Hamiltonian prior to Disentangle Content and Motion in Image Sequences

Asif Khan
School of Informatics
University of Edinburgh
asif.khan@ed.ac.uk

Amos Storkey
School of Informatics
University of Edinburgh
a.storkey@ed.ac.uk

Abstract

We present a deep latent variable model for high dimensional sequential data. Our model factorises the latent space into content and motion variables. To model the diverse dynamics, we split the motion space into subspaces, and introduce a unique Hamiltonian operator for each subspace. The Hamiltonian formulation provides reversible dynamics that learn to constrain the motion path to conserve invariant properties. The explicit split of the motion space decomposes the Hamiltonian into symmetry groups and gives long-term separability of the dynamics. This split also means representations can be learnt that are easy to interpret and control. We demonstrate the utility of our model for swapping the motion of two videos, generating sequences of various actions from a given image and unconditional sequence generation.

1 Introduction

The ability to learn to generate artificial image sequences has diverse uses, from animation, key frame generation, summarisation to restoration and has been explored in previous work over many decades [1, 2, 3, 4, 5]. However, learning to generate arbitrary sequences is not enough; to provide useful value, the user must be able to have control over aspects of the sequence generation, such as the motion being enacted, or the characteristics of the agent doing an action. To enable this, we must be able to learn to decompose image sequences into content and motion characteristics such that we can apply learnt motions to new objects or vary the types of motions being applied.

The dynamical processes creating the evolution of image sequences are highly constrained. Consider the simplistic case of a person walking in a scene with a camera moving around that individual. The walking pose will return to similar positions periodically, and likewise, the revolving camera will revisit previous positions. Even without strict periodicity, many dynamical processes are reversible. Any time that a dynamic could conceivably return to an earlier position suggests an implicit conservation law—the conservation of information in the underlying scene generator as it must be capable of returning to and regenerating the same scene with non-negligible probability. The critical observation we wish to capture in this paper is that understanding the conservation occurring in the context of a set of sequences is a vital ingredient to decompose content from motion.

Given any conserved quantity, any motion must be modelled in a way that maintains the conserved quantity. In physics, such a motion is called a Hamiltonian motion; it keeps the corresponding Hamiltonian function constant. Hence we argue that a flexible latent Hamiltonian model allows us to learn a representation that enables conservation of the right quantities (which are themselves learnt) and models the dynamic evolution. This is mathematically equivalent to learning to represent the underlying motions as combinations of differentiable symmetry groups; all differentiable symmetry transformations follow a conservation law [6].
Identifying fundamental symmetries is essential for developing expressive deep generative models (DGMs) that understand the motion constraints and can generalise beyond the training data [7, 8]. Existing DGMs [9, 10, 11] for sequences using RNNs suffer because they lack such constraint, and accumulate errors as the length of the sequence grows, ending up deviating from the relevant path [12, 13, 14, 15].

In this paper, we intimate the more general applicability of latent Hamiltonian model; previous applications have been limited to fairly constrained physical systems. We propose a VAE framework to model the dynamics of sequences using a collection of Hamiltonian operators. Specifically, for any motion sequence, we model the transition from time step $t$ to step $t + 1$ using a group action of a Hamiltonian operator. The evolution of the dynamics of a sequence leaves certain information unchanged, identified as content, and certain properties that evolve in conjunction known as motion. Since different image sequences involve the evolution of various actions, we split the motion space into subspaces where each subspace is to model a unique action and is unaffected by other actions. We focus in this paper on a discrete, identified set of actions, that we can then compose at generation time. However this could be extended to learning from action compositions themselves in further work. Our main contributions are as follows,

1. We propose learnable Hamiltonian operators which associate conserved quantities with latent dynamics. In contrast to existing Hamiltonian approaches, usually restricted to constrained physical systems, we extend the formulation to more natural image sequences. Furthermore, the explicit form of operator imposes the structure in the motion latent space. In this way, we simultaneously learn a structured representation along with symmetry transformations that act on the space.

2. The high dimensional sequences are composed of various motions; therefore, we model each action in a separate subspace that further allows the separability of dynamics. It reduces the computational cost since the Hamiltonian of the whole space is now in a block diagonal form where each block is a Hamiltonian of a symmetry subgroup.

3. We empirically demonstrate several benefits of our model, i) generation of diverse dynamics from a starting frame, ii) useful notion of disentanglement where the content representation is separated from motion representation and where each motion subspace only controls a single action, iii) we demonstrate its use for motion swapping as well as unconditional sequence generation.

2 Related Work

**Hamiltonian Neural Networks** In recent years, several deep learning (DL) methods have been proposed to learn the dynamics of physical systems using Hamiltonian mechanics. In [16], authors use NNs to predict Hamiltonian from phase-space coordinates $s = (p, q)$ and their derivatives. A similar work [17] used NNs to discover symmetries of Hamiltonian mechanical systems. Much recently, Hamiltonian NNs are used for simulating complex physical systems [18, 19]. The key idea of this work is to represent the states of particles as a graph and use a graph neural network (GNN) to predict the change from the current state to the next state. In a follow-up [20], authors introduce sparsity on the messages in a graph and use the symbolic regression method to search for physical laws that describe the messages in the graph. In a very recent work, Hamiltonian generative network (HGN) [8] proposed to learn Hamiltonian from image sequences. HGN maps a sequence to a latent representation and then projects it to the phase space to unroll the dynamics using an ODE integrator with Hamilton’s equation. In another work [14] use second-order ODE parameterised as a BNN for modelling dynamics of high dimensional sequence data in the latent space of VAE. Most of the developments are built on neural ODE [21] an idea to view layers of NNs as internal states of an ODE. These methods rely on the numerical integration scheme and the stability of the ODE solver. A Hamiltonian formalism is an additional requirement that the dynamics of an ODE should be volume-preserving and reversible.

**Latent Space Models** There is a long history of latent state space models for modelling sequences [22, 23, 24, 25, 26]. Much recently, these methods are combined with DGMs for generating high dimensional sequences as well as learning disentangled representation [12, 9, 27, 28, 29]. MoCoGAN [9] developed an adversarial framework of combining a random content noise with a sequence of random motion noise to generate videos. More recently, DSVAE [27] proposed to split latent space into time-variant and invariant representation and use LSTM [30] to model the prior on time-variant representation. S3VAE [29] improves disentanglement of DSVAE by minimising a mutual information loss between content and motion variables. Some Hamiltonian methods [8, 14]
We propose a deep latent variable model for sets of sequential image data. Each set of sequences within a set may have very different content, but same correspondence of dynamic. E.g. in sprites we use temporal convolution with a kernel size $w$ vector denoting the action associated with the $i$th image sequence, with $x^i_k$ denote the $i$th image in the sequence. Let $u^i$ be an indicator vector denoting the action associated with the $i$th sequence; i.e. $u^i_k = 1$ iff sequence $i$ follows action $k$ and $u^i_k = 0$ otherwise. These sequences and corresponding actions are collected into a dataset $\{(x^i_{1:T}, u^i)\}_{i=1}^N$ of size $N$, where, for the sake of simplicity, we assume they all are of same length $T$.

In this paper, we use a latent space to aid modelling of each sequence and decompose that latent space into two parts, which we call a content space (denoted by $Z$) and a motion space (denoted
by \( \mathbf{S} \). As the data comprises sequences of various actions that take different dynamical form, we further decompose the latent motion space \( \mathbf{S} = \mathbf{S}^1 \oplus \mathbf{S}^2 \oplus \ldots \oplus \mathbf{S}^K \), with one part for each action. In modelling a sequence corresponding to action \( k \), only the subspace \( \mathbf{S}^k \) will be allowed to change across the length of that sequence. Each motion subspace is further decomposed into generalised position and momentum parts; \( \mathbf{S}^k = (\mathbf{p}^k, \mathbf{q}^k) \). Only the position part of this latent space is used generatively to create an individual image. The momentum part only affects the dynamics.

Critical to this work, this decomposition makes it straightforward to define learnable Hamiltonian mechanics to model the dynamic process. This Hamiltonian model provides many advantages; it prevents the neural network from leaking constant content information via the motion representation, and it ensures the possibility of preserving key conservation quantities that must be implicit there are many motion dynamics. This is discussed further in Section 3.1.1. The full framework of our model is illustrated in Figure 1. In the next section, the generative model will be introduced, followed by the variational formalism for inference and the learning.

### 3.1 Generative Model

For completeness we first present the full probabilistic model in (3-6) before describing each component. The dynamic is defined in terms of an action \( \mathbf{u} \); the model is conditioned on this action vector. First, in (3), we sample the content variable \( \mathbf{z} \) from a prior \( p(\mathbf{z}) \). The content variable will describe the constant appearance characteristics expressed through the whole sequence. Next we sample a starting position from prior \( p(\mathbf{q}_1^k) \), and a momentum from a prior \( p(\mathbf{p}_1^k) \). The full state-space representation is given by \( \mathbf{s}_1^k = (\mathbf{p}_1^k, \mathbf{q}_1^k) \). We then use our dynamical model in (8) to trace out the forward trajectory in the phase space. Finally, we combine the position trajectory with the content representation and use a decoder neural network to get the emission distribution of sequence in the data space. In summary,

**GIVEN:** \( k \) denoting action label for a sequence,

1. \( \mathbf{z} \sim p(\mathbf{z}) \) 
2. \( \mathbf{q}_1^k \sim \mathcal{N}(0, \mathbf{I}_d), \quad \mathbf{p}_1^k \sim \mathcal{N}(0, \mathbf{I}_d), \quad \mathbf{s}_1^k = [\mathbf{p}_1^k, \mathbf{q}_1^k] \) 
3. \( \mathbf{s}_t^k = f(\mathbf{s}_{t-1}^k, \omega_k, t), \quad \forall t > 1 \) 
4. \( \mathbf{s}_t^{k'} = 0, \quad \forall t, k' \neq k \) 
5. \( \mathbf{q}_t = [\mathbf{q}_1^k, \ldots, \mathbf{q}_K^k], \quad \mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t|\phi(\mathbf{z}, \mathbf{q}_t), \alpha^2 \mathbf{I}_m), \quad \forall t \)

where \( d \) is the dimensionality of \( k^{th} \) subspace, \( m \) is the dimensionality of data space, \( f \) is a dynamical model \( \mathbf{H} \) and \( \omega_k \) are its parameters corresponding to the \( k^{th} \) subspace. The emission distribution is a spherical Gaussian, with a parameterised mean \( \phi(\cdot, \cdot) \), and a spherical covariance \( \alpha^2 \mathbf{I}_m \). In our work, we choose \( \alpha = 1 \). We have provided probabilistic graph of our GM in Appendix 1.

#### 3.1.1 Dynamical Model

In image sequences, we can view each frames of a sequence as a point in some representation space; the temporal dynamics trace a path connecting the frames forming a 1-submanifold of the image manifold. Most dynamical models either try to capture this structure deterministically \[43\] or probabilistically \[44, 45, 27\] via linear or non-linear state-space models. In either case, small errors in dynamical steps can accumulate and result in a significant deviation from the manifold when unrolling long-term trajectories at inference time \[12, 13\]. Interestingly, Hamiltonian systems associate conserved quantities with the motions and, we argue, alleviate some of these issues by constraining the dynamics to be reversible and volume-preserving. In our work, without significant loss of generality, we propose a linear Hamiltonian system in the latent layer, relying on the nonlinear neural network mapping to data space to handle all nonlinear aspects. This also enhances the interpretability of the dynamics.

**Definition 1.** A matrix \( \mathbf{H} \in \mathbb{R}^{2d \times 2d} \) is an Hamiltonian matrix if \( \mathbf{H}^T \mathbf{J} \mathbf{H} = \mathbf{J} \), where \( \mathbf{J} \) is a skew-symmetric matrix \( \mathbf{J} = \begin{pmatrix} 0 & \mathbf{I}_d \\ -\mathbf{I}_d & 0 \end{pmatrix} \).

Consider a coordinate vector \( \mathbf{s} \in \mathbb{R}^{2d} \) in the phase space \( \mathbf{S} \) that evolves under Hamiltonian energy \( \mathbf{E} \),

\[
\mathbf{E} = \frac{1}{2} \mathbf{s}^T \mathbf{M}(t) \mathbf{s}
\]
where $\mathbf{M}(t)$ is a symmetric matrix and $\mathbf{s}$ is a coordinate in phase space at a time $t$. In Hamiltonian mechanics, the coordinates are specified in terms of position $\mathbf{q}$ and momentum $\mathbf{p}$ variables as $\mathbf{s} = (\mathbf{q}, \mathbf{p})$. Then the equation of motion is given by $\frac{d\mathbf{s}(t)}{dt} = \mathbf{J}\mathbf{M}(t)\mathbf{s}$. Let $\mathbf{H}(t) = \mathbf{J}\mathbf{M}(t)$, we can rewrite the equation of motion as $\frac{d\mathbf{s}(t)}{dt} = \mathbf{H}(t)\mathbf{s}$.

For a time-invariant Hamiltonian we can obtain the solution of the system in the closed-form using matrix exponential $\mathbf{s}(t) = e^{t\mathbf{H}}\mathbf{s}(0)$. The matrix exponent has a connection to Lie algebras, and for small $t$ we can interpret $e^{t\mathbf{H}}$ as an infinitesimal transformation of $\mathbf{s}(0)$ under the group action. We discuss this further in Appendix A.2. For a detailed introduction to the topic we refer to [46].

In order to learn the model parameters, we need to infer the distribution over latent variables; we discuss this further in Appendix A.2. For a more comprehensive overview, we refer readers to [47].

In this work, we consider $K$ Hamiltonians of the form $\mathbf{H}_1, \cdots, \mathbf{H}_K$, each potentially acting on separate parts of the state space $\mathbf{S}^1, \mathbf{S}^2, \cdots \mathbf{S}^K$. To unroll the trajectory of a sequence $i$ with associated motion $k$, we use the group action defined by the matrix exponent of the operator $\mathbf{H}_k$ on a starting phase space representation $\mathbf{s}^k_1 \in \mathbf{S}^k$ given by,

$$\mathbf{s}^k_t = f(\mathbf{s}^k_{t-1}; \omega_k, t) = e^{t\mathbf{H}_k}\mathbf{s}^k_t, \quad \forall t > 1 \quad (8)$$

$$\mathbf{s}^k_t = 0, \quad \forall t, k' \neq k. \quad (9)$$

The backward dynamics can simply be obtained by negating time $-t$. We assume all time steps are equally spaced. The above formulation provides an explicit disentanglement of the motion space. It further allows us to parallelise the computation of matrix exponential by leveraging the block diagonal form of $\mathbf{H}$. To make use of the structure of Hamiltonians, we consider the group of real Hamiltonian matrices that form a Symplectic Lie group under multiplication $Sp(2d)$ with $2d^2 + d$ independent elements. We also look at the symplectic orthogonal group $SpO(2d)$ that further restricts Hamiltonians to skew-symmetric form with $(d^2 - d)/2$ independent elements. We briefly introduce it in Appendix A.2. For a more comprehensive overview, we refer readers to [47].

### 3.2 Inference

In order to learn the model parameters, we need to infer the distribution over latent variables; we follow a variational formalism that provides following evidence lower bound (ELBO),

$$\max_q \mathbb{E}_{q(z, \mathbf{x}_{1:T} | \mathbf{x}_{1:T}, u)} \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}, \mathbf{s}_{1:T} | u)}{q(\mathbf{z}, \mathbf{s}_{1:T} | \mathbf{x}_{1:T}, u)} \quad (10)$$

where $q(\cdot)$ is the approximate posterior distribution and $\mathbf{s}_i = [\mathbf{q}_i, \mathbf{p}_i]$. It remains to define the approximate posterior we use. Since the Hamiltonian dynamics are reversible in time, at inference time, we randomly sample a choice of frame $t$ and use forward and backward dynamics to trace the trajectory of states after and before that frame. For a sequence $\mathbf{x}_{1:T}$, we use the process in (11)-(14) to draw samples from a variational distribution $q(\mathbf{z}, \mathbf{s}_{1:T} | \mathbf{x}_{1:T}, u)$. Simply, we sample the content variable $\mathbf{z}$ conditioned on the observed data and independently sample the motion states $\mathbf{s}^k_t = [\mathbf{q}^k_t, \mathbf{p}^k_t]$ for the reference frame $t$ conditioned on the observed data and the relevant action $k$. Motion states corresponding to other actions are set to zero. The motion states for all the other frames are then created from the forward and backward application of the Hamiltonian motion for the relevant action.

In equations, this is,

$$\mathbf{z} \sim q(\mathbf{z} | \mathbf{x}_{1:T}), \quad t \sim \mathcal{U}(\{1, \ldots, T\}) \quad (11)$$

$$\mathbf{q}^k_t \sim q(\mathbf{q}^k_t | \mathbf{x}_{t}, u), \quad \mathbf{p}^k_t \sim q(\mathbf{p}^k_t | \mathbf{x}_{t-w}, u), \quad \mathbf{s}^k_i = [\mathbf{q}^k_i, \mathbf{p}^k_i] \quad (12)$$

$$\mathbf{s}^k_{t+1} = f(\mathbf{s}^k_t; \omega_k, t), \quad \mathbf{s}^k_{t-1} = f(\mathbf{s}^k_t; \omega_k, -t), \quad \forall t \quad (13)$$

$$\mathbf{s}^k_t = 0, \quad \forall t, k' \neq k. \quad (14)$$

where $t$ is a starting index, $q(\mathbf{q}^k_t | \mathbf{x}_{t}, u)$, is the posterior distributions of $k^{th}$ position subspace conditioned on the frame $\mathbf{x}_t$, $q(\mathbf{p}^k_t | \mathbf{x}_{t-w})$ is the posterior distributions of $k^{th}$ momentum subspace conditioned on $w$ previous frames and $q(\mathbf{z}^k | \mathbf{x}^k_{1:T})$ is the posterior distribution of the content space conditioned on the entire sequence. We parameterise the factorised posterior as a spherical Gaussian distribution,

$$q_{\theta_1 \theta_2}(\mathbf{z} | \mathbf{x}_{1:T}) = \mathcal{N}(\mathbf{z} | \mu_z, \sigma^2_z I), \quad q_{\theta_k \theta_2}(\mathbf{q}^k_t | \mathbf{x}_t, u) = \mathcal{N}(\mathbf{q}^k_t | \mu_{q^k}, \sigma^2_k I), \quad (15)$$

$$q_{\theta_1 \theta_2}(\mathbf{p}^k_t | \mathbf{x}_{t-w}, u) = \mathcal{N}(\mathbf{p}^k_t | \mu^k, \sigma^2_k I) \quad (16)$$
where \( \{ \theta_1, \theta_2, \gamma_k, \delta_k \} \) are parameters of encoder neural network. The parameters \( \theta_1 \) are specific to the content network, \( \gamma_k \) of the network mapping to the \( k^{th} \) position subspace, \( \delta_k \) of the network mapping to the \( k^{th} \) momentum subspace and \( \theta_2 \) are shared parameters of content and motion network. We use reparametrisation trick \(^\text{[43]}\) to sample from latent distribution \( z = \mu + \sigma \odot \epsilon \) where \( \epsilon \sim \mathcal{N}(0, 1) \).

### 3.3 Learning Objective

The final learning problem reduces to the optimisation of the following objective function,

\[
\max_{\theta_1, \theta_2, \phi, \omega_k, \gamma_k, \delta_k} - KL[q(q_t^k | x_t, u) || p(q_t^k)] - KL[q(p_{t,t-u}^k | x_t, u) || p(p_t^k)]
- KL[q(z | x_{1:T}, u) || p(z)] + \mathbb{E}_{q(q_t^k | x_t, u)} \left[ \sum_k \log p(x_t | q_{t-k}, z) \right].
\]

(17)

We have provided the derivation in the Appendix \( \text{[A.1]} \).

### 4 Experiments

We conduct experiments on the following video datasets, i) **Sprites** a sequence of animated character performing different actions as per sprites sheets\(^\text{[1]}\). It comprises three actions: ‘walking’, ‘spell cast’ and ‘slashing’ from three viewing angles: ‘left’, ‘right’ and ‘straight’. The sequences are of length 8 with each frame as an RGB image of size 64 × 64 × 3. The appearance of each character has four attributes: colour of skin, hairstyle, tops and trousers/pants. Each attribute can take six values resulting in 1296 unique characters. We used 1000 characters for training and the rest for evaluation. ii) **MUG** \(^\text{[49]}\) is a dataset of 52 individual performing six facial expressions: anger, disgust, fear, happiness, sadness and surprise. The dataset is made available by signing the license agreement available. The dataset consists sequences of variable length ranging from 50 to 160 frames. For training purpose, we downsample the sequences by a factor of two and use random subsequence of length 8, crop face region and resize it to 64 × 64. The training and evaluation splits are based on \([9]\).

#### 4.1 Results and Discussion

We perform the qualitative and quantitative analysis of all the models. We investigate two choices of structure for the Hamiltonian matrices. We refer to the symplectic group structure as \( H \) and the symplectic orthogonal group as skew-\( H \). To compute the matrix exponential, we use fast Taylor approximation \(^\text{[50]}\) that provides a stable solution under various matrix norms.

**Quantitative Evaluation** We evaluate our model on sequence generation as well as the disentanglement of representations. As a first step we evaluate the generation of two operators \( H \) and skew-\( H \) using the per-frame structural similarity index measure (SSIM) peak signal to noise ratio (PSNR) and mean squared error (MSE). We generate a longer sequence of length 16 (twice the length used for training purpose) conditioned on the starting frame. The sprites consists of periodic sequences of length 8 where the start and end frames are identical, in this case we duplicate the sequence to get a ground truth of length 16. For MUG, we draw sequence of length 16 from the evaluation set. The SSIM scores are between −1 and 1, with a more significant score indicating more similarity between the ground truth and generated sequence. Likewise, higher PSNR and lower MSE implies better generation. Table 1 describes the performance of our model on generation task under different scores. The scores show our model can generate high-quality sequences from an input image. For rest of the paper, we consider only \( H \) as it consistently works better at sequence generation. We compare its performance with the state-of-the-art baselines for sequence disentanglement, namely DSVAE \(^\text{[27]}\), MoCoGAN \([9]\) and S3VAE \([29]\).

To evaluate the disentanglement, we use a classifier pre-trained on the task of action prediction, to evaluate the generated sequences. To begin with, we draw a starting position and momentum from a prior distribution and use a dynamical model to unroll the trajectory in the phase space. Next, we sample the content variable \( z \) from real sequences and combine it with position variables to generate the sequence. We report the performance of the classifier in predicting the action from these generated sequences. 

\(^1\) https://github.com/jrconway3/Universal-LPC-spritesheet
We pre-train a classifier on the task of identity prediction and use it for evaluating the generated sequences. We further evaluate our model in preserving the identity of sequences. The identity for sprites is described in terms of four different attributes, and for MUG [49] it refers to the identity of the person.

Table 1: Evaluation for sequence generation. For SSIM and PSNR higher is better; for MSE lower is better. We can see both choices of operator can generate sequences close to the ground truth.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>MSE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Sprites</td>
<td>0.982 ± 0.005</td>
<td>36.76 ± 1.096</td>
<td>0.0005 ± 0.0002</td>
</tr>
<tr>
<td>H</td>
<td>MUG</td>
<td>0.797 ± 0.003</td>
<td>24.49 ± 0.099</td>
<td>0.0040 ± 0.0001</td>
</tr>
<tr>
<td>Skew-H</td>
<td>Sprites</td>
<td>0.950 ± 0.021</td>
<td>33.88 ± 2.03</td>
<td>0.0026 ± 0.0012</td>
</tr>
<tr>
<td>Skew-H</td>
<td>MUG</td>
<td>0.791 ± 0.003</td>
<td>24.25 ± 0.094</td>
<td>0.0044 ± 0.0001</td>
</tr>
</tbody>
</table>

We can see on MUG our model can preserve the identity with high accuracy. We can make a qualitative evaluation of disentanglement and diversity of generated samples.

| Method   | Data  | Accuracy↑ | H(y|x)↓ | H(y)↑ |
|----------|-------|-----------|--------|-------|
| Ours H   | MUG   | 0.929     | 0.108  | 1.778 |
| DSVAE [27] | Sprites | 0.543     | 0.374  | 1.657 |
| MoCoGAN [9] | Sprites | 0.631     | 0.183  | 1.721 |
| S3VAE [29] | Sprites | 0.705     | 0.135  | 1.760 |

(b) Performance of our model in terms of accuracy of individual attributes in sprites and identity of actors in MUG dataset. This shows our model can preserve content when the motion representation is changed.

We further evaluate our model in preserving the identity of sequences. The identity for sprites is described in terms of four different attributes, and for MUG [49] it refers to the identity of the person. We pre-train a classifier on the task of identity prediction and use it for evaluating the generated sequences. This evaluation gives us a measure of the model’s ability to keep the identity when the motion is changed. For sprites, we report the accuracy of individual attributes. Results are outlined in Table 2. We can see on MUG our model can preserve the identity with high accuracy. We can make a similar observation for different attributes of sprites sequences. Thus, the good performance indicates that the content is preserved when traversing the motion subspace, and the motion space is invariant when changing the content variables. This is also reflected in the qualitative results.

Qualitative Evaluation For the qualitative analysis, we report results with the Hamiltonian H. We observed the skew-H resulted in similar performance; we omit it here due to limited space. To begin with, Figure 2 gives an example of original, reconstruction and generated sequences. We generate a sequence by applying the motion operator on the latent encoding of the first time step. On the left are the results for sprites and on the right of MUG video sequences. To evaluate disentanglement, we use the model for the task of motion swapping. We start by encoding two sequences \(x_{1:T}^1\) and \(x_{1:T}^2\) to their latent representations \((z^1, q_{1:T}^1)\) and \((z^2, q_{1:T}^2)\), next we swap the motion variables \((z^1, q_{1:T}^1)\) and \((z^2, q_{1:T}^2)\) between the two representation spaces, and then pass the resulting representations through the decoder to generate the sequences \(x_{1:T}^{1→2}\) and \(x_{1:T}^{2→1}\). Figure 3 shows the result of this. On the left, the pair of consecutive rows are of original sequences and on the right of the sequences generated by swapping the motion representations. We can see swapping the motion part does not affect the identity of the sequences.
Figure 2: In each patch, the first row is the original sequence, the second row is its reconstruction, and third row is a sequence generated by an action of the operator on the phase space representation of the starting time step. On left are results on sprites and on right on MUG. The reconstruction shows our model can learn good representations and the generation shows the dynamical operator can generate realistic motions from a starting frame.

Figure 3: Left: original sequence pairs; right: reconstructions after swapping the motion variables in row pairs. The content space is disentangled from the motion space.

To further investigate the generation quality of different motion operators, we use our model for an image to sequence generation. We first encode the image to its latent space representation. Next, we obtain its representation in the different motion spaces and use the respective operators to unroll the trajectories in phase space for each different motion, which are then combined with content and transformed to the image space using a decoder network. The left side of Figure 4 shows examples of decoding different motion from the same input image. For sprites, the actions are in order ‘walk’, ‘spell card’, ‘slash’ and for MUG they are ordered ‘anger’, ‘disgust’, ‘fear’, ‘happiness’, ‘sadness’ and ‘surprise’. We can see the visual dynamics associated with all the operators are well separated from one another. More results are presented in Appendix A.3.3, where it is apparent that longer-term sequences maintain the consistency associated with the content-motion pair.

5 Conclusions and Future Work

We introduced a DGM that uses Hamiltonian operators in the latent space to model the manifold of various motions. We decompose the latent space into content and motion subspace, and for each action, associate a unique partition in the motion subspace. The dynamics in each partition are unrolled using a Hamiltonian operator. Our formulation, associates conserved quantities with the
Figure 4: On left (a) and (c) are examples of image to sequence generated by an action of Hamiltonian operator on the phase space representation of the starting frame. We can observe all dynamics are well separated that demonstrates the disentanglement of various actions. On right (b) and (c) are examples of unconditional generation.

dynamics; we empirically show it provides a helpful notion of disentanglement for image sequence data. We demonstrate the performance on several tasks such as image-to-sequence, motion swapping and unconditional generation. The main advantage of our model is we can generate long term trajectories and traverse the motion manifolds of different actions in the latent space. We look forward to future applications to other sequential data types, including music audio and speech.

One potential limitation associated with the presented approach is that it is rigid in its form since we explicitly restrict every motion to a unique subspace that requires prior knowledge of the action label of a sequence. In future work we will address this by allowing each motion in the data to be composed from primitive motion components via a prior mixing distribution. Another limitation of our model is that it is less able to deal with irregularly sampled sequences, changes in tempo or reversals. In future work, this will be addressed by allowing a more flexible prior on the spacing between time steps. Nevertheless, all such approaches depend critically on the decomposition demonstrated here, and we argue that this work has demonstrated the significant promise of content and Hamiltonian motion decomposition of image sequences.

Acknowledgments and Disclosure of Funding

The authors would like to thank Joseph Mellor and William Toner for the helpful discussion during the project and Elliot J. Crowley for the valuable feedback on the paper. This research was funded in part by an unconditional gift from Huawei Noah’s Ark Lab, London.

References


