LONG-HORIZON VIDEO PREDICTION USING A DYNAMIC LATENT HIERARCHY

Anonymous authors
Paper under double-blind review

ABSTRACT

The task of video prediction and generation is known to be notoriously difficult, with the research in this area largely limited to short-term predictions. Though plagued with noise and stochasticity, videos consist of features that are organised in a spatiotemporal hierarchy, different features possessing different temporal dynamics. In this paper, we introduce *Dynamic Latent Hierarchy* (DLH) – a deep hierarchical latent model that represents videos as a hierarchy of latent states that evolve over separate and fluid timescales. Each latent state is a mixture distribution with two components, representing the immediate past and the predicted future, causing the model to learn transitions only between sufficiently dissimilar states, while clustering temporally persistent states closer together. Using this unique property, DLH naturally discovers the spatiotemporal structure of a dataset and learns disentangled representations across its hierarchy. We hypothesise that this simplifies the task of modeling temporal dynamics of a video, improves the learning of long-term dependencies, and reduces error accumulation. As evidence, we demonstrate that DLH outperforms state-of-the-art benchmarks in video prediction, is able to better represent stochasticity, as well as to dynamically adjust its hierarchical and temporal structure. Our paper shows, among other things, how progress in representation learning can translate into progress in prediction tasks.

1 INTRODUCTION

Video data is considered to be one of the most difficult modalities for generative modelling and prediction, characterised by high levels of noise, complex temporal dynamics, and inherent stochasticity. Even more so, modelling long-term videos poses a significant challenge due to the problem of sequential error accumulation, largely restricting the research in this topic to short-term predictions.

Deep learning has given rise to generative latent-variable models with the capability to learn rich latent representations, allowing to model high-dimensional data by means of more efficient, lower-dimensional states (Kingma & Welling, 2014; Higgins et al., 2022; Vahdat & Kautz, 2020; Rasmus et al., 2015). Here, of particular interest are hierarchical latent models, which possess an extra degree of representational power. Employing hierarchies has so far proved to be an effective method for generating high-fidelity visual data, as well as concurrently producing more meaningful and disentangled latent representations in both static (Vahdat & Kautz, 2020) and temporal (Zakharov et al., 2022) datasets.

Unlike images, videos possess a *spatiotemporal* structure, in which a collection of spatial features adhere to the intrinsic temporal dynamics of a dataset – often evolving at different and fluid timescales. For instance, consider a simplistic example shown in Figure 1, in which the features of a video sequence evolve within a strict temporal hierarchy: from the panda continuously changing its position to the background elements being static over the entire duration of the video.

Discovering such a temporal structure in videos complements nicely the research into hierarchical generative models, which have been shown capable of extracting and disentangling features across a hierarchy of latent states. Relying on this notion of inherent spatiotemporal organisation of features, several hierarchical architectures have been proposed to either enforce a generative temporal hierarchy explicitly (Saxena et al., 2021), or discover it in an unsupervised fashion (Kim et al., 2019; Zakharov et al., 2022). In general, these architectures consist of a collection of latent states that

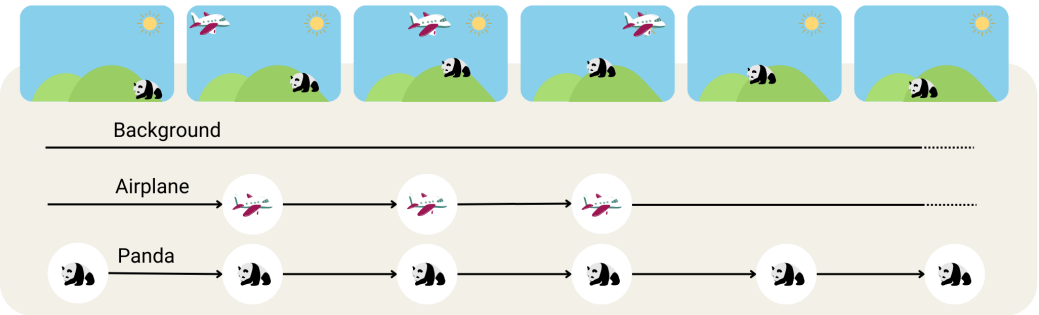

Figure 1: Videos can be viewed as a collection of features organised in a spatiotemporal hierarchy. This graphic illustrates a sequence of frames, in which the components of the video possess different temporal dynamics (white circles indicate feature changes). Notice the irregularities in their dynamics – the panda continuously changes its position, the airplane is seen only for a few timesteps, while the background remains static throughout. Similar to this, our model learns hierarchically disentangled representations of video features with the ability to model their irregular dynamics.

transition over different timescales, which has been shown to benefit long-term predictions (Saxena et al., 2021; Zakharov et al., 2022).

Building upon these notions, we propose an architecture of a hierarchical generative model for long-horizon video prediction – *Dynamic Latent Hierarchy* (DLH). The principle ideas underlying this work are two-fold. First, we posit that learning disentangled hierarchical representations and their separate temporal dynamics increases the model's expressivity and breaks down the problem of video modelling into simpler sub-problems, thus benefiting prediction quality. As such, our model is capable of discovering the appropriate hierarchical spatiotemporal structure of the dataset, seamlessly adapting its generative structure to a dataset's dynamics. Second, the existence of a spatiotemporal hierarchy, in which some features can remain static for an arbitrary period of time (e.g. background in Fig. 1), implies that predicting the next state at every timestep may introduce unnecessary accumulation of error and computational complexity. Instead, our model learns to transition between states only if a change in the represented features has been observed (e.g. airplane in Fig. 1). Conversely, if no change in the features has been detected, the model clusters such temporally-persistent states closer together, thus building a more organised latent space.

2 Dynamic Latent Hierarchy

We propose an architecture of a hierarchical latent model for video prediction - *Dynamic Latent Hierarchy*. DLH consists of a hierarchy of latent states that evolve over different and flexible timescales. Each latent state is a mixture of two Gaussian components that represent the *immediate past* and the *predicted future* in a single belief state. Using this formalisation, the model dynamically assigns every newly inferred posterior state to one of these clusters, and thus implicitly learns the temporal hierarchy of the data in an unsupervised fashion.

2.1 Generative model

We consider sequences of observations, $\{\mathbf{o}_1, ..., \mathbf{o}_T\}$, modelled by a hierarchical generative model with a joint distribution in the form (Fig. 2),

$$\prod_{t=1}^{T} p(\mathbf{o}_t, \vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t) = \prod_{t=1}^{T} p(\mathbf{o}_t|\vec{\mathbf{s}}_t) \prod_{n=1}^{N} p(\mathbf{s}_t^n| \underbrace{\mathbf{e}_t^n}_{\text{indicator}}, \underbrace{\mathbf{s}_{<t}^n}_{\text{temporal}}, \underbrace{\mathbf{s}_t^{>n}}_{\text{hierarchical}})p(\mathbf{e}_t^n|\mathbf{e}_{<t}^n, \mathbf{s}_{<t}^n), \tag{1}$$

where $\mathbf{s}_t^n \sim \mathcal{N}(\cdot, \cdot)$ is a diagonal Gaussian latent state, $\mathbf{e}_t^n \sim \text{Ber}(\cdot)$ is the corresponding Bernoulli variable at a hierarchical level n and timestep t, while $\vec{\mathbf{s}} = \{\mathbf{s}^1, ..., \mathbf{s}^N\}$ and $\vec{\mathbf{e}} = \{\mathbf{e}^1, ..., \mathbf{e}^N\}$ denote collections of all variables in a hierarchy. Notice that each state $\mathbf{s}_t^n$ is conditioned on all of the hierarchical states above, past states in the same level, and an indicator variable $\mathbf{e}_t^n$.

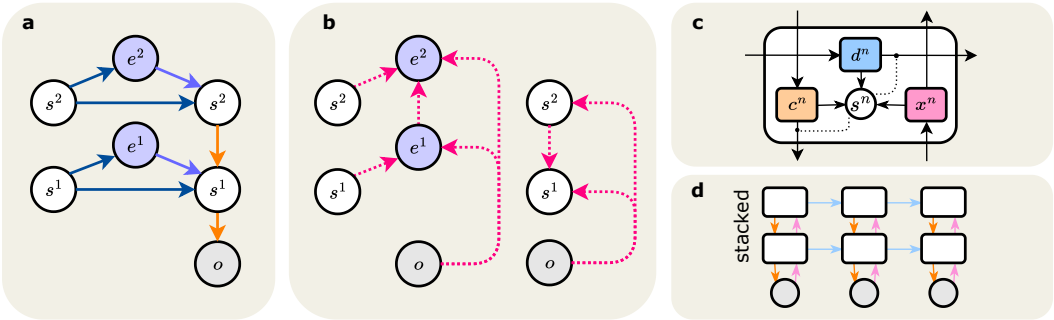

Figure 2: **(a)** Generative model of DLH. **(b)** Inference models of DLH. **(c)** Architectural components of DLH, showing the deterministic variables that mediate bottom-up, top-down, and temporal information. **(d)** Example of a two-level DLH model rolled-out over three timesteps (each white rectangle indicating a block from (c)).

One of the key features of DLH is the representation of a latent state as a temporal mixture of Gaussians (MoG). In particular, variables $\mathbf{s}_t^n$ and $\mathbf{e}_t^n$ together define a MoG, $p(\mathbf{s}_t^n, \mathbf{e}_t^n) = p(\mathbf{s}_t^n|\mathbf{e}_t^n)p(\mathbf{e}_t^n)$[1], with just two components such that,

$$p(\mathbf{s}_t^n|\mathbf{e}_t^n) = \begin{cases} p(\mathbf{s}_{t-1}^n) & \text{if } \mathbf{e}_t^n = 0, \text{ (previous state: } \textit{static prior}) \\ p_\theta(\mathbf{s}_t^n|\mathbf{s}_{<t}^n) & \text{if } \mathbf{e}_t^n = 1, \text{ (predicted state: } \textit{change prior}). \end{cases} \tag{2}$$

As such, at every timestep, DLH holds two prior beliefs over the state of the world: (1) it can remain *static*, or (2) it can progress through time and thus *change*. In this view, variable $\mathbf{e}_t^n$ can be informally described as the probability of whether state $\mathbf{s}_t^n$ should be updated or remain fixed at timestep t (Fig. 3). This property allows DLH to model the data as a collection of hierarchical latent states that evolve over different and flexible timescales, determined by the indicator variable $\mathbf{e}^n$.

2.2 Inference

In order to train the model using a variational lower bound, we must estimate the posterior distribution $q(\mathbf{s}_t^n, \mathbf{e}_t^n|\mathbf{o}_t)$, for which we assume a mean-field factorisation $q(\mathbf{s}_t^n)q(\mathbf{e}_t^n)$; therefore, the two distributions are approximated separately.

Estimating $q(\mathbf{s})$ In DLH, posterior $q(\mathbf{s}_t^n)$ is assumed to be an isotropic Gaussian, amortised using a neural network $q_\psi(\mathbf{s}_t^n|\mathbf{s}_t^{>n}, \mathbf{o}_t)$ with parameters ψ and conditioned on hierarchical states above and the latest data point $\mathbf{o}_t$. In line with the established procedure, the approximate posterior is trained using the reparametrisation trick (Kingma & Welling, 2014).

Estimating $q(\mathbf{e})$ Using reparametrisation tricks for discrete latent variables poses a significant challenge for a stable training procedure of deep learning models, which can be further exacerbated in hierarchical models (Falck et al., 2021). To avoid this, we estimate $q(\mathbf{e})$ using a non-parametric method.

Inferring distribution $q(\mathbf{e}^n)$ can be conceptualised as a clustering problem of $q(\mathbf{s}^n)$ with respect to the *static* and *change* priors of the model, with the central question being: under which temporal mixture component in Eq. 2 is the inferred state most likely? Has the state of the world changed or has it remained the same?

As such, similar to the event detection method in Zakharov et al. (2022), we formulate the approximation of $p(\mathbf{e}|\mathbf{s})$ as model selection using expected likelihood ratio, where the two components of the MoG (Eq. 2) are the competing models. Under the inferred state, $q_\psi(\mathbf{s}^n)$, the expected loglikelihood ratio is,

$$\mathbb{E}[\log \Lambda(\mathbf{s}^n)] = \mathbb{E}_{q_\psi(\mathbf{s}^n)} \log \frac{p(\mathbf{s}^n|\mathbf{e}^n = 0)}{p(\mathbf{s}^n|\mathbf{e}^n = 1)} \tag{3}$$

[1]Stripping away the hierarchical and temporal conditioning for clarity.

Assuming the selection of the most likely component under the inferred posterior, we come to the following selection criterion,

$$D_{\mathrm{KL}}[q_\psi(\mathbf{s}^n)||p(\mathbf{s}^n|\mathbf{e}^n=1)] \underset{\mathbf{e}^n=0}{\overset{\mathbf{e}^n=1}{\lessgtr}} D_{\mathrm{KL}}[q_\psi(\mathbf{s}^n)||p(\mathbf{s}^n|\mathbf{e}^n=0)], \tag{4}$$

where the most likely component i is approximated to have a probability $q(\mathbf{e}^n = i) = 1$. This approximation relates to the VaDE trick, which is similarly a non-parametric method of estimating the posterior component variable of a MoG (Jiang et al., 2017; Falck et al., 2021). In particular, our method can be viewed as taking a sample from the most likely component of the VaDE-estimated $q(\mathbf{e}^n)$ under the assumption of equal prior probabilities (see Appendix B). Though this method introduces bias, in practice, we found that it performs better than the VaDE trick. We hypothesise that this relates to a relatively fast convergence of the $p_\theta(\mathbf{e}^n)$ model, which becomes overly confident in its predictions (even before any video features have been learned), thus irreversibly skewing the approximation of $q(\mathbf{e}^n)$. Nevertheless, we believe this direction of future work may merit further investigation.

2.3 Nested timescales

We add a constraint on the hierarchical temporal structure of the generative model similar to Zakharov et al. (2022); Saxena et al. (2021); Kim et al. (2019). In particular, $q(\mathbf{e}^{n+1}|\mathbf{e}^n = 0) = 0$. Enforcing the constraint of nested timescales has been shown to be an effective method to promote spatiotemporal disentanglement of features in hierarchical models (Zakharov et al., 2022), encouraging the representation of progressively slower features in the higher levels of the model. Furthermore, to reduce the computational complexity of the model, we block any further inference above the hierarchical level where $\mathbf{e}^n = 0$ is inferred, such that:

$$\text{if } \mathbf{e}_t^{n-1} = 0, \text{ then } \mathbf{e}_t^n = 0 \text{ and } q(\mathbf{s}_t^n) \leftarrow q(\mathbf{s}_{t-1}^n). \tag{5}$$

Lastly, to model continuously changing videos, we assume $q(\mathbf{e}^1 = 1) = 1$, which allows for the bottom level of DLH to always be in use. It is worth noting that all these three model constraints can be relaxed in different implementations of DLH, which may be explored in future work.

2.4 Lower bound estimation

To train the model, we derive a variational lower bound (ELBO), for which we introduce an approximate posterior distribution $q(\vec{\mathbf{s}}, \vec{\mathbf{e}})$ so that,

$$\sum_{t=1}^{T} \log p(\mathbf{o}_t) = \sum_{t=1}^{T} \log \int_{\vec{\mathbf{s}}} \sum_{\vec{\mathbf{e}}} q(\vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t) \frac{p(\mathbf{o}_t, \vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t)}{q(\vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t)} \tag{6}$$

$$\geq \sum_{t=1}^{T} \mathbb{E}_{q(\vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t)} \log p(\mathbf{o}_t|\vec{\mathbf{s}}_t) + \mathbb{E}_{q(\vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t)} \Big[\log \frac{p(\vec{\mathbf{s}}_t|\vec{\mathbf{e}}_t) p(\vec{\mathbf{e}}_t)}{q(\vec{\mathbf{s}}_t, \vec{\mathbf{e}}_t)} \Big], \tag{7}$$

where we omitted temporal and hierarchical conditioning for notational simplicity. Assuming posterior factorisation of $q(\mathbf{s}_t^n, \mathbf{e}_t^n) = q(\mathbf{s}_t^n) q(\mathbf{e}_t^n)$, we write the complete formulation of the ELBO,

$$\mathcal{L}_{\text{ELBO}} = \sum_{t=1}^{T} \Big[\mathbb{E}_{q(\vec{\mathbf{s}}_t)} \log p(\mathbf{o}_t|\vec{\mathbf{s}}_t) \Big] \tag{8a}$$

$$- \sum_{t=1}^{T} \sum_{n=1}^{N} \Big[\mathbb{E}_{q(\mathbf{e}_t^n) q(\mathbf{s}_{<t}^n, \mathbf{s}_t^{>n})} D_{\mathrm{KL}} \big[q(\mathbf{s}_t^n) || p(\mathbf{s}_t^n | \mathbf{e}_t^n, \mathbf{s}_{<t}^n, \mathbf{s}_t^{>n}) \big] \Big] \tag{8b}$$

$$- \sum_{t=1}^{T} \sum_{n=1}^{N} \Big[\mathbb{E}_{q(\mathbf{s}_{<t}^n)} D_{\mathrm{KL}} \big[q(\mathbf{e}_t^n) || p(\mathbf{e}_t^n | \mathbf{s}_{<t}^n) \big] \Big]. \tag{8c}$$

To better understand the optimisation objective and the role of a temporal Gaussian mixture from Eq. 2, it is useful to break the down the three components of the ELBO. First, component 8a is the

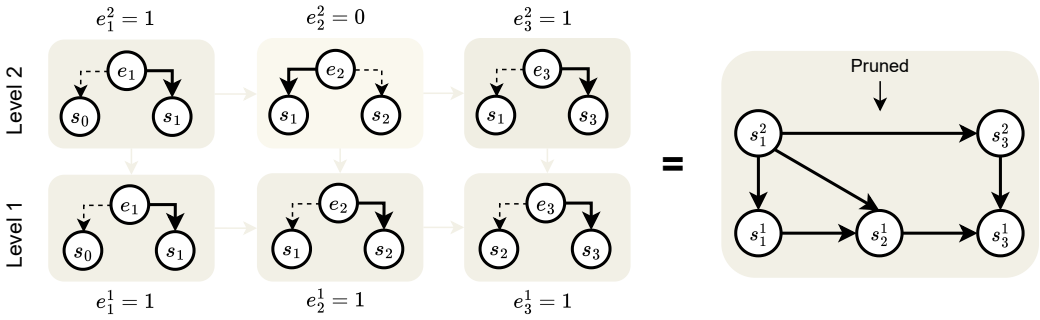

Figure 3: Sampling MoG components can be seen as changing the generative structure of the model. The diagram shows a two-level DLH unrolled over three timesteps. Bold arrows indicate the sampled component of a MoG. At $n, t = 2$, the model samples the static component $0 \sim p(\mathbf{e}_2^2)$, thus state $\mathbf{s}_1^2$ remains fixed before being updated to $\mathbf{s}_3^2$ (indicated by the 'Pruned' label on the right).

likelihood of the data under the inferred posteriors $q(\vec{\mathbf{s}}_t)$, which improves the quality of frame reconstructions. Second, component 8c is the KL divergence between the posterior and prior Bernoulli distributions over $\mathbf{e}$, allowing the parametrised prior model to learn the evolution of static and change priors over time. Lastly, component 8b regularises the latent belief space by bringing the posterior either closer to the static or to the change component of a prior MoG. This can be seen more clearly if we expand the expectation (removing temporal and hierarchical conditioning for simplicity),

$$= q(\mathbf{e}_t^n = 0)\, \underbrace{D_{\mathrm{KL}}\big[q(\mathbf{s}_t^n)||p(\mathbf{s}_t^n|\mathbf{e}_t^n = 0)\big]}_{\text{posterior} \leftrightarrow \text{static prior}} + q(\mathbf{e}_t^n = 1)\, \underbrace{D_{\mathrm{KL}}\big[q(\mathbf{s}_t^n)||p(\mathbf{s}_t^n|\mathbf{e}_t^n = 1)\big]}_{\text{posterior} \leftrightarrow \text{change prior}}. \tag{9}$$

Depending on the inferred posterior distribution $q(\mathbf{e}_t^n)$, the model will employ the appropriate part of Eq. 9 in the optimisation. For example, if inferred that state $\mathbf{s}_t^n$ has not changed ($\mathbf{e}_t^n = 0$), the model will naturally bring the new posterior and the static prior closer together, and vice versa. Ultimately, this allows the model to naturally cluster similar temporal states together, while learning to transition between states that are sufficiently separated in the belief space.

2.5 Model components

DLH's architecture is similar to that of VPR (Zakharov et al., 2022) – an event-based hierarchical generative model capable of learning the temporal structure of video datasets. The architecture of VPR is particularly appealing given its distinct separation of channels of information flow by means of deterministic variables. This gives the model more representational power and provides better disentangling properties along its hierarchy (Zakharov et al., 2022).

More specifically, DLH consists of the following model components,

Encoder,	$x_t^{n+1} = f_{\text{enc}}^n(x_t^n)$	(10)	Posterior,	$q_\psi(\mathbf{s}_t^n \| x_\tau^n, \mathbf{s}_t^{>n})$	(13)
Decoder,	$c_t^{n-1} = f_{\text{dec}}^n(\mathbf{s}_t^n, c_t^n)$	(11)	Prior state,	$p_\theta(\mathbf{s}_t^n \| \mathbf{s}_{<t}^n, \mathbf{s}_t^{>n})$	(14)
Temporal,	$d_{t+1}^n = f_{\text{tem}}^n(\mathbf{s}_t^n, d_t^n)$	(12)	Prior factor,	$p_\theta(\mathbf{e}_t^n \| \mathbf{e}_{<t}^n, \mathbf{s}_{<t}^n)$	(15)

where deterministic variables x_t^n, c_t^n, d_t^n correspond to the bottom-up, top-down, and temporal variable transformations, as shown in Figure 2c. Variables c_t^n and d_t^n are non-linear transformations of samples from $\mathbf{s}_t^{>n}$ and $\mathbf{s}_{<t}^n$, respectively. We use a GRU model (Cho et al., 2014) for the transition model f_{tem} and fully-connected MLP layers for all other models.

3 Related work

Video prediction Early works in video prediction largely focused on different variants of deterministic models (Oh et al., 2015; Finn et al., 2016; Byravan & Fox, 2017; Vondrick & Torralba, 2017); however, it has been widely suggested that these models are poorly suited for capturing stochasticity that is often present in video datasets.

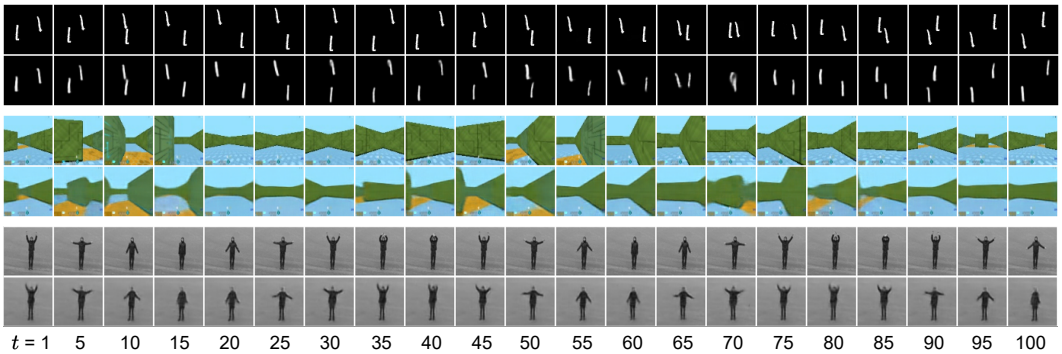

Figure 4: Open-loop video prediction with 30 context frames. DLH maintains the important contextual information about the video, without significant degeneration in the reconstruction quality.

The problem of stochastic video prediction has been addressed using a variety of generative architectures. Models autoregressive in image space (Babaeizadeh et al., 2021; Reed et al., 2017; Weissenborn et al., 2020; Kalchbrenner et al., 2016; Denton & Fergus, 2018) demonstrate good results but suffer from high computational complexity, particularly for long-term predictions. GAN-based (Goodfellow et al., 2014) approaches have been popular due to their ability to produce sharp predictions (Clark et al., 2019; Arr; Mathieu et al., 2016; Lee et al., 2018). More recently, transformers (Vaswani et al., 2017) have been used to model video datasets, both in latent (Rakhimov et al., 2020; Yan et al., 2021) and pixel space (Weissenborn et al., 2020). A fairly large category of video architectures is based on using Variational Autoencoders (Kingma & Welling, 2014), which have been shown to produce meaningful latent representations on image (Vahdat & Kautz, 2020; Higgins et al., 2022) and video data (Zakharov et al., 2022). Variational autoencoders (VAE)-based models that attempt to learn temporal dependencies in the latent space (Wu et al., 2021; Villegas et al., 2019; Castrejon et al., 2019; Franceschi et al., 2020; Saxena et al., 2021; Yan et al., 2021; Zakharov et al., 2022) generate good performance but generally suffer from blurry predictions, referred to as the 'underfitting problem' (Babaeizadeh et al., 2021; Wu et al., 2021; Villegas et al., 2019). Nevertheless, these models benefit from computational efficiency since the learning of temporal video dynamics commonly happens in a lower-dimensional latent space. Most recently, diffusion models have been shown to produce great performance on both short (Yang et al., 2022; Höppe et al., 2022) and long (Harvey et al., 2022) videos.

Hierarchical generative models Hierarchical generative models have been shown to be an effective way of modelling high-dimensional data, including images (Rasmus et al., 2015; Sønderby et al., 2016; Maaløe et al., 2019; Vahdat & Kautz, 2020; Child, 2021) and videos (Saxena et al., 2021; Kim et al., 2019; Pertsch et al., 2020; Hsu et al., 2019; Zakharov et al., 2022), producing rich latent representations and demonstrating strong representational power.

Temporal abstraction The topic of learning temporal abstractions from sequential data has been harmoniously rising in popularity alongside the progress in deep and hierarchical latent models. Temporal abstraction models often operate a number of hierarchical latent variables updating over different timescales, with the goal of capturing the different components of a dataset's temporal structure (Chung et al., 2017; Mujika et al., 2017; Kim et al., 2019; Saxena et al., 2021; Fountas et al., 2022; Zakharov et al., 2022), though other proposals learn the relevant prediction timescales without resorting to hierarchical methods (Chung et al., 2017; Neitz et al., 2018; Jayaraman et al., 2018; Shang et al., 2019; Kipf et al., 2019; Kim et al., 2019; Pertsch et al., 2020; Zakharov et al., 2021)

Gaussian Mixtures in VAEs Our work similarly touches on the topic of VAEs with Gaussian Mixture latent states. Generally, these models are aimed at producing meaningful structure of the latent space, in which data points are clustered in an unsupervised fashion (Dilokthanakul et al., 2016; Jiang et al., 2017; Falck et al., 2021). Though highly relevant conceptually (unsupervised clustering), these works deal with non-temporal data and therefore have fundamentally different formulations.

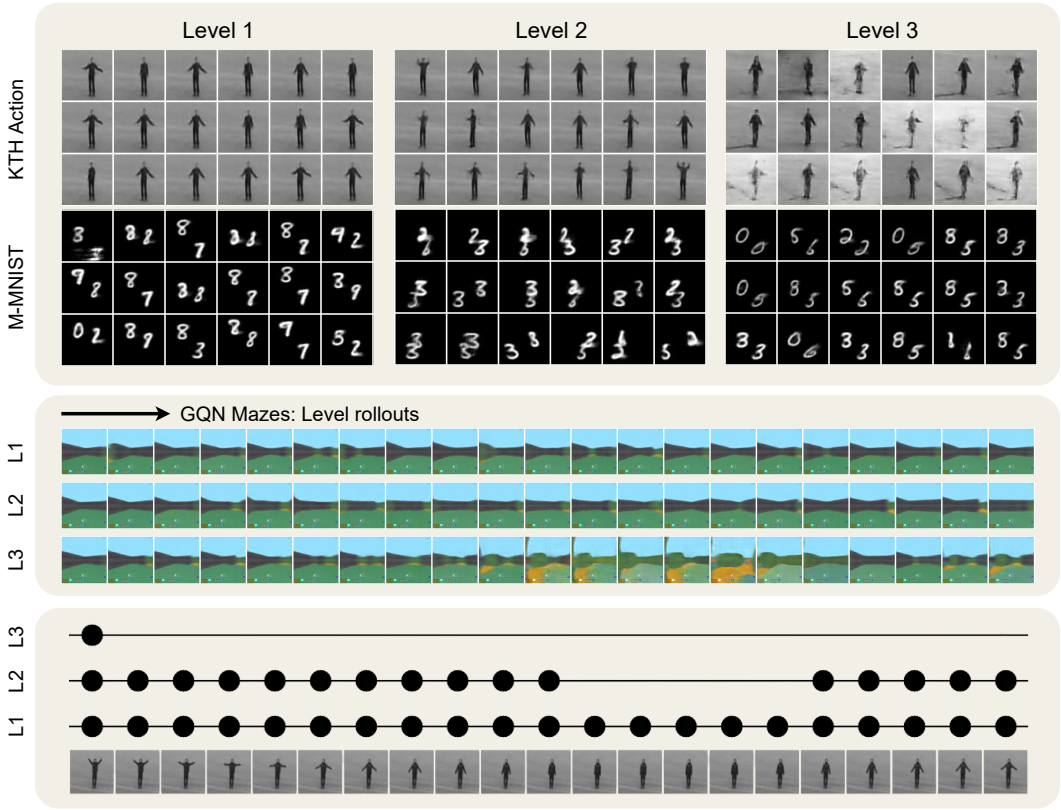

Figure 5: Hierarchical disentanglement in DLH. (Top): Random samples drawn from the different hierarchical levels (other levels are fixed). In KTH, L1 and L2 tend to encode motion; L3 encodes the general context of the frame. In MNIST, L1 encodes slight variations in the position and digits; L2 represents position; L3 contains both style and digit types. (Middle): Rolling out hierarchical levels in the DML Mazes (other levels are fixed). Level 1 includes minor variations in the view angle; L2 changes the position of the observer and wall shape; L3 predicts transitions to the different parts of the maze. (Bottom): Inferred components of the temporal MoGs at every level of the model. Black circles indicate $\mathbf{e}^n = 1$ (change component) inferred by DLH. L2 updates only when the person's arms are in motion. Similarly, L3 remains static throughout the duration of the video.

4 Experiments

In this section, we showcase the representational properties of DLH, and their resulting impact on the performance of the model for long-term video prediction. In particular, we demonstrate that DLH: (a) outperforms benchmarks in long-term video prediction, (b) produces an organised hierarchical latent space with spatiotemporal disentanglement and temporal abstraction, (c) generates coherent videos even in datasets characterised by temporal stochasticity, and (d) dynamically regulates its structural complexity. In the analysis probing DLH's expressivity and representations, we emphasise how the presented formulation of the generative model, in particular the use of temporal MoG, naturally results in the emergent properties of the model.

4.1 Datasets and benchmarks

Datasets To test the model's ability in long-term video prediction, we use Moving MNIST (Srivastava et al., 2016) with 300 timesteps, KTH Action (Schuldt et al., 2004) with 300 timesteps, and DeepMind Lab Mazes (Eslami et al., 2018) with 200 timesteps. For a more detailed analysis of the model's properties, we use a toy Moving Ball dataset (Zakharov et al., 2022).

Benchmarks Clockwork Variational Autoencoder (CWVAE) (Saxena et al., 2021) is a hierarchical VAE for video prediction, in which latent variables operate over fixed-temporal schedules, similarly subject to nested timescales. CW-VAE demonstrated state-of-the-art performance in long-term video prediction, indicating the merit of the slower-evolving context states. VTA (Kim et al., 2019) is a two-level hierarchical model for video prediction that employs a parametrised boundary detector to learn sub-sequences of a video and generate temporally-abstracted representations. LMC-Memory (Lee et al., 2021) learns and stores long-term motion context for better long-horizon video prediction, which has been shown to outperform other RNN-based approaches.

Table 1: Open-loop video prediction

Moving MNIST	SSIM↑	PSNR↑
DLH (Ours)	**0.76***	**15.1***
CWVAE	0.68*	13.11*
VTA	0.58	12.18
LMC-Memory	0.75*	13.73*
KTH Action	SSIM↑	PSNR↑
DLH (Ours)	**0.84***	**24.7**
CWVAE	0.80	22.0
VTA	0.77	22.41
LMC-Memory	0.83*	23.44
DML Mazes	SSIM↑	PSNR↑
DLH (Ours)	**0.59**	**14.3**
CWVAE	0.44	13.71*
VTA	0.55	13.51*

Metrics To evaluate stochastic video prediction, we employ the standard procedure of sampling 100 conditionally generated sequences and picking the best one to report (Denton & Fergus, 2018). For metrics, we use Structural Similarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR) to test the performance of a model with respect to the ground-truth videos.

4.2 Video prediction and generation

Table 1 shows the evaluation of DLH and its benchmarks in the task of long-horizon video prediction. As evident, DLH outperforms other models across all of the presented datasets. Figure 4 shows some examples of long-horizon open-loop rollouts. For Moving MNIST, DLH maintains the information about the digits throughout the sequence, while also accurately predicting their positions. For DML Mazes, DLH correctly predicts the colours and wall positions, without switching to a configuration of another maze. Similarly, for KTH, our model preserves the important contextual knowledge (e.g. background) and accurately predicts the long sequence of arm swings.

4.3 Hierarchical abstraction

DLH exhibits characteristics of a model that learns temporally-abstracted and hierarchically disentangled representations. Figure 5 demonstrates reconstructed frames retrieved by sampling the different hierarchical levels of the model. Here, we observe the variations in the samples that correspond to meaningful and interpretable spatiotemporal features of the videos. In the same figure, we show rollouts of the model's levels (with all other levels being fixed) using the DML Mazes dataset, which indicate that DLH learns to transition between progressively slower features in the higher levels of its hierarchy.

Similarly, the bottom sub-figure demonstrates another telling qualitative evaluation of DLH and its representations – the inferred components of $\mathbf{e}^n$ (static or change) for a given video. In particular, it shows that the model continuously detects feature changes in the second level of its hierarchy (L2) when the person in the frame is moving their arms, and conversely when the person's arms remain static. Furthermore, it can be seen that the the top level (L3) remains static throughout. Notably, these results are in agreement with the random samples shown above, and more clearly illustrate the property of hierarchical disentanglement present in the model.

Table 2: Average number of employed levels ($\bar{L}$) and the total KL loss in the instances of DLH with different number of hierarchical levels (Moving Ball)

Levels	$\bar{L}$	KL loss
2	1.22 ± 0.05	41.9 ± 0.3
3	1.24 ± 0.05	42.8 ± 1.1
4	1.32 ± 0.12	41.1 ± 1.0
5	1.38 ± 0.07	43.1 ± 2.1

The capacity of DLH to learn the spatiotemporal representation of features along its hierarchy is largely driven by the dynamic manipulation of its hierarchical and temporal structure. Interestingly, we observe that DLH consistently converges to similar structures even when possessing different number of levels. Table 2 shows the average hierarchical depth employed by the model ($\bar{L}$) over a video length given the total number of hierarchical levels it has (trained using the Moving Ball

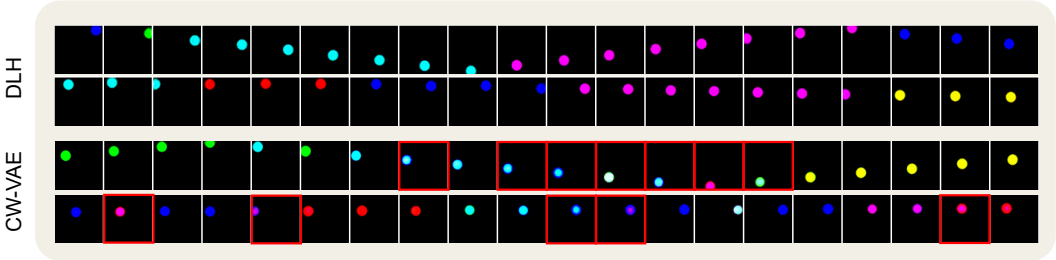

Figure 6: Open-loop rollouts in a stochastic Moving Ball dataset using DLH (top) and CW-VAE (bottom) models. DLH successfully produces realistic rollouts, while CW-VAE struggles to produce rollouts with sharp and random colour changes (frames highlighted in red).

dataset). As evident, the models converge to similar values despite their size differences, indicating that DLH naturally simplifies its structure and does not employ more resources than necessary. This is similarly substantiated by the comparable magnitudes of the total KL loss, which is commonly used to indicate the amount of information stored in the latent states.

4.4 Temporal stochasticity and dynamic structure

Videos often contain temporal stochasticity, where features may change at seamingly random times. How would a generative model represent such uncertainty? In the context of employing a Gaussian latent state, the uncertainty about the next state would have to be reflected in the higher variance, in order to cover both possible outcomes; however, this necessarily increases the chance of sampling areas of the latent space that do not correspond to any meaningful states, harming the prediction performance. In DLH, by virtue of the temporal MoG, such stochasticity can be effectively captured by variable $\mathbf{e}^n$, which decides whether the latent state $\mathbf{s}^n$ should be updated or remain fixed, alleviating the need to sample from degenerate regions of the latent space. To demonstrate this, we modify the Moving Ball dataset to include random colour changes that can occur at every timestep with a probability of λ. Figure 6 shows a comparison of open-loop rollouts generated by DLH and CW-VAE, trained on the Moving Ball with $\lambda = 0.1$. While CW-VAE struggles to generate rollouts with consistent and sharp colour changes, DLH faces no such problems, producing sequences with both deterministic and random colour switches.

Table 3: Average predicted probability of $p_\theta(\mathbf{e} = 1)$ under the different levels of temporal stochasticity (λ) in the Moving Ball dataset

λ	$p_\theta(\mathbf{e} = 1)$ change	$p_\theta(\mathbf{e} = 1)$ static
0.0	$.97 \pm 0.01$	$.007 \pm .003$
0.1	$.82 \pm 0.02$	$.074 \pm .007$
0.3	$.73 \pm 0.02$	$.167 \pm .011$

The behaviour of prior $p_\theta(\mathbf{e})$ under temporal stochasticity can be more clearly understood using the results in Table 3, which shows the average predicted probability of the change component under the inferred posterior $q(\mathbf{e})$ being either *change* ($\mathbf{e} = 1$) or *static* ($\mathbf{e} = 0$). As the stochasticity of the dataset, λ, rises, the model becomes more cautious in its predictions about the exact timing of when the video features will change.

5 Discussion

Our work demonstrates that building generative models with better representational properties, such as spatiotemporal and hierarchical disentanglement, translates to better predictive capabilities in long and complex time series. Furthermore, we believe that improving the quality of latent representations is of high importance for model-based reinforcement learning agents, where accurate predictions of the future lead to better planning and offline credit assignment, while a hierarchical and nested treatment of time could allow for temporally-abstract reasoning. Nevertheless, one of the limitations facing VAE-based models, and by extension our own, is the lack of sharpness in the predictions. Though significant progress has been made in the recent years (Babaeizadeh et al., 2021; Wu et al., 2021), addressing this problem in DLH can be a significant next step for further improving the performance of the model.

REFERENCES

ArrowGAN : Learning to generate videos by learning Arrow of Time | Elsevier Enhanced Reader.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in Pixel-Level Video Prediction, June 2021.

Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 173–180, 2017.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved Conditional VRNNs for Video Prediction, April 2019.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *ArXiv*, abs/2011.10650, 2021.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *International Conference on Learning Representations, ICLR*, 2017.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial Video Generation on Complex Datasets, September 2019.

Emily Denton and Rob Fergus. Stochastic Video Generation with a Learned Prior, March 2018.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, M. J. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *ArXiv*, abs/1611.02648, 2016.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, June 2018. doi: 10.1126/science.aar6170.

Fabian Falck, Haoting Zhang, Matthew Willetts, George Nicholson, Christopher Yau, and Christopher C. Holmes. Multi-facet clustering variational autoencoders. *ArXiv*, abs/2106.05241, 2021.

Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *ArXiv*, abs/1605.07157, 2016.

Zafeirios Fountas, Anastasia Sylaidi, Kyriacos Nikiforou, Anil K. Seth, Murray Shanahan, and Warrick Roseboom. A Predictive Processing Model of Episodic Memory and Time Perception. *Neural Computation*, 34(7):1501–1544, 06 2022. ISSN 0899-7667. doi: 10.1162/neco_a_01514. URL https://doi.org/10.1162/neco_a_01514.

Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic Latent Residual Video Prediction, August 2020.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible Diffusion Modeling of Long Videos, May 2022.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, July 2022.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion Models for Video Prediction and Infilling, August 2022.

Wei-Ning Hsu, Y. Zhang, Ron J. Weiss, H. Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Z. Chen, Jonathan Shen, P. Nguyen, and Ruoming Pang. Hierarchical generative modeling for controllable speech synthesis. *ArXiv*, abs/1810.07217, 2019.

Dinesh Jayaraman, Frederik Ebert, Alexei A Efros, and Sergey Levine. Time-agnostic prediction: Predicting predictable video frames. *arXiv preprint arXiv:1808.07784*, 2018.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 2017.

Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video Pixel Networks, October 2016.

Taesup Kim, Sungjin Ahn, and Yoshua Bengio. Variational temporal abstraction. *Advances in Neural Information Processing Systems*, 32:11570–11579, 2019.

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

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, May 2014.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic Adversarial Video Prediction, April 2018.

Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In *NeurIPS*, 2019.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error, February 2016.

Asier Mujika, Florian Meier, and Angelika Steger. Fast-slow recurrent neural networks. *arXiv preprint arXiv:1705.08639*, 2017.

Alexander Neitz, Giambattista Parascandolo, Stefan Bauer, and Bernhard Schölkopf. Adaptive skip intervals: Temporal abstraction for recurrent dynamical models. *arXiv preprint arXiv:1808.04768*, 2018.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-Conditional Video Prediction using Deep Networks in Atari Games, December 2015.

Karl Pertsch, Oleh Rybkin, Jingyun Yang, Shenghao Zhou, Konstantinos Derpanis, Kostas Daniilidis, Joseph Lim, and Andrew Jaegle. Keyframing the future: Keyframe discovery for visual prediction and planning. In *Learning for Dynamics and Control*, pp. 969–979. PMLR, 2020.

Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent Video Transformer, June 2020.

Antti Rasmus, Mathias Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, 2015.

Scott Reed, Aäron van den Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Dan Belov, and Nando de Freitas. Parallel Multiscale Autoregressive Density Estimation, March 2017.

Vaibhav Saxena, Jimmy Ba, and Danijar Hafner. Clockwork Variational Autoencoders, February 2021.

C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pp. 32–36 Vol.3, August 2004. doi: 10.1109/ICPR.2004.1334462.

Wenling Shang, Alex Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Learning world graphs to accelerate hierarchical reinforcement learning. *arXiv preprint arXiv:1907.00664*, 2019.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *NIPS*, 2016.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs, January 2016.

Arash Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder. *ArXiv*, abs/2007.03898, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks, November 2019.

Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2992–3000, 2017.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling Autoregressive Video Models, February 2020.

Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy Hierarchical Variational Autoencoders for Large-Scale Video Prediction, June 2021.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video Generation using VQ-VAE and Transformers, September 2021.

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion Probabilistic Modeling for Video Generation, May 2022.

Alexey Zakharov, Matthew Crosby, and Zafeirios Fountas. Episodic memory for subjective-timescale models. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

Alexey Zakharov, Qinghai Guo, and Zafeirios Fountas. Variational predictive routing with nested subjective timescales. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=JxFgJbZ-wft`.

A TRAINING AND ARCHITECTURAL DETAILS

For all datasets, DLH was trained with video sequences of length 100 and batch 100. We used Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.0005 and $\epsilon = 0.0001$. We also find that it is beneficial to multiply the KL loss with a parameter β, which is slowly increased to the value of 1.0 over the course of the first 10k training iterations. This promotes the model to learn good reconstructions before being severely restricted in increasing its latent complexity. We also optimise the standard deviation of the

For this paper, we used a three-level DLH for all datasets (except for Moving Ball, where we used just two levels). Each level possesses deterministic variables, for which we set $|x^n| = |d^n| = |c^n| = 200$, and random variables, where $|\mathbf{s}^n| = 20$. For encoding and decoding image data, we use 4 convolutional and 4 transposed-convolutional layers, respectively. Table 4 shows the architectural details of all other model components, including the sizes of neural networks. The total number of parameters of a three-level DLH is 7M, and its training takes $\sim$3 days on a single Tesla V100.

Table 4: Architecture of models from Eq.10-15.

Model	Hidden layers
Encoder, $\mathrm{f}^n_{\text{enc}}(x^n_t)$	MLP [200]
Decoder, $\mathrm{f}^n_{\text{dec}}(\mathbf{s}^n_t, c^n_t)$	MLP [200]
Temporal, $\mathrm{f}^n_{\text{tem}}(\mathbf{s}^n_t, d^n_t)$	GRU [200]
Posterior, $\mathrm{q}_\psi(\mathbf{s}^n_t \| x^n_\tau, \mathbf{s}^{>n}_t)$	MLP [40, 40, 40]
Prior state, $\mathrm{p}_\theta(\mathbf{s}^n_t \| \mathbf{s}^n_{<t}, \mathbf{s}^{>n}_t)$	MLP [40, 40, 40]
Prior factor, $\mathrm{p}_\theta(\mathbf{e}^n_t \| \mathbf{e}^n_{<t}, \mathbf{s}^n_{<t})$	GRU [200]

B RELATIONSHIP TO VADE TRICK

VaDE trick is a non-parametric technique for estimating posterior $p(\mathbf{e}|\mathbf{s})$ in Gaussian Mixture VAE models (Jiang et al., 2017; Falck et al., 2021). In particular, Falck et al. (2021) proved that an approximate posterior distribution $q(\mathbf{e})$ that minimises the KL divergence to the true posterior $p(\mathbf{e}|\mathbf{s})$ will take the form,

$$\arg\min_{q(\mathbf{e})} D_{\mathrm{KL}}[q(\mathbf{e})||p(\mathbf{e}|\mathbf{s})] = \frac{\exp(\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}|\mathbf{s}))}{\sum_{\mathbf{e}\in\mathbf{E}} \exp(\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}|\mathbf{s}))}, \tag{16}$$

where $K = |\mathbf{E}|$ is the number of components in a Gaussian mixture $p(\mathbf{s}, \mathbf{e})$, and $q(\mathbf{s})$ is an approximate posterior over $\mathbf{s}$.

In DLH, the number of components in a MoG is limited to $K = 2$. Using Eq. 16, we can compute the posterior odds and solve for one of the two components (e.g. $q(\mathbf{e} = 0)$),

$$\frac{q(\mathbf{e}=0)}{q(\mathbf{e}=1)} = \frac{\exp(\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}=0|\mathbf{s}))}{\exp(\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}=1|\mathbf{s}))} \tag{17}$$

$$= \exp\Big[\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}=0|\mathbf{s}) - \mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{e}=1|\mathbf{s})\Big] \tag{18}$$

$$= \exp\Big[\mathbb{E}_{q(\mathbf{s})} \log \frac{p(\mathbf{s}|\mathbf{e}=0)p(\mathbf{e}=0)}{\cancel{\sum_{\mathbf{e}\in\mathbf{E}} p(\mathbf{s}|\mathbf{e})p(\mathbf{e})}} - \mathbb{E}_{q(\mathbf{s})} \log \frac{p(\mathbf{s}|\mathbf{e}=1)p(\mathbf{e}=1)}{\cancel{\sum_{\mathbf{e}\in\mathbf{E}} p(\mathbf{s}|\mathbf{e})p(\mathbf{e})}}\Big] \tag{19}$$

$$= \exp\Big[\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{s}|\mathbf{e}=0)p(\mathbf{e}=0) - \mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{s}|\mathbf{e}=1)p(\mathbf{e}=1)\Big] \tag{20}$$

$$= \exp\Big[\mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{s}|\mathbf{e}=0) - \mathbb{E}_{q(\mathbf{s})} \log p(\mathbf{s}|\mathbf{e}=1) + \log \frac{p(\mathbf{e}=0)}{p(\mathbf{e}=1)}\Big] \tag{21}$$

Adding and subtracting $\mathbb{E}_{q(\mathbf{s})} \log q(\mathbf{s})$ inside the exponent,

$$= \exp\Big[D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=1)] - D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=0)] + \log \frac{p(\mathbf{e}=0)}{p(\mathbf{e}=1)}\Big]. \tag{22}$$

Finally, we use the fact that $\sum_{\mathbf{e}\in\mathbf{E}} q(\mathbf{e}) = 1$ to get $q(\mathbf{e}=0)$,

$$\frac{q(\mathbf{e}=0)}{1-q(\mathbf{e}=0)} = \exp\Big[D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=1)] - D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=0)] + \log\frac{p(\mathbf{e}=0)}{p(\mathbf{e}=1)}\Big], \tag{23}$$

and solving for $q(\mathbf{e}=0)$ yields,

$$q(\mathbf{e}=0) = \frac{\exp\Big[D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=1)] - D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=0)] + \log\frac{p(\mathbf{e}=0)}{p(\mathbf{e}=1)}\Big]}{1+\exp\Big[D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=1)] - D_{\mathrm{KL}}[q(\mathbf{s}||p(\mathbf{s}|\mathbf{e}=0)] + \log\frac{p(\mathbf{e}=0)}{p(\mathbf{e}=1)}\Big]}. \tag{24}$$

Equation 24 shows the formulation of distribution $q(\mathbf{e})$ in terms of the VaDE trick. We note that under the assumption that $p(\mathbf{e}=0) = p(\mathbf{e}=1) = 0.5$, the formulation is a softmax function of the difference between the KL divergences indicated in Eq. 4. As such, the proposed estimation of $q(\mathbf{e})$ may be seen as taking a sample from the most likely component in the VaDE-based distribution in Eq. 24, under the assumption of equal prior probabilities.