VDSM: Unsupervised Video Disentanglement with State-Space Modeling and Deep Mixtures of Experts

Matthew J. Vowels
m.j.vowels@surrey.ac.uk

Necati Cihan Camgoz
n.camgoz@surrey.ac.uk

Richard Bowden
r.bowden@surrey.ac.uk

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK

Abstract

Disentangled representations support a range of down-stream tasks including causal reasoning, generative modeling, and fair machine learning. Unfortunately, disentanglement has been shown to be impossible without the incorporation of supervision or inductive bias. Given that supervision is often expensive or infeasible to acquire, we choose to incorporate structural inductive bias and present an unsupervised, deep State-Space-Model for Video Disentanglement (VDSM). The model disentangles latent time-varying and dynamic factors via the incorporation of hierarchical structure with a dynamic prior and a Mixture of Experts decoder. VDSM learns separate disentangled representations for the identity of the object or person in the video, and for the action being performed. We evaluate VDSM across a range of qualitative and quantitative tasks including identity and dynamics transfer, sequence generation, Fréchet Inception Distance, and factor classification. VDSM achieves state-of-the-art performance and exceeds adversarial methods, even when the methods use additional supervision.

1. Introduction

In general, humans are able to reason about the identity of an object and the object's motion independently, thereby implying that identity and motion are considered as disentangled generative attributes [38, 9]. In other words, a change to an object's motion does not affect the object's identity. For example, in sign language translation, the canonical form of a gesture exists independently of the identity or appearance of the signer. In order to reason independently about the latent factors underlying identity and motion, it is therefore desirable to seek disentanglement.

Achieving disentangled representations is a long standing goal for machine learning, and supports causal reasoning [72, 10, 90, 97], fair machine learning [63, 66, 20, 99],
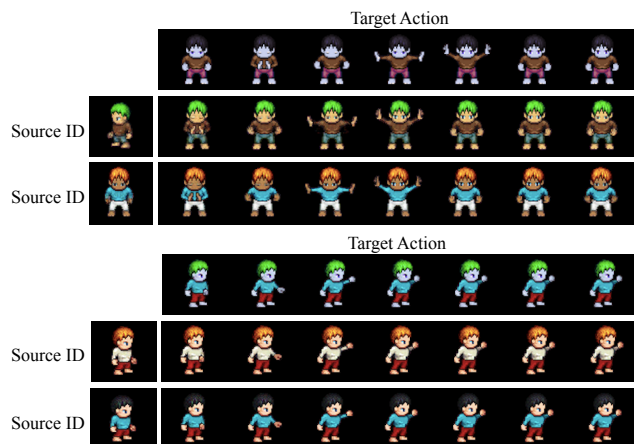

Figure 1. VDSM action transfer for the Sprites dataset [60]. The target action embedding is transferred swapped with one from a different identity and the sequence is generated.

generalizability [41, 13], structured/controllable inference and prediction [106], attribute transfer [14, 73], and improved performance on downstream tasks [9, 57, 79, 74, 98]. Unfortunately, being able to consistently learn a disentangled representation such that the factors correspond with meaningful attributes has been shown to be both theoretically and empirically impossible without the use of some form of supervision or inductive bias [21, 64]. However, acquiring high-quality supervision is expensive and time-consuming. Whilst many methods rely on such supervision, we consider how the implicit structure embedded in video data can be leveraged and reflected in the structure of the model, in order to achieve unsupervised disentanglement.

We propose Video Disentanglement via State-Space-Modeling (VDSM). VDSM is motivated by a careful consideration of the generative structure of a video sequence, which is assumed to be composed of identity (i.e. the canonical appearance of an object or person), an action

(i.e. dictating the dynamics governing change over time), and pose (i.e. the time varying aspects of appearance). In summary, VDSM: 1) Is a completely unsupervised approach that avoids the need for adversarial training. 2) Incorporates a novel structure designed to factorise appearance and motion, using a strong mixture of decoders to separate identity. 3) Produces embeddings that achieve state-of-the-art classification performance. 4) Far exceeds GAN based approaches in sequence generation (evidenced by FID scores). 5) Produces superior disentanglement compared to approaches which use forms of supervision. 6) Exceeds accuracy consistency by over 30% compared to the nearest competing approach. 7) Produces qualitative results that reflect the quantitative performance in terms of disentanglement and image quality.

The rest of this paper is structured as follows. First we discuss related work in Section 2 before describing the structure and training of VDSM in Section 3. We provide qualitative and quantitative experiments in Section 4 and conclude in Section 5.

2. Related Work

Disentanglement is an ongoing challenge for machine learning. However, achieving consistent, interpretable disentanglement without some form of supervision or inductive bias is impossible [64]. A large body of recent work seeks disentanglement via the incorporation of various levels of weak- [98, 65, 22, 13, 85, 17] or semi-supervision [66, 71, 62, 87]. Many of these methods seek disentanglement between static and time-varying factors in sequences (such as content and pose). Impressive results have been achieved in extending this goal to multi-object tracking [3, 53, 24, 54, 89, 42]. Such an approach has the advantage of trading the need for explicit supervision with structural inductive bias. For example, Grathwohl & Wilson (2016) [30] utilize variational inference with hierarchical latent factors and a slow-feature regularizing penalty to disentangle content and pose. Denton & Birodkar (2017) [22] disentangle content from pose by combining adversarial training with two deterministic, recurrent encoder-decoders. They incorporate an adversarial component which discriminates between content encodings from within the same or between different sequences. Other methods which seek disentanglement between static and time-varying factors include: S3VAE [111], which uses recurrent variational autoencoders; VideoVAE [36], a semi-supervised technique that enables attribute control for video generation; Factorizing VAE (FAVAE) [105], which uses multi-scale time-based convolution with variational autoencoders; Disentangled Sequential Autoencoders (DSA) [60] and Disentangled State Space Model (DSSM) [70] which undertake structured variational inference; and G3AN [100] and MoCo-GAN [93] both Generative Adversarial Networks (GANs).

Other networks which leverage structure to model sequence dynamics include a wide range of latent variable and state-space variational methods. For instance, Deep Kalman Filters [55] extend the structure of the traditional Kalman filter to incorporate recent deep learning techniques for video prediction. Structured inference networks [56] leverage similar structural considerations for sequential data, facilitating time-series causal inference from the disentangled representations. These networks are similar to other stochastic latent variable networks for sequence modeling, which tend to vary according to the structure imposed in the generative and inference models (see *e.g.* [8, 108, 101, 33, 34, 80, 82, 7, 18, 68, 4, 101, 31, 26, 29]).

Not all methods are concerned with achieving factor disentanglement, and many are primarily designed to generate or predict future video frames. These networks may still include structured variational and/or adversarial techniques. Recent attempts at video generation include Sin-GAN [83] which produces a video sequence from a single image; various multi-scale generative adversarial networks [69, 19, 1]; the Latent Video Transformer [75] which applies transformers [95] to discrete latent representations learned in an autoencoder [76]; and pixel-level autoregressive methods [102, 47].

Evidently, the tasks of video generation and disentanglement overlap, particularly when structured networks are used. The more inductive bias is used, the more domain-specific the network tends to become. For example, the network may incorporate inductive bias corresponding to the physical laws of interaction and motion [25, 6, 107, 16, 92, 32]. We prefer to keep our method general, such that it may be applied to non-vision related tasks for which the benefits of such inductive bias may be inappropriate.

In terms of structure, our method 'VDSM', is probably closest to DSA [60], Structured Inference Networks [56], DSSM [70] and Factorized Hierarchical VAE (FS-VAE) [43]. In terms of the computer vision applications, VDSM is most similar to the recently released S3VAE [111] and G3AN [100]. The latter two methods seek to disentangle motion and appearance for video data, and incorporate various forms of weak- or semi-supervision, such as optical flow [111], or labels for conditional motion generation [100]. Importantly, and in contrast with these two methods, our network does not use any form of supervision.

Whilst adversarial methods such as G3AN and MoCo-GAN are popular and have been shown to work well for density estimation and sequence generation, they are also notoriously difficult and unreliable to train [71, 59, 27]. Furthermore, recent work has highlighted that adversarial training is potentially unnecessary, and that non-adversarial methods can achieve comparable or better results across a wide range of tasks [71, 76, 39, 94, 103, 27]. Finally, supervision may not always be available, and unsupervised
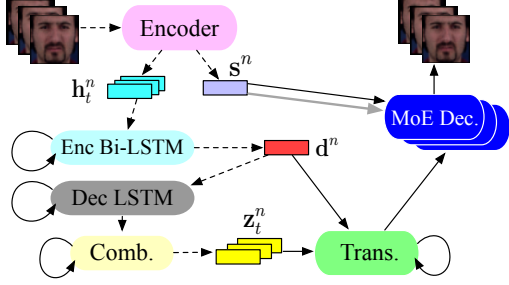
Figure 2. Block diagram of VDSM architecture. Gray lines indicate mixture of expert selection, dashed lines indicate inference, solid lines indicate generation, and looped arrows indicate autoregressive dependency.
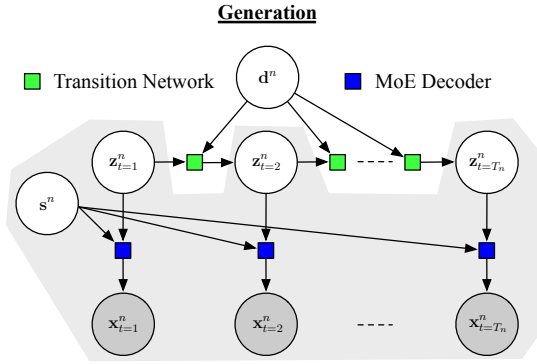

Figure 3. Generative model for VDSM with images $\mathbf{x}$, latent pose $\mathbf{z}$, motion dynamics $\mathbf{d}$, and static factors / content $\mathbf{s}$. Grey shading indicates pre-trained components.

methods are more generally applicable in scenarios where labels (even partial labels) are not available. VDSM intends to address the considerable challenge of disentangling key generative factors in sequences (pose, identity, dynamics) without supervision, and without adversarial training.

3. Video Disentanglement with State-Space Modeling (VDSM)

Consider N sequences of natural images $\{\mathbf{x}_{1:T_n}^n\}_{n=1}^N$ for which we may assume each image in a sequence lies on some significantly lower-dimensional manifold. We wish to encode the images via a stochastic mapping $p_\theta(\mathbf{z}_t^n, \mathbf{s}^n, \mathbf{d}^n | \mathbf{x}_{t=1:T_n}^n)$ where $\mathbf{x}_{t=1:T_n}^n$ is the nth sequence of T_n images, and $\mathbf{z} \in \mathbb{R}^{\kappa_z}$, $\mathbf{s} \in \mathbb{R}^{\kappa_s}$ and $\mathbf{d} \in \mathbb{R}^{\kappa_d}$ provide compact representations for the time varying latent factors (e.g. pose), static latent factors (e.g. identity), and the action dynamics (e.g. waving), respectively.

A block diagram for VDSM is shown in Figure 2. Inspecting this figure, it can be seen that images from a sequence are first encoded individually into two latent partitions $\mathbf{h}_t^n$ (which vary over the sequence), and $\mathbf{s}^n$ (which remain constant throughout a sequence. The time varying features are summarized by $\mathbf{d}^n$ which represents the dynamics

or action being performed, and then decoded following a sequence-to-sequence (seq2seq) structure [91]. A combination network (in yellow) takes the output of the seq2seq network and parameterizes a posterior distribution $\mathbf{z}_t^n$ which, when combined with the dynamics $\mathbf{d}^n$ and the static factors $\mathbf{s}^n$, are decoded to reconstruct the original images. In the decoding stage, the static factors $\mathbf{s}^n$ are also used to blend weights from a bank of decoders that form the Mixture of Experts (MoE) decoder which specializes in reconstructing information relevant to identity.

According to the task of image reconstruction through a structured, probabilistic bottleneck, VDSM resembles a variational autoencoder [51, 78]. We incorporate inductive bias in the form of hierarchical latent structure, dynamic priors, Markov factorization, and architectural constraints (e.g. MoE and seq2seq). The **generative model** for VDSM is shown in Figure 3 and can be factorized as follows:

$$p_\theta(\mathbf{x}_{t=1:T_n}^n, \mathbf{s}^n, \mathbf{d}^n, \mathbf{z}_{t=1:T_n}^n) =$$

$$p_\theta(\mathbf{s}^n) p_\theta(\mathbf{d}^n) p_\theta(\mathbf{z}_{t=1}^n) \prod_{t=2}^{T_n} p_\theta(\mathbf{x}_t^n | \mathbf{s}^n, \mathbf{z}_t^n) p_\theta(\mathbf{z}_t^n | \mathbf{z}_{t-1}^n, \mathbf{d}^n)$$

$$p_\theta(\mathbf{s}^n) = \mathcal{N}(0, \frac{1}{\mathrm{N}_s})$$

$$\hat{\mathbf{s}}^n = \mathrm{softmax}\,(\mathbf{s}^n / \tau_s)$$

$$p_\theta(\mathbf{d}^n) = \mathcal{N}(\mathbf{0}, \mathbf{1})$$

$$p_\theta(\mathbf{x}_t^n | \mathbf{s}^n, \mathbf{z}_t^n) = \mathrm{Bern}\,(\mathrm{DEC}_{MoE}(\hat{\mathbf{s}}^n, \{\mathbf{s}^n, \mathbf{z}_t^n\}))$$

$$p_\theta(\mathbf{z}_{t=1}^n) = \mathcal{N}(\mathbf{0}, \mathbf{1})$$

$$p_\theta(\mathbf{z}_{t>1}^n) = \mathcal{N}\left(\mu_{tr}(\mathbf{z}_{t-1}^n, \mathbf{d}^n), \sigma_{tr}(\mathbf{z}_{t-1}^n, \mathbf{d}^n)\right)$$

(1)

Considering this factorization and Figure 3, for sequence n we sample an action from the dynamics/action factor $\mathbf{d}^n$ and an initial pose $\mathbf{z}_{t=1}^n$. These are fed into a transition network (green) which produces the next pose factor $\mathbf{z}_{t=2}^n$ and we repeat this for T_n timepoints. The transition network outputs the location $\mu_{tr}(\mathbf{z}_{t-1}^n, \mathbf{d}^n)$ and scale $\sigma_{tr}(\mathbf{z}_{t-1}^n, \mathbf{d}^n)$ for a diagonal Gaussian parameterizing the next timepoint $\mathbf{z}_t^n$. We then sample an identity/static factor $\mathbf{s}^n$. For each timepoint, we concatenate this factor with the pose $\mathbf{z}_t$ for timepoints $t = 1 : T_n$ and pass them into a Mixture of Experts (MoE) Decoder, resulting in the conditional likelihood $p_\theta(\mathbf{x}_t^n | \mathbf{s}^n, \mathbf{z}_t^n)$. The MoE weights are determined by $\hat{\mathbf{s}}^n$ which is derived by dividing factors $\mathbf{s}^n$ by temperature τ_s and applying a softmax operation. The conditional likelihood (i.e., the distribution of generated images) is Bernoulli distributed according to Eq. 1 but can be chosen according to the data. N_s (line 3 Eq. 1) represents a prior 'guess' as to the number of distinct individuals/identities. $\hat{\mathbf{s}}^n$ is simplectic and similar to the Dirichlet distribution; it sums to 1, and the parameter τ_s controls the temperature [28].[1] A Dirichlet distribution was found to be less stable to train.

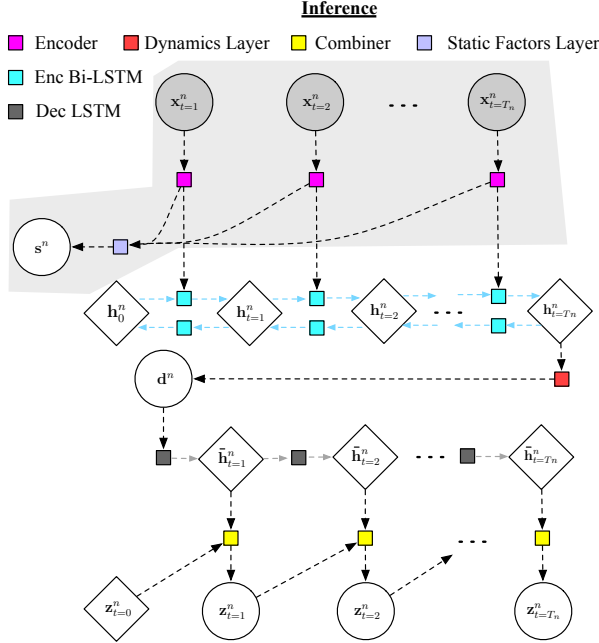[1]Similar to the Gumbel Softmax, or Concrete distributions [67, 44].

Figure 4. Inference model for VDSM with images $\mathbf{x}$, latent pose $\mathbf{z}$, motion dynamics $\mathbf{d}$, encoder bi-LSTM and decoder LSTM hidden states $\mathbf{h}$ and $\bar{\mathbf{h}}$ resp., and static factors / content $\mathbf{s}$. [49]. Together the encoder and decoder LSTMs form a seq2seq model with dynamics $\mathbf{d}^n$ as the bottleneck/summary vector. Grey background shading indicates pre-trained components.

In order to undertake this inference, we leverage stochastic variational inference (SVI) [45, 78, 51] to learn an approximate posterior distribution $q_\phi(.)$ according to the VDSM **inference model** shown in Equation 2. In words, the static factor representing the identity of an individual $\mathbf{s}^n$ is represented as a diagonal-Gaussian distribution. The location and scale for this Gaussian are parameterized as functions f_s of the average of the identity encodings of the images in a sequence using encoder $\text{ENC}_s(\mathbf{x}^n_{t=1:T_n})$. The function f_s is a linear/fully-connected layer with a non-linear activation. By taking the average, we force the network to learn factors which remain constant over the course of a sequence (i.e., the identity). Before the softmax operation, τ_s is used to anneal the temperature of this distribution during training. The time-varying pose embeddings from the encoder ENC_z (not shown in Eq. 2) are fed into a recurrent, bidirectional Long Short Term Memory (bi-LSTM) [40] network, the final hidden state of which $\mathbf{h}^n$ is fed through functions $f_{d\mu}(.)$ and $f_{d\sigma}(.)$ (also linear layers with non-linear activations) to parameterize a Gaussian distribution for dynamics factors $\mathbf{d}^n$. The dynamics embedding $\mathbf{d}^n$ is duplicated and used as the initial hidden and cell states for a decoding (uni-directional) LSTM. This decoder LSTM produces hidden states at each timestep $\bar{\mathbf{h}}_t$ which are fed to the combiner functions μ_{comb} and σ_{comb}, with $\mathbf{z}^n_{t-1}$

from the previous timestep.

$$q_\phi(\mathbf{s}^n, \mathbf{d}^n, \mathbf{z}_{t=1:T_n} | \mathbf{x}^n_{t=1:T_n}) =$$
$$q_\phi(\mathbf{s}^n | \mathbf{x}^n_{t=1:T_n}) q_\phi(\mathbf{d}^n | \mathbf{x}^n_{t=1:T_n}) q_\phi(\mathbf{z}^n_{t=1}, | \mathbf{x}^n_{t=1:T_n}, \mathbf{d}^n)$$
$$\prod_{t=2}^{T_n} q_\phi(\mathbf{z}^n_t | \mathbf{z}^n_{t-1}, \mathbf{x}^n_{t=1:T_n}, \mathbf{d}^n)$$
$$q_\phi(\mathbf{s}^n | \mathbf{x}^n_{t=1:T_n}) = \mathcal{N}(s_{loc}, s_{scale})$$
$$s_{loc} = f_{s,loc}\left(\frac{1}{T_n} \sum_{t=1}^{T_n} (\text{ENC}_s(\mathbf{x}^n_t))\right)$$
$$s_{scale} = f_{s,scale}\left(\frac{1}{T_n} \sum_{t=1}^{T_n} (\text{ENC}_s(\mathbf{x}^n_t))\right)$$
$$q_\phi(\mathbf{d}^n | \mathbf{x}^n_{t=1:T_n}) = (\mathcal{N} f_{d\mu}(\mathbf{h}^n), f_{d\sigma}(\mathbf{n}^n))$$
$$q_\phi(\mathbf{z}^n_t | \mathbf{x}^n_{t=1:T_n}, \mathbf{d}^n) =$$
$$\mathcal{N}(\mu_{comb}(\mathbf{z}^n_{t-1}, \bar{\mathbf{h}}^n_t, \mathbf{d}^n), \sigma_{comb}(\mathbf{z}^n_{t-1}, \bar{\mathbf{h}}_t, \mathbf{d}^n))) \quad (2)$$

$$\max_{\phi,\theta} \left\langle \left\langle \frac{p_\theta(\mathbf{x}^n_{t=1:T_n}, \mathbf{s}^n, \mathbf{d}^n, \mathbf{z}^n_{t=1:T_n})}{q_\phi(\mathbf{s}^n, \mathbf{d}^n, \mathbf{z}^n_{t=1:T_n} | \mathbf{x}^n_{i=1:T_n}))} \right\rangle_{q_\phi} \right\rangle_{p_{\mathcal{D}}(\mathbf{x}^n)} \quad (3)$$

The goal of the inference model is to make the problem of latent inference possible through the use of parametric approximating posteriors. Together, the generation and inference models can be optimized as part of a stochastic, amortized, variational inference objective, known as the Evidence Lower Bound (ELBO), given in Equation 3. The ELBO objective may be derived similarly to [56] (also see supplementary material) and is shown in Equation 4. In Eq. 4, each λ term represents a hyperparameter used for annealing the corresponding KL (Kullback-Liebler divergence) objective term during training.

$$\mathcal{L}(\mathbf{x}^n_{1:T_n}; (\theta, \phi)) =$$
$$\sum_{t=1}^{T_n} \mathbb{E}_{q_\phi(\mathbf{z}^n_t, \mathbf{s}^n | \mathbf{x}^n_{1:T_n})} [\log p_\theta(\mathbf{x}^n_t | \mathbf{z}^n_t, \mathbf{s}^n)]$$
$$-\lambda_d(\text{KL}(q_\phi(\mathbf{d}^n | \mathbf{x}^n_{1:T_n}) || p_\theta(\mathbf{d}^n)))$$
$$-\lambda_s(\text{KL}(q_\phi(\mathbf{s}^n | \mathbf{x}^n_{1:T_n}) || p_\theta(\mathbf{s}^n))) \quad (4)$$
$$-\lambda_z(\text{KL}(q_\phi(\mathbf{z}^n_1 | \mathbf{x}^n_{1:T_n}, \mathbf{d}^n) || p_\theta(\mathbf{z}^n_1))$$
$$-\lambda_z \sum_{t=2}^{T_n} \mathbb{E}_{q_\phi(\mathbf{z}^n_{t-1} | \mathbf{x}^n_{1:T_n}, \mathbf{d}^n)}$$
$$\text{KL}(q_\phi(\mathbf{z}^n_t | \mathbf{z}^n_{t-1}, \mathbf{d}^n, \mathbf{x}^n_{1:T_n}) || p_\theta(\mathbf{z}^n_t | \mathbf{z}^n_{t-1}, \mathbf{d}^n)))$$

3.1. Functional Form

As depicted in Figures 3 and 4, there are a number of functions with learnable parameters in the generative and inference models of VDSM.

Encoder and Static Factors: The encoder functions $\text{ENC}_{\{s,z\}}(.)$ (pink block in Figure 4) are used to infer the static and time varying embeddings of the images at each timestep. It comprises 5 layers of anti-aliased convolution [110] downsampling, as well as two separate fully-connected embedding layers with non-linear activations. The identity embedding is averaged over a sequence and fed through a fully-connected layer $f_s(.)$ (light gray block in Figure 4) which is used to infer the identity/static factor for a sequence $\mathbf{s}^n$.

Dynamics Layer and LSTMs: The Dynamics layer (red block Figure 4) comprises $f_{d\mu}(.)$ and $f_{d\sigma}(.)$ which are fully-connected neural network layers with non-linear activations used to infer the location and scale of the dynamics/action factor $\mathbf{d}^n$ and are fed with the last hidden state from the encoder bi-LSTM. It therefore represents a bottleneck inside a seq2seq network. The seq2seq network's unidirectional LSTM decoder uses the inferred $\mathbf{d}^n$ as the initial hidden and initial cell states, and produces per-timestep hidden states $\bar{\mathbf{h}}_t^n$ which are fed to the combiner.

The Combiner: The Combiner (yellow block in Figure 4) is used to infer the current latent pose factor at time t given the current hidden state from the decoder LSTM, the previous latent pose factor, and the dynamics: $\mathbf{z}_t^n | \mathbf{z}_{t-1}^n, \bar{\mathbf{h}}_t^n$. The function is parameterized as follows:

$$
\begin{aligned}
\hat{\mathbf{h}}_t^n &= \{\bar{\mathbf{h}}_t^n, \mathbf{d}^n\} \\
\mathbf{c}_t^n &= 0.5(\tanh(f_7(\mathbf{z}_{t-1}^n)) + \hat{\mathbf{h}}_t^n) \\
\text{loc-comb}_t^n &= f_8(\mathbf{c}_t^n) \\
\text{scale-comb}_t^n &= \log(1 + \exp(f_9(\mathbf{c}_t^n))) \\
\mathbf{z}_t^n &\sim \mathcal{N}(\text{loc-comb}_t^n, \text{scale-comb}_t^n)
\end{aligned}
\tag{5}
$$

Where $\{\bar{\mathbf{h}}_t^n, \mathbf{d}^n\}$ is the concatenation of the LSTM decoder hidden state and the dynamics, and functions $f_{7-9}(.)$ are fully-connected neural network layers with non-linear activations. During development we explored the use of Inverse Autogressive Flows (IAFs) to increase the expressivity of the approximating posterior. IAFs are an adapted form of Normalizing Flow [52, 77] designed for efficient transformation of simple posterior approximating distributions. IAFs leverage an invertible transformation function with a particularly tractable Jacobian determinant [77, 52, 49]. However, at least in our particular experiments, we found that the concomitant increase in computational complexity was not justified by the negligible performance increase.

Transition Network: Similar to the state-space transition model in [55], the transition network (green block Figure 3) parameterizes the latent time-varying factors $\mathbf{z}_t^n$ with functions $\mu_{tr}(.), \sigma_{tr}(.)$, each with the following parameterization:

$$
\begin{aligned}
\mathbf{g}_t^n &= \text{sigmoid}(f_2(\text{ReLU}(f_1(\{\mathbf{z}_{t-1}^n, \mathbf{d}^n\})))) \\
\mathbf{h}_t^n &= f_4(\text{ReLU}(f_3(\{\mathbf{z}_{t-1}^n, \mathbf{d}^n\}))) \\
\text{loc-trans}_t^n &= \mathbf{g}_t \circ \mathbf{h}_t^n + (1 - \mathbf{g}_t^n) \circ f_5(\{\mathbf{z}_{t-1}^n, \mathbf{d}^n\}) \\
\text{scale-trans}_t^n &= \log(1 + \exp(f_6(\text{ReLU}(\mathbf{h}_t^n))))
\end{aligned}
\tag{6}
$$

where $\circ$ indicates the Hadamard/elementwise product, $\{\mathbf{z}_{t-1}^n, \mathbf{d}^n\}$ is the concatenation of the previous pose vector with the dynamics vector for sequence n, and the functions $f_{1-6}(.)$ are fully-connected neural network layers with non-linear activations. This network is used as part of the generation process to generate successive latent pose factors $\mathbf{z}_t^n | \mathbf{z}_{t-1}^n$.

Mixture of Experts Decoder: The Mixture of Experts (MoE) [109, 84, 46, 23] decoder $\text{DEC}_{MoE}(.)$ (blue block in Figure 3) constitutes a bank of N_s upsampling convolution image decoders. Each decoder comprise a series of 2D transpose convolution and Leaky ReLU activation layers. The weights and biases for the transpose convolution operations of the bank of MoE decoders are *blended* according to a weighted average, where the weighting is determined by the identity mixture parameter $\hat{\mathbf{s}}^n$. In addition, $\mathbf{s}^n$ (which is $\hat{\mathbf{s}}^n$ before temperature annealing and the softmax operation) is concatenated with $\mathbf{z}_t^n$ and fed to the input of the decoder. As the temperature parameter τ_s is annealed during training, the mixture parameter becomes more 'peaked' resulting in a tendency to select individual decoders from the bank, rather than mix between them. Each decoder in the bank is therefore given the opportunity to specialize in reconstructing a particular identity, whilst also leveraging information encoded in the weights in the entire bank. The MoE decoder is used as part of the generation process to generate the images $\mathbf{x}_t^n | \mathbf{z}_t^n, \mathbf{s}^n$.

3.2. Training and Testing VDSM

Training: Training is split into two stages. Without two stages, the network tends to push both pose and identity information into $\mathbf{d}$ to avoid the averaging operation associated with $\mathbf{s}$. We start by **pretraining** the encoder $\text{ENC}_{\{s,z\}}$, MoE decoder DEC_{MoE} and the Static Factors Layer f_s without modeling the autoregressive structure of the data (this is illustrated in the gray shaded regions of Figures 3 and 4. In other words, we begin by treating the data as non-sequential, and group images from the same individuals/identities into batches in order to facilitate the inference of the identity factor $\mathbf{s}^n$ and i.i.d. $\mathbf{z}_t^n$. During this first stage, the objective function is reduced to the following:

$$
\begin{aligned}
\mathcal{L}(\mathbf{x}_{1:T_n}^n; (\theta', \phi')) = \frac{1}{N} \sum_{i=1}^{T} (\mathbb{E}_{q_{\phi'}(\mathbf{z}_i, \mathbf{s}|\mathbf{x}_i)}[\log p_{\theta'}(\mathbf{x}_i|\mathbf{z}_i, \mathbf{s})] \\
- \lambda_z \text{KL}[q_{\phi'}(\mathbf{z}_i|\mathbf{x}_i)\|p_{\theta'}(\mathbf{z})] - \lambda_s \text{KL}[q_{\phi'}(\mathbf{s}|\mathbf{x}_i)\|p_{\theta'}(\mathbf{s})])
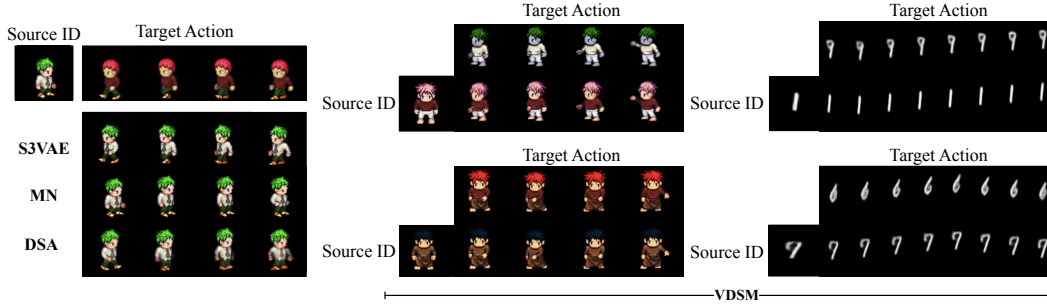\end{aligned}
\tag{7}
$$

Figure 5. **Left**: Sprites dataset [60] action transfer examples for S3VAE [111], MN (MonkeyNet) [86] and DSA [60] (results adapted from [111]). **Middle** and **right**: VDSM action transfer on Sprites and moving MNIST datasets [88] respectively. 'Target Action' indicates the desired action, whilst 'Source ID' indicates the identity to which the action is transferred during sequence generation.

Here we deliberately use subscript i rather than t to emphasize that the images need not be in any sequential order, they just need to be grouped according to the same identity. Furthermore, we indicate that we are only training a subset of the inference and generation parameters with (θ', ϕ'). During pretraining, the KL weight on the static factors is inversely annealed $\lambda_s = \{0.1, ..., 1.0\}$ (i.e. low to high) with respect to the KL weight on the pose factor $\lambda_z = \{30, ..., 1.0\}$ (i.e. high to low), with the number of steps equal to the number of training epochs. This forces information to flow through the static factor $\mathbf{s}^n$ because the high weight λ_z pushes the time varying factors $\mathbf{z}_i$ to zero. The static factor is computed as an average over embeddings and as such, it is forced to represent the information which is consistent over the group of images (i.e. the identity). Simultaneously, the temperature parameter τ_s is gradually increased, resulting in a shift from uniform MoE decoder blending weights, to increasing specialization. Then, as the weight λ_z falls, the pose factors $\mathbf{z}_i$ start to become useful in encoding information that varies across the groups of images of the same individual. As such, the pretraining enables us to learn the two most highly-parameterized deterministic functions used for mapping images to latent factors and vice versa: the encoder and the MoE decoder. It is worth noting that the pretraining stage yields a valuable model in its own right, that disentangles identity from pose. For instance, in Figure 8, the expression can be swapped with any ID and vice versa.

The **second stage of training** is concerned with disentangling the sequence dynamics, and introduces the seq2seq encoder and decoder LSTMs, the Combiner, and the transition network. The training objective is given by the full ELBO in Eq. 4. The weights and biases for the encoder and MoE decoder are all frozen at the end of pretraining, apart from the final layers preceding the parameterizations of $\mathbf{s}^n$ and $\mathbf{z}_t^n$ which are allowed to vary during this second stage. Allowing these layers to be fine-tuned is important in ensuring that the model learns a valid ELBO during the sequential modeling stage (i.e. by allowing the parameters to

vary we facilitate variational inference). Annealing is used with $\lambda_z = \{0.1, ..., 1.0\}$ to improve inference of $\mathbf{z}_t$ and to help prevent posterior collapse during training [15].

To infer $\mathbf{s}^n$, which is primarily learned during the first stage of training (but fine-tuned in the second stage), we only need to be able to group images of the same individual (and these may or may not be from the same sequence). For $\mathbf{d}^n$ the images need to be from sequences, but no supervision is required. This allows us to sample multiple images of the same individual, without needing labels for which individual is being sampled. So long as there is, on average, some variation across identity between groups of images (or between sequences), the network can infer the identity of the individual. VDSM is trained using the Adam [50] optimizer with the Stochastic Variational Inference algorithm in the Pyro probabilistic programming language [11]. Minimal hyperparameter tuning was undertaken.[2]

Testing: At test time, the generative model may either be used unconditionally (where the factors $\mathbf{s}^n, \mathbf{d}^n$, and $\mathbf{z}_t^n$ are sampled from their priors), or be used conditioned on some initial state. In the latter case, the inference network may be used to derive $\mathbf{s}^n, \mathbf{d}^n, \mathbf{h}_{t=1}^n$, and $\mathbf{z}_{t=1}^n$ from a sample sequence, or even a single image. The generative model is then conditioned on these initial factors, and further sampling is undertaken according to Figure 3. The benefit of this flexible structure is that it allows different factors to be swapped or controlled independently in order to mix any identity with any action dynamics.

4. Experiments

VDSM is evaluated on four datasets: MUG [2]; colored Sprites [60]; synthetic swinging pendula; moving MNIST (similar to [88]). MUG [2] comprises 3528 videos of 52 individuals performing different facial expressions for anger, disgust, fear, happiness, sadness, and surprise. The performances vary in length, and were downsampled by a factor of two (to approximately 8 fps) to improve data efficiency. The

---

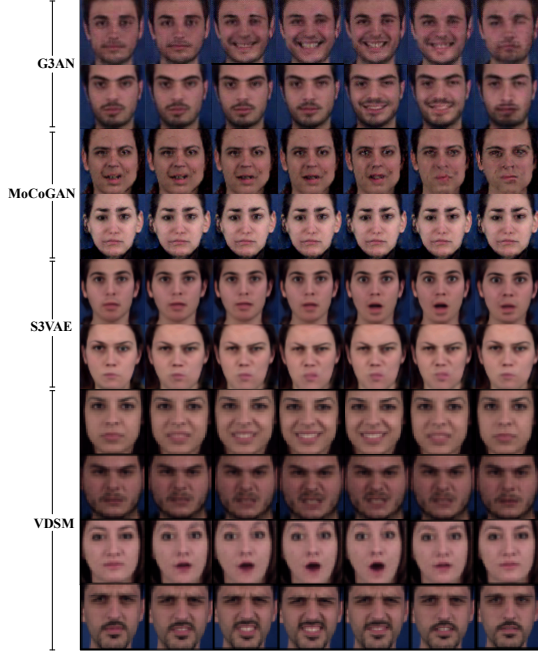[2]Network details can be found in supplementary material.

Figure 6. Downsampled sequence generation samples from G3AN [100], MoCoGAN [93], S3VAE [111], and VDSM (ours). Figure adapted from [111, 100].
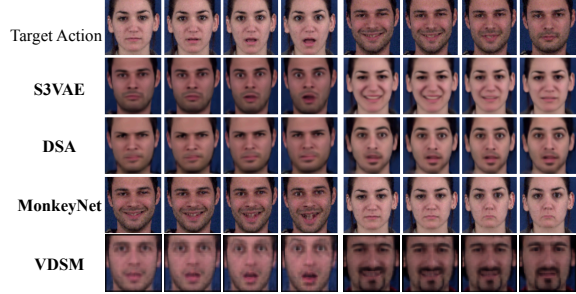

Figure 7. Action transfer on the MUG dataset [2] for S3VAE [111], DSA [60], MonkeyNet [86], and VDSM (ours). Figure adapted from [111].

Attr.	Acc. Cons. ↑	ID Acc.	Dyn. Acc.
Body (Sprites)	99.67	98.67 ↑	52.67 ↓
Shirt (Sprites)	99.67	99.67 ↑	47.33 ↓
Pant (Sprites)	99.00	97.67 ↑	31.67 ↓
Hair (Sprites)	100.00	99.33 ↑	21.00 ↓
Action (Sprites)	99.33	45.00 ↓	99.33 ↑
ID (MUG)	98.67	99.00 ↑	65.67 ↓
Action (MUG)	88.00	57.33 ↓	83.33 ↑

Table 1. For each ground-truth attribute of Sprites and MUG, this table shows a breakdown of VDSM's consistency (Acc. Cons.) and disentanglement accuracy scores in percent using the separate identity (ID) and dynamics/action (Dyn.) embeddings. Arrows indicate whether a higher ↑ or lower ↓ score is preferred.

Method	Acc. **Sprites**	Acc. **MUG**	$H(y)$	$H(y|x)$
VDSM	**99.53**	**93.33**	**2.21**	0.203
S3VAE [111]	**99.49**	70.51	1.760	**0.135**
DSA [60]	90.73	54.29	1.657	0.374
MoCo [93]	92.89	63.12	1.721	0.183

Table 2. Results for accuracy consistency (averaged across attributes) between different methods for the Sprites and MUG datasets, and results for Inter-Entropy $H(y)$ (higher is better) and Intra-Entropy $H(y|x)$ (lower is better) [36] on the MUG dataset.

images were aligned using OpenFace [5], centre cropped, and resized to 64x64 (similar to [93, 111]). Random segments of length 16 frames were sampled from the sequences for training, and a 15% holdout set was used for testing.

The Sprites dataset comprises 64x64 sequences of cartoon characters from the Universal LPC SpriteSheet Character Generator performing 3, 8-frame long action sequences (spellcast, slashing, and walking), from 3 viewing angles. Following [60] we create characters with 7 body types, 4 shirts, 8 hairstyles, and 5 pants, resulting in 1120 identities. A 10% holdout set was used for testing. The moving MNIST dataset [88] comprises 55,000 16-frame long sequences of randomly sampled MNIST digits [58] moving in random trajectories. A 10% holdout set was used for testing. Finally, the pendulum dataset represents a synthetic dataset which comprises seven different colored pendula swinging at two speeds (fast and slow).

We evaluate quantitatively in terms of (1) Fréchet Inception Distance (FID) [37, 12] using a 3D ResNeXt 101 [104, 35] pre-trained on Kinetics [48] as per [100], (2) consistency at identity and action classification between real and generated sequences [61, 111], and (3) in terms of identity and action classification score using the separate identity and dynamics embeddings.[3] For (2), the consistency is measured by comparing the predictions of a classifier trained to predict ground truth factors from real images with predictions from sampled images. If the quality of the im-

ages is high and the network is encoding information about the identity and action, then the predictions for real and generated images should be close. For (3) we expect that disentangled identity and action embeddings to be informative for predicting their respective factors, but not informative for predicting each other's factors. Finally, we evaluate quantitatively, in terms of identity swapping, dynamics swapping, and sequence generation quality.

4.1. Quantitative Evaluation

Quantitative results were obtained for 1000 samples from the test sets. The results in Table 1 show that the ID embedding is predictive of ID or ID-related attributes (acc. > 97% for Sprites), but not of action (acc. = 45% for Sprites). Similarly, the dynamics embedding was highly predictive of action (acc. > 99% for Sprites) but not of ID-related attributes (acc. < 53% for Sprites). Interestingly,

---

[3]Additional results may be found in supplementary material.

the ID could be classified using the dynamics embedding for MUG (acc. 66%). We believe this may be because individuals exhibit unique action dynamics that make them identifiable. This behavior may have relevant application elsewhere (*e.g.* gait recognition). Future work should explore whether this behavior is exhibited by other methods.

The breakdown of VDSM's accuracy consistency scores in Table 1 demonstrates that there is high corroboration between action and ID-related attribute classifier predictions for real and generated images. Indeed, the results for VDSM exceed those from competing methods. In particular, see the results in Table 2 where MUG performance was $> 30\%$ higher than the best competing method (S3VAE) despite this method using additional supervision (*e.g.* optical flow). Table 2 also shows the results for Inter- and Intra-Entropy [36] for the same classifier used for the disentanglement evaluation. These are computed as $H(y) = -\sum_y p(y) \log p(y)$ and $H(y|x) = -\sum_y p(y|x) \log p(y|x)$ respectively, where y is the predicted attribute label, and x is the given sequence. The inter- and intra-entropies are averaged across identity and action prediction results. The results for inter-entropy provide a proxy for diversity, and indicate that, in generation mode, VDSM produces the most diverse samples. The results for intra-entropy demonstrate competitive performance with another unsupervised method, DSA. However, together with the disentanglement and accuracy results in Table 2 suggest that VDSM's primary strength is superior, unsupervised, disentangled representation learning.
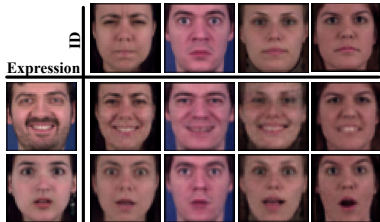

Figure 8. Identity/pose transfer using the pre-trained (non-sequential) components of VDSM on the MUG dataset [2].

FID scores for MUG are shown in Table 3. For this evaluation 1000 samples were generated firstly by using $\mathbf{d}^n$ and $\mathbf{s}^n$ embeddings derived from real sequences in the test set (denoted 'reconstruction' in the table), and secondly by generating images unconditionally from the priors (denoted 'generation' in the table). In both case, the FID substantially improves upon competing adversarial methods.

4.2. Qualitative Evaluation

Figures 1 and 5 illustrate VDSM's action transfer performance on the Sprites dataset, and some comparisons are included in the leftmost part of Figure 5. It can be seen that VDSM facilitates disentangled transfer, with clean separation of time varying and time static information. In contrast,

Method	FID
VDSM (ours, generation)	**44.12**
VDSM (ours, reconstruction)	**17.64**
G3AN [100]	67.12
MoCo [93]	87.11
VGAN [96]	160.76
TGAN [81]	97.07

Table 3. FIDs for MUG. We include VDSM in generation and reconstruction modes, where samples are drawn unconditionally from priors or are drawn from approximate posteriors derived from test set sequences, respectively.

DSA and MonkeyNet struggle to transfer the action, and S3VAE requires auxiliary information to achieve comparably (*e.g.* optical flow). The rightmost part of Fig. 5 shows action transfer of VDSM on the moving MNIST dataset.

Figure 6 compares VDSM's sequence generation performance against G3AN, MocoGAN, and S3VAE, where it can be seen that VDSM outperforms the adversarial methods, and performs comparably to S3VAE, despite not requiring additional information (*e.g.* optical flow).[4] Figure 7 compares the action transfer performance of VDSM against DSA, S3VAE and MonkeyNet, again highlighting that VDSM is comparable to S3VAE despite not requiring additional information during training (*e.g.* optical flow). Figure 8 demonstrates the *pre-trained* components of VDSM performing identity and expression transfer, illustrating how the model can disentangle pose and identity without needing to model sequences.

Results on the synthetic pendulum dataset are shown in Figure 3 in the supplementary material. These results demonstrate that, even though VDSM was only trained on sequence segments which were 16 frames long, the network can generate sequences following the dynamics of a pendulum with an arbitrary number of frames (40 frames are shown in the figure). The figure also demonstrates action swapping, where the dynamics of one pendulum is transferred to another.

5. Summary

We presented VDSM, an unsupervised state-space model for video disentanglement. VDSM incorporates a range of inductive biases including a hierarchical latent structure, a dynamic prior, a seq2seq network, and a mixture of experts decoder. The evaluation demonstrated that the network informatively embeds and disentangles static and time-varying factors, as well as generating quality video many frames into the future. VDSM's performance matches or exceeds that of methods requiring additional supervision (such as optical flow) or adversarial training.

---

[4]Sequences are downsampled - for full sequences see supplementary material.

References

[1] D. Acharya, Z. Huang, D.P. Paudel, and L. Van Gool. Towards high resolution video generation with progressive growing of sliced Wasserstein GANs. *arXiv:1810.02419*, 2018.

[2] N. Aifanti, C. Papachristou, and A. Delopoulos. The MUG facial expression database. *Proc. 11th Int. Workshop on Image Analysis for Multimedia Interactive Services*, 2010.

[3] A. Akhundov, M. Soelch, J. Bayer, and P. van der Smagt. Variational tracking and prediction with generative disentangled state-space models. *arXiv:1910.06205v1*, 2017.

[4] E. Aksan and O. Hilliges. STCN: stochastic temporal convolution networks. *ICLR*, 2019.

[5] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L-P. Morency. OpenFace 2.0: Facial behavior analysis toolkit. *13th IEEE International Conference on Automatic Face and Gesture Recognition*, 2018.

[6] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects relations and physics. *arXiv:1612.00222v1*, 2016.

[7] J. Bayer and C. Osendorfer. Learning stochastic recurrent networks. *arXiv:1411.7610v3*, 2015.

[8] P. Becker-Ehmck, J. Peters, and P. van der Smagt. Switching linear dynamics for Variational Bayes Filtering. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[9] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on pattern analysis and machine intelligence*, 2013.

[10] M. Besserve, A. Mehrjou, R. Sun, and B. Scholkopf. Counterfactuals uncover the modular structure of deep generative models. *arXiv:1812.03253v2*, 2019.

[11] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20, 2019.

[12] A. Borji. Pros and cons of GAN evaluation measures. *arXiv:1802.03446v5*, 2018.

[13] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: learning disentangled representations from grouped observations. *arXiv:1705.08841v1*, 2017.

[14] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. *arXiv:1608.06019*, 2016.

[15] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in Beta-VAE. *arXiv:1804.03599v1*, 2018.

[16] C. Chen, C. X. Lu, N. Trigoni, and A. Markham. DynaNet: Neural Kalman dynamical model for motion estimation and prediction. *arXiv:1908.03918v1*, 2019.

[17] J. Chen and K. Batmanghelich. Weakly supervised disentanglement by pairwise similarities. *arXiv:1906.01044v1*, 2019.

[18] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *arXiv:1506.02216v6*, 2016.

[19] A. Clark, J. Donahue, and K. Simonyan. Adversarial video generation on complex datasets. *arXiv:1907.06571v2*, 2019.

[20] E. Creager, D. Madras, J-H. Jacobsen, M. A. Weis, K. Swersky, T. Pitassi, and R. Zemel. Flexibly fair representation learning by disentanglement. *arXiv:1906.02589v1*, 2019.

[21] B. Dai and D. Wipf. Diagnosing and enhancing VAE models. *arXiv:1903.05789v1*, 2019.

[22] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. *NIPS*, 2017.

[23] D. Eigen, M.A. Ranzato, and I. Sutskever. Learning factored representations in a deep mixture of experts. *arXiv:1312.4312*, 2013.

[24] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: fast scene understanding with generative models. *arXiv:1603.08575v3*, 2016.

[25] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *31st Conference on Neural Information Processing Systems*, 2017.

[26] M. Fraccaro, S. K. Sonderby, U. Paquet, and O. Winther. Sequential neural models with stochastic layers. *arXiv:1605.07571v2*, 2016.

[27] A. Gabbay and Y. Hosen. Demystifying inter-class disentanglement. *arXiv:1906.11796v2*, 2019.

[28] A. Gelman, F. Bois, and J. Jiang. Physiological phamacokinetic analysis using population modeling and informative prior distributions. *American Statistical Association*, 91(436), 1996.

[29] A. Goyal, A. Sordoni, M. A. Cote, N. R. Ke, and Y. Bengio. Z-forcing: training stochastic recurrent networks. *31st Conference on Neural Information Processing Systems*, 2017.

[30] W. Grathwohl and A. Wilson. Disentangling space and time in video with hierarchical variational auto-encoders. *arXiv:1612.04440v2*, 2016.

[31] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber. Temporal difference variational auto-encoder. *International Conference on Learning Representations*, 2019.

[32] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. *arXiv:1906.01563*, 2019.

[33] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: learning behaviors by latent imagination. *arXiv:1912.01603v1*, 2019.

[34] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[35] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? *arXiv:1711.09577*, 2017.

[36] J. He, A. Lehrmann, M. Marino, G. Mori, and L. Sigal. Probabilistic video generation using holistic attribute control. *arXiv:1803.08085v1*, 2018.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to local Nash equilibrium. *NIPS*, 2017.

[38] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a definition of disentangled representations. *arXiv:1812.02230v1*, 2018.

[39] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv:2006.11239v1*, 2020.

[40] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1–32, 1997.

[41] H. Hosoya. Group-based learning of disentangled representations with generalizability for novel contents. *Proc. 28th IJCAI*, 2019.

[42] J. T. Hsieh, B. Liu, D. A. Huang, L. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. *32nd Conference on Neural Information Processing Systems*, 2018.

[43] W.-N. Hsu, Y. Zhang, and J. Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. *31st Conference on Neural Information Processing Systems*, 2017.

[44] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-Softmax. *arXiv:1611.01144v5*, 2017.

[45] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[46] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Proceedings of 1993 Internationl Joint Conference on Neural Networks*, 1993.

[47] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[48] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The Kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[49] D.P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. *29th Conference on Neural Information Processing Systems*, 2016.

[50] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980v9*, 2017.

[51] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114v10*, 2014.

[52] I. Kobysev, S.J.D. Prince, and M.A. Brubaker. Normalizing flows: an introduction and review of current methods. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2020.

[53] A. R. Kosiorek, H. Kim, I. Posner, and Y. W. Teh. Sequential attend, infer, repeat: generative modelling of moving objects. *32nd Conference on Neural Information Processing Systems*, 2018.

[54] J. Kossen, K. Stelzner, M. Hussing, C. Voelcker, and K. Kersting. Structured object-aware physics prediction for video modeling and planning. *arXiv.1910.02425v1*, 2019.

[55] R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv:1511.0512v2*, 2015.

[56] R. G. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. *Association for the Advancement of Artificial Intelligence*, 2017.

[57] 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.

[58] Y. LeCun, C. Cortes, and C. J. Burges. MNIST handwritten digit database. *ATT Labs*, 2010.

[59] J. Lezama. Overcoming the disentanglement vs reconstruction trade-off via Jacobian supervision. *ICLR*, 2019.

[60] K. Li and J. Malik. Implicit maximum likelihood estimation. *arXiv:1809.09087*, 2018.

[61] Y. Li and S. Mandt. Disentangled sequential autoencoder. *Proceedings on the 35th International Conference on Machine Learning*, 2018.

[62] M. llse, J. M. Tomczak, C. Louizos, and M. Welling. DIVA: domain invariant variational autoencoders. *arXiv:1905.10427*, 2019.

[63] F. Locatello, G. Abbati, T. Rainforth, T. Bauer, S. Bauer, B. Scholkopf, and O. Bachem. On the fairness of disentangled representations. *arXiv:1905.13662v1*, 2019.

[64] F. Locatello, S. Bauer, M. Lucic, G. Ratsch, S. Gelly, B. Scholkopf, and Bachem O. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv:1811.12359v3*, 2019.

[65] F. Locatello, B. Poole, G. Ratsch, B. Scholkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. *arXiv:2002.02886*, 2020.

[66] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel. The variational fair autoencoder. *arXiv:1511.00830*, 2017.

[67] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: a continuous relaxation of discrete random variables. *arXiv:1611.00712v3*, 2017.

[68] J. Marino, L. Chen, J. He, and S. Mandt. Improving sequential latent variable models with autoregressive flows. *2nd Symposium on Advances in Approximate Bayesian Inference*, 2019.

[69] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean squared error. *ICLR*, 2016.

[70] D. Miladinovic, M. W. Gondal, B. Scholkopf, J. M. Buhmann, and S. Bauer. Disentangled state space representations. *arXiv:1906.03255v1*, 2019.

[71] D. Moyer, S. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan. Invariant representations without adversarial training. *NeurIPS*, 2018.

[72] G. Parascandolo, N. Kilbertus, M. Rojas-Carulla, and B. Scholkopf. Learning independent causal mechanisms. *Proceedings of the 35 th International Conference on Machine Learning Learning*, 2018.

[73] O. Press, T. Galatni, S. Benaim, and L Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. *ICLR*, 2019.

[74] A. Raffin, A. Hill, K. R. Traore, T. Lesort, N. Diaz-Rodriquez, and D. Filliat. Decoupling feature extraction

from policy learning: assessing benefits of state representation learning in goal based robotics. *arXiv:1901.08651*, 2019.

[75] R. Rakhimov, D. Vollkhonskiy, A. Artemov, D. Zorin, and E. Burnaev. Latent video transformer. *arXiv:2006.10704v1*, 2020.

[76] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *arXiv:1906.00446v1*, 2019.

[77] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv:1505.05770v6*, 2016.

[78] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv:1401.4082*, 2014.

[79] K. Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv:1612.05299*, 2016.

[80] O. Rybkin, K. Pertsch, K. G. Derpanis, K. Daniilidis, and A. Jaegle. Learning what you can do before doing anything. *ICLR*, 2019.

[81] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. *ICCV*, 2017.

[82] K. Schmeckpeper, A. Xie, O. Rybkin, S. Tian, K. Daniilidis, S. Levine, and C. Finn. Learning predictive models from observations and interaction. *arXiv:1912.12773v1*, 2019.

[83] T.R. Shaham, T. Dekel, and T. Machaeli. SinGAN: learning a generative model from a single natural image. *ICCV*, 2019.

[84] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q.V. Le, G. E. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ICLR*, 2017.

[85] R. Shu, Chen Y., A. Kumar, S. Ermon, and B. Poole. Weakly supervised disentanglement with guarantees. *arXiv:1910.09772v1*, 2019.

[86] A. Siarohin, S. Lathuiliere, S. Tulyakov, E. Ricci, and N. Sebe. Animating arbitrary objects via deep motion transfer. *CVPR*, 2019.

[87] N. Siddharth, B. Paige, V. de Meent, A. Desmaison, F. Wood, N. D. Goodman, P. Kohli, and P. H. Torr. Learning disentangled representations with semi-supervised deep generative models. *arXiv:1706.00400*, 2017.

[88] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. *arXiv:1502.04681v3*, 2016.

[89] K. Stelzner, R. Peharz, and K. Kersting. Faster attend-infer-repeat with tractable probabilistic models. *Proc. 36th International Conference on Machine Learning*, 2019.

[90] R. Suter, D. Miladinovic, S. Bauer, and B. Scholkopf. Interventional robustness of deep latent variable models. *arXiv:1811.00007v1*, 2018.

[91] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *arXiv:1409.3215*, 2014.

[92] P. Toth, D. J. Rezende, A. Jaegle, S. Racaniere, A. Botev, and I. Higgins. Hamiltonian generative networks. *arXiv:1909.13789v1*, 2019.

[93] S. Tulyakov, M-Y. Liu, and J. Kautz. MoCoGAN: decomposing motion and content for video generation. *arXiv:1707.04993v2*, 2017.

[94] A. Vahdat and J. Kautz. NVAE: A deep hierarchical variational autoencoder. *34th Conference on Neural Information Processing Systems*, 2020.

[95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *31st Conference on Neural Information Processing Systems*, 2017.

[96] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *NeurIPS*, 2016.

[97] M. J. Vowels, N.C. Camgoz, and R. Bowden. Targeted VAE: Structured inference and targeted learning for causal parameter estimation. *Under Review*, 2020.

[98] M. J. Vowels, N. C. Camgoz, and R. Bowden. Gated variational autoencoders: Incorporating weak supervision to encourage disentanglement. *arXiv:1911.06443v1*, 2019.

[99] M. J. Vowels, N. C. Camgoz, and R. Bowden. NestedVAE: Isolating common factors via weak supervision. *Conference on Computer Vision and Pattern Recognition*, 2020.

[100] I. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. G3AN: disentangling appearance and motion for video generation. *CVPR*, 2020.

[101] M. Watter, J. T. Springenberg, M. Riedmiller, and J. Boedecker. Embed to control: a locally linear latent dynamics model for control from raw images. *arXiv:1506.07365v3*, 2015.

[102] D. Weissenborn, O. Tackstrom, and J. Uszkoreit. Scaling autoregressive video models. *arXiv:1906.02634*, 2019.

[103] Z. Xiao, K. Kreis, J. Kautz, and A. Vahdat. VAEBM: A symbiosis between variational autoencoders and energy-based models. *arXiv:2010.00654v1*, 2020.

[104] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *CVPR*, 2017.

[105] M. Yamada, H. Kim, K. Miyoshi, T. Iwata, and H. Yamakawa. Disentangled representations for sequence data using informationa bottleneck principle. *Proceedings of Machine Learning Research*, 2020.

[106] X. Yan. *Learning deep controllable and structured representations for image synthesis, structured prediction and beyond*. PhD thesis, University of Michigan, 2019.

[107] T. Ye, X. Wang, J. Davidson, and A. Gupta. Interpretable intuitive physics model. *arXiv:1808.10002v1*, 2018.

[108] H. Yin, F. S. Melo, A. Billard, and A. Paiva. Learning variational latent dynamics: towards model-based imitation and control. n.d.

[109] H. Zhang, S. Starke, T. Komura, and J. Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), 2018.

[110] R. Zhang. Making convolutional networks shift-invariant again. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[111] Y. Zhu, M. R. Min, A. Kadav, and H.P. Graf. S3VAE: self-supervised sequential VAE for representation disentanglement and data generation. *CVPR*, 2020.