataset by adding 3 variations for colour (red, green, blue) and constrain the motion of each video sequence to be simple horizontal or vertical motion.

For each sequence, the scale is set to gradually increase or decrease a notch in each frame. Similarly, after an initial random selection for orientation, the subsequent frames rotate the shape clockwise or anti-clockwise one notch per frame. The final dataset consists of a total of approximately 100,000 datapoints.
Sprites [26] comprises of around 17,000 animations of synthetically rendered animated caricatures. There are 7 attributes: body type, sex, hair type, arm type, armor type, greaves type, and weapon type, with a total of 672 unique characters. In each animation, the physical traits of the sprite remain constant while the pose (hand movement, leg movement, orientation) is varied.

4.2   Network Architecture and Implementation Details

For the encoder, we use 8 convolutional layers with batch normalization between each layer. The number of filters begins with 16 in the first layer and increases to a maximum of 128 in the last layer. An MLP layer follows the last layer, and this is followed by another batch normalization layer. Two separate MLP layers are then applied, one which outputs a lower-triangular matrix which represents the Cholesky factor of the covariance matrix of $q(z \,|\, x)$ and the other outputs the mean vector.

For the decoder, we have 7 deconvolutional layers, with batch normalization between each layer. The first layer begins with 64 filters and this decreases to 16 filters by the last layer. We use ELU for the activation functions between all layers to ensure differentiability, with the exception of the last layer, where we use a hyperbolic tangent function.

Table 1 lists the settings for the manually tuned hyperparameters in the experiments. All channels utilizing Brownian bridge (BB) are conditioned to start at -2 and end at 2.

---

[4] http://www.cs.toronto.edu/ nitish/unsupervised_video
[5] https://github.com/deepmind/dsprites-dataset

Table 1. Hyperparameter settings for all datasets

	Moving MNIST	Coloured dSprites	Sprites
Gaussian processes	Channel 1: fBM (H = 0.1) Channel 2: fBM (H = 0.9)	5 Channels of BBs	5 Channels of BBs
σ	0.25	0.25	0.25
β	2	2	2
Learning Rate	0.001	0.008	0.010
No. of epochs	200	120	150

4.3   Qualitative Analysis

Figure 4 shows the results from swapping latent channels in the Moving MNIST dataset, where we see that channel 1 (fBM($H = 0.1$)) captures the digit identity, whereas channel 2 (fBM($H = 0.9$)) captures the motion.
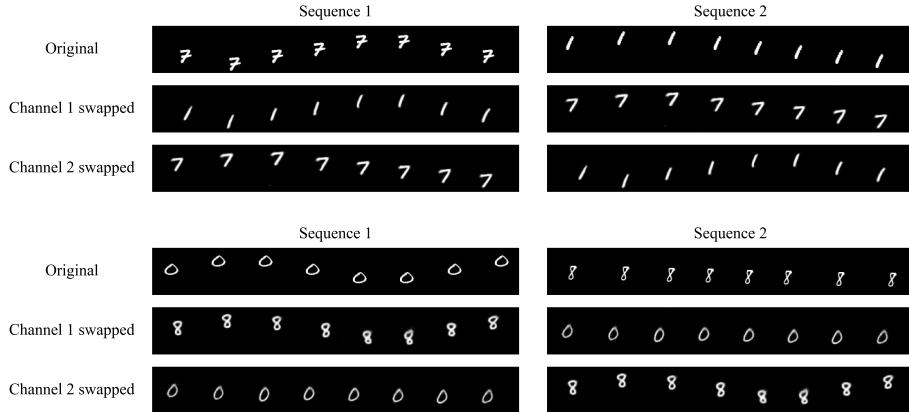

Fig. 4. Results from swapping latent channels in Moving MNIST; channel 1 (fBM($H = 0.1$)) captures digit identity; channel 2 (fBM($H = 0.9$)) captures motion.

Figure 5 gives a visualization of the latent space (here we use two channels with $H = 0.1$ and two channels with $H = 0.9$). In our experiments, we observe that fBM channels with $H = 0.9$ are able to better capture motion in comparison to setting $H = 0.5$ (simple-symmetric random walk, cf. [11]). We hypothesize that shifting the value of H away from that of the static channel sets the distributions apart and allows for better disentanglement.
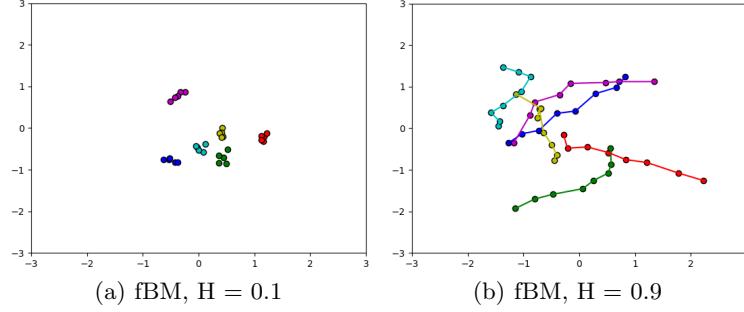
(a) fBM, H = 0.1   (b) fBM, H = 0.9

Fig. 5. Latent space visualization of fBM channels for 6 videos. Each point represents one frame of a video. The more tightly clustered points in (a) capture digit identity whereas the scattered points in (b) capture motion.

Fig. 6. Results from swapping latent channels in Sprites; channel 1 captures hair type, channel 2 captures armor type, channel 3 captures weapon type, and channel 4 captures body orientation.

Figures 6 and 7 show the results from swapping latent channels in the Sprites dataset and Coloured dSprites dataset respectively.
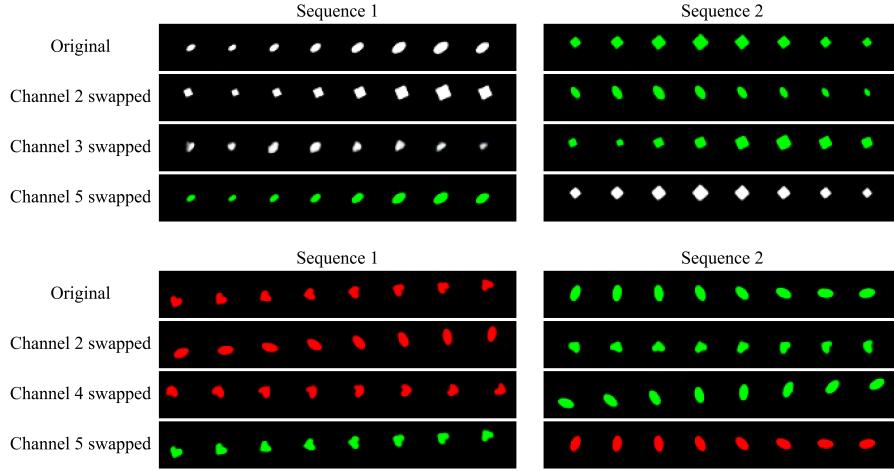
Fig. 7. Results from swapping latent channels in Coloured dSprites; channel 2 captures shape, channel 3 captures scale, channel 4 captures orientation and position, and channel 5 captures color.

Discussion.   The disentanglement results were the best for Moving MNIST, where we achieved full disentanglement in more than 95% of the cases. We were also able to consistently disentangle three or more features in Coloured dSprites and Sprites, but disentanglement of four or more features occurred less frequently due to their complexity.

We found that including more channels than the number of factors of variation in the dataset improved disentanglement, even as the extra channels did not necessarily encode anything new. For the Coloured dSprites and Sprites dataset, we originally experimented with different combinations of fBMs (with varying H) and Brownian bridges, but found that simply using 4-5 channels of Brownian bridges gave comparable results. We observed that with complex videos not easily separated into static or dynamic content, incorporating multiple Brownian bridge channels each with different start and end points led to good disentanglement. We hypothesize that anchoring the start and end points of the sequence at various places in latent space "spreads out" and improves the representation.

Finally, we also tested other Gaussian processes such as the Ornstein-Ulenbeck process [25] but as the results were not satisfactory, we shall defer a more detailed investigation to future work.

4.4   Evaluating Disentanglement Quality

We first evaluate the disentangled representations by computing the mean average precision of a k-nearest neighbor classification over labeled attributes in the Coloured dSprites and Sprites datasets.

Table 2. mAP values (%) for Coloured dSprites and Sprites

Model	Coloured dSprites						Sprites						
	Shape	Color	Scale	x-Pos	y-Pos	Avg.	Gender	Skin	Vest	Hair	Arm	Leg	Avg.
MCnet	95.6	94.0	69.2	69.7	70.2	79.7	78.8	70.8	76.6	80.2	78.2	70.7	75.9
DRNet	95.7	94.8	69.6	72.4	70.6	80.6	80.5	70.8	77.0	78.6	79.7	71.4	76.3
DDPAE	95.6	94.2	70.3	71.6	72.4	80.8	79.8	72.0	77.4	79.3	78.3	74.6	76.9
MGP-VAE	96.2	94.0	77.9	76.4	72.8	**83.4**	80.3	71.8	76.8	82.3	79.9	79.8	**78.5**

Table 2 shows that our model is able to capture multiple features more effectively than the baselines MCnet[6] [31], DRNet[7] [7] and DDPAE[8] [14].

Next, we use a non-synthetic benchmark in the form of a video prediction task to illustrate the improvement in the quality of MGP-VAE's disentangled representations. We train a prediction network with the geodesic loss function as outlined in Section 3.4, where we set the number of interpolated points to be four. In addition, to speed up the algorithm for faster training, we ran the loop in Algorithm 1 for a fixed number of iterations (10-15) instead of until convergence.

We compute the pixel-wise mean-squared-error and binary cross-entropy between the predicted k frames and the actual last k frames, given the first $n - k$ frames as input (n is set to 8 for Moving MNIST and Coloured dSprites, and set to 7 for Sprites). Tables 3 and 4 below summarize the results.

Table 3. Prediction results on Moving MNIST

	$k = 1$		$k = 2$	
Model	MSE	BCE	MSE	BCE
MCnet [31]	50.1	248.2	91.1	595.5
DRNet [7]	45.2	236.7	86.3	586.7
DDPAE [14]	35.2	201.6	75.6	556.2
Grathwohl, Wilson [11]	59.3	291.2	112.3	657.2
MGP-VAE	25.4	198.4	72.2	554.2
MGP-VAE (with geodesic loss)	**18.5**	**185.1**	**69.2**	**531.4**

The results show that MGP-VAE [9], even without using the geodesic loss function, outperforms the other models. Using the geodesic loss functions further lowers MSE and BCE. DDPAE, a state-of-the-art model in video disentanglement, achieves comparable results, although we note that we had to train the model considerably longer on the Coloured dSprites and Sprites datasets as compared to Moving MNIST to get the same performance.

Using the geodesic loss function during the training of the prediction network also leads to qualitatively better results. Figure 8 below shows that in a sequence

---

[6] https://github.com/rubenvillegas/iclr2017mcnet

[7] https://github.com/ap229997/DRNET

[8] https://github.com/jthsieh/DDPAE-video-prediction

[9] https://github.com/SUTDBrainLab/MGP-VAE

Table 4. Last-frame ($k = 1$) prediction results for Coloured dSprites and Sprites

Dataset	Coloured dSprites		Sprites	
Model	MSE	BCE	MSE	BCE
MCnet [31]	20.2	229.5	100.3	2822.6
DRNet [7]	15.2	185.2	94.4	2632.1
DDPAE [14]	12.6	163.1	75.4	2204.1
MGP-VAE	6.1	85.2	68.8	1522.5
MGP-VAE (with geodesic loss)	**4.5**	**70.3**	**61.6**	**1444.4**

with large MSE and BCE losses, the predicted point can generate an image frame which differs considerably from the actual image frame when the normal loss function is used. This is rectified with the geodesic loss function.

Fig. 8. Qualitative improvements from using the geodesic loss function: Left: without geodesic loss function; Right: with geodesic loss function; Top row: original video; Bottom row: video with the predicted last frame.

5   Conclusion

We introduce MGP-VAE, a variational autoencoder for obtaining disentangled representations from video sequences in an unsupervised manner. MGP-VAE uses Gaussian processes, such as fractional Brownian motion and Brownian bridge, as a prior distribution for the latent space. We demonstrate that different parameterizations of these Gaussian processes allow one to extract different static and time-varying features from the data.

After training the encoder which outputs a disentangled representation of the input, we demonstrate the efficiency of the latent code by using it as input to a MLP for video prediction. We run experiments on three different datasets and demonstrate that MGP-VAE outperforms the baseline models in video frame prediction. To further improve the results, we introduce a novel geodesic loss function which takes into account the curvature of the data manifold. This contribution is independent of MGP-VAE, and we believe it can be used to improve video prediction in other models as well.

For future work, we will continue to experiment with various combinations of Gaussian processes. In addition, enhancing our approach with more recent methods such as FactorVAE, β-TCVAE, or independent subspace analysis [30] may lead to further improvements.

References

1. Arvanitidis, G., Hansen, L.K., Hauberg, S.: Latent Space Oddity: on the Curvature of Deep Generative Models. In: International Conference on Learning Representations (ICLR) (2017)
2. Bayer, C., Friz, P., Gatheral, J.: Pricing under rough volatility. Quantitative Finance **16**(6), 887–904 (2016)
3. Bengio, Y., Courville, A.C., Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 1798–1828 (2012)
4. Casale, F.P., Dalca, A.V., Saglietti, L., Listgarten, J., Fusi, N.: Gaussian Process Prior Variational Autoencoders. In: Conference on Neural Information Processing Systems (NeurIPS) (2018)
5. Chen, R.T.Q., Li, X., Grosse, R., Duvenaud, D.: Isolating Sources of Disentanglement in VAEs. In: Advances in Neural Information Processing Systems (2018)
6. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In: Conference on Neural Information Processing Systems (NIPS) (2016)
7. Denton, E.L., Birodkar, V.: Unsupervised Learning of Disentangled Representations from Video. In: Conference on Neural Information Processing Systems (NIPS) (2017)
8. Fortuin, V., Rätsch, G., Mandt, S.: Multivariate Time Series Imputation with Variational Autoencoders. arXiv:1907.04155 (2019)
9. Glasserman, P.: Monte-Carlo Methods in Financial Engineering. Springer-Verlag, NY (2003)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
11. Grathwohl, W., Wilson, A.: Disentangling Space and Time in Video with Hierarchical Variational Auto-encoders. arXiv:1612.04440 (2016)
12. Hida, T., Hitsuda, M.: Gaussian Processes. American Mathematical Society (2008)
13. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In: International Conference on Learning Representations (ICLR) (2017)
14. Hsieh, J.T., Liu, B., Huang, D.A., Li, F.F., Niebles, J.C.: Learning to Decompose and Disentangle Representations for Video Prediction. Conference on Neural Information Processing Systems (NeurIPS) (2018)
15. Hyvarinen, A., Morioka, H.: Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 3765–3773. Curran Associates, Inc. (2016)
16. Karatzas, I., Shreve, S.E.: Brownian Motion and Stochastic Calculus. Springer-Verlag, NY (1998)
17. Khemakhem, I., Kingma, D., Monti, R., Hyvarinen, A.: Variational Autoencoders and Nonlinear ICA: A unifying framework. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 2207–2217. PMLR, Online (26–28 Aug 2020)

18. Kim, H., Mnih, A.: Disentangling by Factorising. In: International Conference on Machine Learning (ICML) (2018)
19. Kim, M., Wang, Y., Sahu, P., Pavlovic, V.: Bayes-Factor-VAE: Hierarchical Bayesian Deep Auto-Encoder Models for Factor Disentanglement. arXiv:1909.02820 (2019)
20. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: International Conference on Learning Representations (ICLR) (2013)
21. Kühnel, L., Fletcher, T.E., Joshi, S.C., Sommer, S.: Latent Space Non-Linear Statistics. arXiv:1805.07632 (2018)
22. Li, Y., Mandt, S.: Disentangled Sequential Autoencoder. In: International Conference on Machine Learning (ICML) (2018)
23. Lim, N., Gong, T., Cheng, L., Lee, H.K., et al.: Finding distinctive shape features for automatic hematoma classification in head CT images from traumatic brain injuries. International Conference on Tools with Artificial Intelligence (ICTAI) (2013)
24. Mandelbrot, B.B., Van Ness, J.W.: Fractional brownian motions, fractional noises and applications. SIAM Review **10**(4), 422–437 (1968)
25. Ornstein, L.S., Uhlenbeck, G.E.: On the theory of Brownian motion. Physical Review **36**, 823–841 (1930)
26. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep Visual Analogy-Making. In: Conference on Neural Information Processing Systems (NIPS) (2015)
27. Shao, H., Kumar, A., Fletcher, P.T.: The Riemannian Geometry of Deep Generative Models. In: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018)
28. Shukla, A., Uppal, S., Bhagat, S., Anand, S., Turaga, P.K.: Geometry of Deep Generative Models for Disentangled Representations. arXiv:1902.06964 (2019)
29. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised Learning of Video Representations using LSTMs. In: International Conference on Machine Learning (ICML) (2015)
30. Stuehmer, J., Turner, R., Nowozin, S.: Independent Subspace Analysis for Unsupervised Learning of Disentangled Representations. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 1200–1210. PMLR, Online (26–28 Aug 2020)
31. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing Motion and Content for Natural Video Sequence Prediction. International Conference on Learning Representations (ICLR) (2017)
32. Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning, vol. 2. MIT press Cambridge, MA (2006)