Learning Disentangled Behavior Embeddings

Changhao Shi*
University of California, San Diego
cshi@ucsd.edu

Sivan Schwartz*, Shahar Levy, Shay Achvat, Maisan Abboud, Amir Ghanayim
Technion - Israel Institute of Technology
{sivan.s,shahar86,shay.achvat,maisanabboud,amir.gh122}@campus.technion.ac.il

Jackie Schiller
Technion - Israel Institute of Technology
jackie@technion.ac.il

Gal Mishne
University of California, San Diego
gmishne@ucsd.edu

Abstract

To understand the relationship between behavior and neural activity, experiments in neuroscience often include an animal performing a repeated behavior such as a motor task. Recent progress in computer vision and deep learning has shown great potential in the automated analysis of behavior by leveraging large and high-quality video datasets. In this paper, we design Disentangled Behavior Embedding (DBE) to learn robust behavioral embeddings from unlabeled, multi-view, high-resolution behavioral videos across different animals and multiple sessions. We further combine DBE with a stochastic temporal model to propose Variational Disentangled Behavior Embedding (VDBE), an end-to-end approach that learns meaningful discrete behavior representations and generates interpretable behavioral videos. Our models learn consistent behavior representations by explicitly disentangling the dynamic behavioral factors (pose) from time-invariant, non-behavioral nuisance factors (context) in a deep autoencoder, and exploit the temporal structures of pose dynamics. Compared to competing approaches, DBE and VDBE enjoy superior performance on downstream tasks such as fine-grained behavioral motif generation and behavior decoding.

1 Introduction

Understanding the relationship between animal behavior and neural activity is a long-standing goal in neuroscience. To this end, recent advances in deep learning and computer vision has led to significant progress in the essential task of automatic analysis of high-resolution behavioral videos.

To extract behavioral information from rich video recordings, two avenues of research relying on deep learning have been proposed: landmark-based pose estimation methods [22, 26, 32] and autoencoder-based dimensionality reduction methods [2]. Pose estimation methods characterize animal behavior with the trajectories of body-part landmarks. Such methods train deep neural networks to predict manually labeled landmarks from raw video frames, then use the trained models to generate trajectories for new videos. While these methods have been highly successful in behavior analysis, they 1) require manual labeling which is often expensive; 2) heavily rely on the manual selection of the landmarks, which may differ across human annotators; and 3) struggle with subtle behavior that is hard to track, such as facial movements.

*Equal contributions

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

The complementary approach seeks to reduce the high-dimensional video data to low-dimensional latent factors through non-linear dimensionality reduction. These methods [2] rely on deep autoencoders to encode video frames by solving an unsupervised reconstruction task. The learned latent embeddings can then be used for downstream tasks, such as clustering, neural decoding, etc. While unsupervised learning does not rely on manual labeling, existing approaches currently suffer from certain drawbacks. First of all, canonical autoencoders can be easily biased by the varying visual nuisances across videos, such as lighting, distance to camera and physical attributes of animals, resulting in substantial distributional shifts among the learned behavior embeddings. While non-behavioral visual variability is usually negligible within single session videos, it makes downstream analysis difficult when applied to more than one session and to different animals. Secondly, end-to-end training on videos becomes challenging when considering temporal structures, as traditional temporal modeling methods are mostly designed for low-dimensional data. How to better exploit temporal structures of high-dimensional videos for action recognition, motion planning, etc., remains an open problem for computer vision research.

In this paper, we design Disentangled Behavior Embedding (DBE) to learn robust behavior embeddings from large, unlabeled, multi-session videos. Inspired by previous works [33], DBE mitigates the distributional gaps in multi-session videos by explicitly disentangling the dynamic behavioral factors (pose) from time-invariant, non-behavioral factors (context). A video frame is thus represented by a pair of disentangled pose and context components in contrast to a single entangled representation. For a given video, the context component is designed to be time-invariant to exclude any behavioral dynamics, whereas the pose component is bottlenecked to keep it from duplicating context information. DBE latents can be used in conjunction with temporal analysis models, e.g., Variational Animal Motion Embedding (VAME) [21], to perform downstream analysis. We also propose Variational Disentangled Behavior Embedding (VDBE), a fully end-to-end trainable model, which further exploits the temporal structures of the pose components using a stochastic dynamic model. Using variational inference, a Gaussian mixture prior is trained to capture the multi-modality of the transition between consecutive pose components and generate both continuous and discrete representations of the underlying animal behaviors in videos. Our methods are not only fully unsupervised, but also enjoy superior performances on downstream tasks such as behavioral state estimation and fine-grained behavioral motif generation. To summarize, the main contributions of this paper are:

1. We develop DBE to tackle the distributional shift across multi-session videos with adaptation to the behavioral neuroscience setting, e.g., extension to multi-view videos, proper design of context embedding with standard behavioral neuroscience paradigms.
2. On top of DBE, we design VDBE to learn both continuous and discrete latent representations for simultaneous embedding and segmentation within the same model. VDBE is end-to-end trainable in an unsupervised fashion, alleviating the need to train a second post-hoc model applied to latent embeddings.

2 Related Works

2.1 Video disentanglement

Disentanglement of the pose and context in a video is an area of increasing interest in computer vision. The explicit separation of the pose from the context not only facilitates the analysis of the temporal dynamics but also makes video manipulation easier. As a pioneer in this field, MCNet [30] is an end-to-end framework that models motion (pose) and content (context) independently. MCNet uses a convolutional LSTM to model temporal dynamics and shows good performance on pixel-level video prediction on real-world datasets. Sharing similar ideas, DRNet [5] also separates the pose from the content but models temporal dynamics directly on low-dimensional pose embeddings. It also utilizes an explicit adversarial loss for disentanglement. Although DRNet shows superior performance in video prediction, it is not end-to-end trainable. DDPAE [13] tackles videos of multiple targets on the moving MNIST dataset and performs disentanglement on each individual target. Yingzhen and Mandt [33] proposed a disentangled VAE and shows controllable video generation on the synthetic Sprite dataset. MoCoGAN [29] employs a generative adversarial network that decomposes the random noises into a content part and a motion part for video generation. S3VAE [34] learns disentangled context and pose representations of videos with self-supervised regularization. In our approach we

build a disentanglement mechanism in a similar paradigm as [33], by fixing context embeddings across time and constraining the dimension of pose embeddings.

2.2 Stochastic dynamic modeling

Latent state-space models (SSM) are widely used to model the temporal structures for sequential data [3, 8, 17, 20, 7]. While these models have achieved fruitful results in many areas such as natural language processing and speech recognition, adapting these models to high-dimensional, non-synthetic videos is still challenging to this date. Hafner et al. [11] proposed PlaNet, with a recurrent state-space model at its core, to learn environment dynamics from high-dimensional, synthetic videos and auxiliary action variables for planning. Franceschi et al. [9] proposed a latent dynamical model enhanced by residual connections for stochastic video prediction. Although there are other papers on stochastic video generation and prediction on low-dimensional frame embeddings [1, 6, 18], these methods tackle the problems in an auto-regressive fashion and thus offer no single latent state vector that captures all the dynamics, as opposite to state-space models. Our approach closely relates to [11, 9], but differs by using a multi-modal distribution for the transition model and the ability to generate both continuous and discrete representations.

2.3 Behavioral video analysis

Recently, supervised deep pose estimation methods have been widely used for behavior analysis for many animal models. Methods such as DeepLabCut (DLC) [22], DeepPoseKit [10] and LEAP [26] train deep neural networks to produce confidence maps for predefined landmark annotations, all in a similar fashion. Leveraging the success of DeepLabCut, Wu et al. [32] improved the quality of pose estimation by imposing additional spatial and temporal constraints on multiple landmarks during training. Luxem et al. [21] further developed a recurrent variational autoencoder (VAME) to encode the pose estimation from DLC to a higher-level behavioral space. Prior to deep learning approaches, Kabra et al. [15] provided a supervised pipeline to automatically annotate animal behavior based on user-defined labeling.

On the other hand, unsupervised dimensionality reduction methods like PCA have been used to analyze behavior for decades [28, 4, 31]. More recently, BehaveNet [2] presented meaningful results of behavior segmentation and neural decoding. BehaveNet first uses a deep convolutional autoencoder to compress behavioral videos, then fits an autoregressive hidden Markov (ARHMM) model to exploits the temporal structures of the learned frame embeddings. However, BehaveNet was evaluated only on videos in a single experimental session and cannot handle the distributional shift across multiple sessions. Our approach, which falls into this second category, not only inherits the advantage of being unsupervised but also handles the distributional gaps across multiple sessions by explicitly disentangling poses from contexts.

3 Methods

3.1 Problem formulation

Formally, given a behavioral video $x_{1:T}$ of length T, we are interested in disentangling the dynamic behavioral factors from the time-invariant non-behavioral nuisances in the video in an unsupervised fashion. Our goal is to learn meaningful embedding $z^p_{1:T}$ that capture common behavioral structures across multiple animals in multiple recording sessions. We assume that the animal is performing a repetitive behavioral protocol, such as a mouse performing a forepaw reach or lever press [19, 24], and not freely moving. Additionally, in behavior experiments, videos are often recorded from different angles, resulting in multi-view data, e.g. $(x_a, x_b)_{1:T}$ where a and b denote the two views. For multi-view videos, we treat each view as a separate image and fuse the outputs of encoders of each view as the representation of this timestamp, whereas for mono-view videos the embedding of the frame is the natural representation. For the rest of this paper, we will slightly abuse the notations and omit the view subscripts, e.g., $x_{1:T}$ refer to either mono-view frames or multi-view frames $(x_a, x_b)_{1:T}$, and $z_{1:T}$ refer to either the embeddings of mono-view frames or the fused embeddings of multi-view frames.

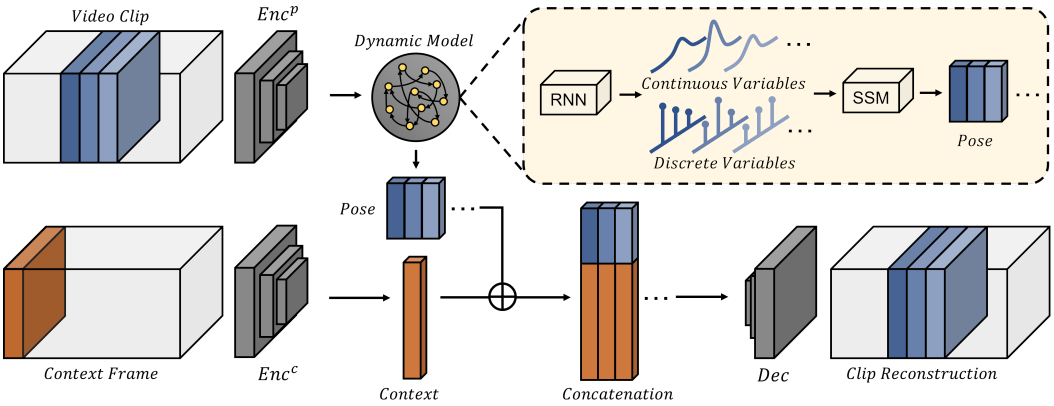


Figure 1: Overview of VDBE on a mono-view video. The pose encoder encodes individual frames into a sequence of frame embeddings, which are then passed through a stochastic dynamic model to generate pose embeddings. The context embeddings are generated from a single conditional context frame by the context encoder. Both embeddings are concatenated at each timestamp as inputs to the decoder which reconstructs the original video clip. The dynamic model (top right) uses an RNN to infer both latent continuous embeddings and discrete states that capture the transition of the frame embeddings, and uses a linear state-space model for generation. When the dynamic model is not used, the full VDBE model becomes DBE. The input video can be either single view or multi-view. For multi-view videos, encoders and decoders are used for each view separately, and the pose and context components are concatenations of the corresponding encoder outputs from each view.

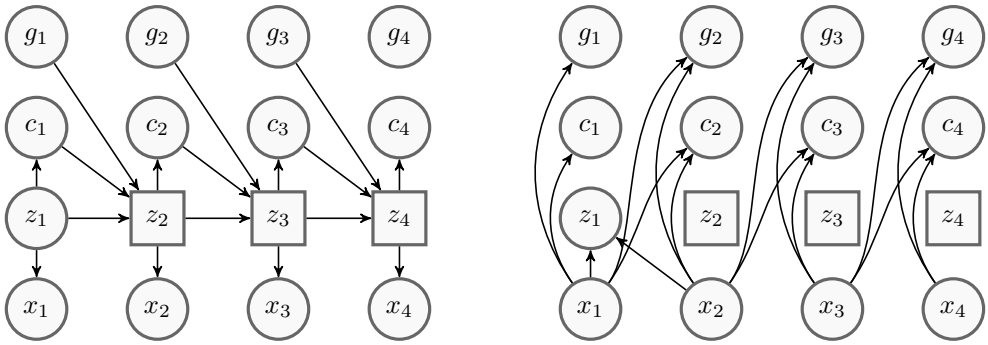
3.2 Disentanglement

In order to factor out the non-behavioral factors, our approach utilizes two separate lines of encoders, a context encoder E^c and a pose encoder E^p, for each view of the videos. Assuming the non-behavioral factors are time-invariant within each video, the convolutional context encoder E^c infers a video-level context embedding z^c from a single frame of the video. For multi-view videos, the context of each view is further concatenated into a single context embedding z^c, then used for the reconstruction of all the frames in this video. The pose encoder E^p shares the same architecture with E^c, and encodes the behavioral dynamics of the animal from all video frames $x_{1:T}$ into the pose embeddings $z^p_{1:T}$. For multi-view videos, the outputs of pose encoders are concatenated to get $z^p_{1:T}$. The temporal structures of the pose component are exploited by a stochastic dynamic model stacked on top of the pose encoders, which we will detail in the following. To reconstruct the input video $x_{1:T}$, the context embedding z^c is duplicated and concatenated with each pose embedding $z^p_{1:T}$, then for each view these concatenated embeddings are further decoded into frame reconstructions by the corresponding convolutional decoders D.

Because the context embedding z^c is time-invariant and therefore incapable of capturing dynamics, we ensure that it contains no behavioral information about the animal. Thus, the key question about the disentanglement is how to ensure that the pose embeddings contain no context information. To enforce disentanglement, we create a bottleneck on the pose embeddings inspired by [33]. We choose the dimension of the pose embeddings to be as small as possible in the premise of good reconstruction. This ensures that the capacity of pose embeddings is only sufficient for capturing the temporal dynamics without any redundant context information. As we will show in the Sec 4, the disentangled pose embeddings of different sessions are well-aligned with each other, in sharp contrast against the results from canonical autoencoders where the embeddings of different sessions form distinct clusters. The disentangled context embedding, on the other hand, capture these non-behavioral nuisances and form into clusters by sessions as expected. The full disentanglement mechanism is shown in Figure 1.

3.3 Dynamic model

We model the temporal dynamics of the pose embeddings $z^p_{1:T}$ using a sequential variational autoencoder. For the sake of brevity, we will use z to refer to z^p in this section. Similar abbreviations are applied for other variables as well. We assume that the pose components govern the latent state of the behaviors and we want to model the temporal transition of the states. In a typical stochastic



(a) Generation (b) Inference

Figure 2: The generation (a) and inference (b) process of our stochastic dynamic model. Round nodes represent stochastic variables and squared nodes represent deterministic variables. Note that in this illustration, z_1 is inferred from two conditional frames x_1 and x_2; and for c_t and g_t only the paths from their immediate past frames are shown for the sake of clarity.

state-space model, the transition from the previous state z_{t-1} to the current state z_t is modeled stochastically. However, a purely stochastic transition makes it hard for the model to keep long-term information [11]. Thus, we factor out the stochasticity of the dynamics into a separate component s. This means that at a given timestamp, the stochastic latent variable s_t is first sampled from a prior distribution, usually a Gaussian distribution, then the current state z_t is generated from the previous state z_{t-1} and the stochasticity s deterministically.

While this strategy has been proved to be effective in many scenarios [6, 11, 9], the uni-modal Gaussian transition distribution is limited for behavior analysis because it does not capture the multi-modality of behavior. For example, in the video dataset we will show in Section 4, the transition of the animal poses is usually caused by a set of distinct, prototypical actions, such as lift, reach and grab. A uni-modal Gaussian transition distribution is inadequate for modeling such multi-modal transitions. Furthermore, for the purpose of motif segmentation it offers no discrete states which are highly desired.

A more suitable choice for capturing such multi-modality is a Gaussian mixture model [27]. Shu et al. [27] design the transition distribution in the generative model to be a Gaussian mixture while keeping the posterior inference approximation to be a uni-modal Gaussian. A closed-form approximation of the exact KL divergence is then used in evidence lower bound for optimization. In this paper, we use an alternative approach to introduce the Gaussian mixture. Instead of using a single stochastic latent variable s with a Gaussian mixture prior, we decompose it into two stochastic variables, the mixture index c and a Gaussian noise g. This Gaussian mixture prior also enables the model to perform embedding (z_t) and segmentation (c_t) simultaneously. With all these, we can dive into the details of our variational autoencoder, as illustrated in the zoomed-in region in Figure 1.

Generation The generative procedure is shown in Figure 2a. We model the transition between the previous state z_{t-1} and the current state z_t using a Gaussian mixture. The generative model is:

$$z_1 \sim \mathcal{N}(0, I),\ z_{t+1} = f_\theta(z_t, s_t), \tag{1}$$

$$c_t \sim \text{cat}(\pi_\theta(z_t)),\ g_t \sim \mathcal{N}(0, I),\ s_t = \mu_{c_t} + \sigma_{c_t} \odot g_t, \tag{2}$$

$$x_t = \mathcal{N}(\text{Dec}_\theta(z_t, z_c)), \tag{3}$$

where π_θ and f_θ are linear projection layers. At each timestamp, stochastic variables c_t and g_t model the discrete state and the additional stochasticity, then the continuous state z_t is generated with a deterministic path from z_{t-1}, c_t and g_t. Note that although both c_t and g_t capture the stochasticity of the transition, only c_t is dependent on the previous state z_{t-1}. c_t is sampled from a categorical distribution parameterized by z_{t-1} and g_t is simply sampled from a standard Gaussian distribution. The inferred c_t is used for for motif segmentation in the test stage. We assume the covariance matrix of each Gaussian component is diagonal and $\odot$ denotes element-wise multiplication (Hadamard product).

Inference Because of the non-linearity in the generative model, the true posterior distributions of the latent variables are intractable. As shown in Figure 2b, we use variational inference to approximate these posteriors. The inferred posterior is factorized as $q_\phi(c_{1:T}, g_{1:T}, z_1) = \prod_{t=1}^{T} q(c_t|x_{1:T})q(g_t|x_{1:T})q(z_1|x_{1:C})$, where $q(c_t|x_{1:T})$ is a conditional categorical distribution, and $q(g_t|x_{1:T})$ as well as $q(z_1|x_{1:C})$ is a conditional Gaussian distribution. We use a shared GRU network followed by separate projection heads for $q(c_t|x_{1:T})$ and $q(g_t|x_{1:T})$. For $q(z_1|x_{1:C})$, we use an MLP whose inputs are the first C conditional frames in the sequence. At test-time, we also use these inference networks to generate behavior representation, i.e. we use the mean of the Gaussian distribution and the highest category of the categorical distribution as the continuous and discrete representations of the behavior of each frame.

Optimization We use the following evidence lower bound (ELBO) for optimization:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x_{1:T}) = \sum_{t=1}^{T} \Big[&\mathbb{E}_{q_\phi(c_{1:T}, g_{1:T}, z_1)} \log p_\theta(x_t|z_t) - \alpha \mathcal{D}_{KL}(q_\phi(c_t|x_{1:t})||p_\theta(c_t|z_{t-1})) \\ &- \beta \mathcal{D}_{KL}(q_\phi(g_t|x_{1:t})||p_\theta(g_t)) \Big] - \gamma \mathcal{D}_{KL}(q_\phi(z_1|x_{1:C})||p_\theta(z_1)),\end{aligned} \tag{4}$$

where α, β and γ are the trade-off parameters that control the information flow to each stochastic latent variables $c_{1:T}$, $g_{1:T}$ and z_1 respectively [12]. The first log-likelihood term corresponds to the frame reconstruction and the other KL-divergence terms are the regularization of the inferred stochastic latent variables to their prior knowledge.

The above expectation is estimated with a single sample from the approximated posterior distribution. To allow gradients to back-propagate through the non-differentiable sampling operation, we use the re-parameterization tricks. For the Gaussian posterior distributions $q(g_t|x_{1:T})$ and $q(z_1|x_{1:C})$, we re-parameterize the distribution into their estimated mean and standard deviation with a standard Gaussian noise [16]. For the categorical posterior distribution $q(c_t|x_{1:T})$ we use the Gumbel-Softmax re-parameterization trick [14]: the sampling process is re-parameterized into the estimated discrete log-probability and a Gumbel noise, following by a softmax normalization. The outputs of softmax are also discretized to one-hot encodings to ensure consistency between training and testing. Note that theoretically one can compute the exact expectation over the discrete latent variable c_t without using Monte Carlo estimation, but our experiments show that one-time sampling can generate satisfying results with the advantage of being more efficient.

4 Results

We evaluate our methods on two multi-session video datasets: the Hand-reach dataset [19] we collected, and the Wide-Field Calcium Imaging (WFCI) dataset [24, 25]. The Hand-reach dataset contains videos of head-fixed mice who were trained to reach from their perch to a food pellet. The datasets consists of 263 successful trials of 3 expert animals (who have learned the motor task) in 8 sessions, and includes fine-grained behavior. In a successful execution of the task, the animal will lift its paw from the perch, reach and grab the pellet and bring it to its mouth and eat it. The animal might perform multiple grab attempts in a single trial before successfully grabbing the pellet. An illustration of the Hand-reach dataset is shown in Figure 3. The videos were acquired on multiple dates, within which the experimental layout is potentially different, e.g., location of the table. Two cameras simultaneously record from the front and right side of the mouse. Each video is 12 seconds long, with a frame rate of 200Hz. The video begins with a 4-seconds inter-trial interval (ITI), the end of which we set as the origin (0) of our time axis. A cue of 0.2 seconds is shortly followed by table-turning which brings the food in front of

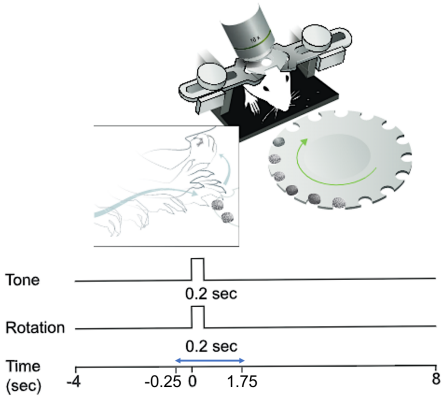


Figure 3: Hand-reach dataset. Mice were head-fixed and trained to grab a food pellet. Trial structure: 4 seconds ITI period followed by the tone and table turning. We analyze the [-0.25, 1.75] segment which contains most of the significant behaviors for expert mice.

the mouse. Since the mice in this dataset are experts, and thus mostly stationary before the cue and chewing the pellet after successful execution of the hand-reach, we use a 2-second clip of the main dynamic behavior of each full video to train the model. Each clip starts at 0.25 second before the cue and ends at 1.75 seconds after, which contains most of the motions of interest. The context frame for each video is selected from the ITI period, sampled at 1 second before the cue, when most of the subjects are stationary.

In the WFCI dataset (an extended version of the dataset in [2]), head-fixed mice perform a visual decision making task, using levers to initiate trials and reporting choices by licking from spouts. The behavioral videos are recorded from a side and a bottom view of the subjects, with a length of 189 frames and a frame rate of 30Hz. The dataset contains 3708 videos from 8 sessions of 4 animals in total. We randomly select 80% of the videos as training set and use the remaining 20% as test set.

We train both DBE and VDBE, without and with the stochastic dynamic modeling component. We compare our methods against two behavior embedding methods: BehaveNet [2] and VAME [21], with implementation details of each provided in Appendix B and Appendix C, respectively. In Section 4.1 and 4.2, we demonstrate the efficacy of our disentanglement component. In Section 4.3, we show both DBE (combined with a motif segmentation model) and VDBE achieve superior results for motif segmentation compared to the original models.

4.1 Disentanglement

To demonstrate the efficacy of disentanglement, we focus on session recognition: given the behavior embedding of a specific video frame, can we identify which session (animal and date) this frame comes from? We train a simple linear classifier to classify the experimental session from different pose embeddings, and ideal disentanglement should generate session-agnostic pose embeddings. The accuracy of the session classification for each model is reported in Table 1. DBE as well as our full VDBE model generates session-agnostic pose embeddings, compared to the plain CAE autoencoder used in Behavenet. Although DLC also provides session-agnostic coordinates, selected markers do not encode enough useful information for extracting robust behavioral motifs across sessions (Section 4.3). The context embedding of our methods (DBE-context and VDBE-context) accurately classify the session, further demonstrating disentanglement of context from pose.

We further visualize the advantage of disentanglement in Figure 4. We plot the 2D PCA applied to the BehaveNet and DBE embeddings of two videos that show similar behaviors but are recorded in different sessions.The plain CAE of BehaveNet generates separate pose embeddings even though the behaviors are similar. This distributional shift will be problematic when one wants to transfer the knowledge learned in one session to another. This means that one needs to learn a new model every time when there are new sessions, and learn a mapping across animals to relate the embeddings of the same behavior. In contrast to BehaveNet, our DBE nicely aligns the two videos in the pose embedding space. As we will show in the later section, this consistency is crucial to motif segmentation and behavior decoding. More results comparing BehaveNet and DBE embeddings, as well as ethograms of both the Hand-reach and the WFCI dataset, are provided in Appendix F and G.

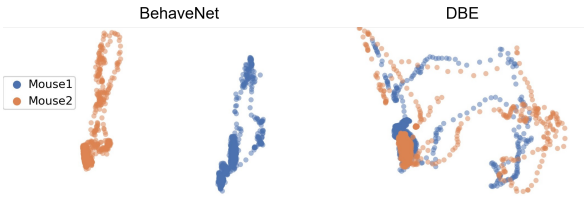


Figure 4: Embeddings of 2 similar videos (Hand-reach)

Model	Hand-reach	WFCI
BehaveNet	100.00	99.99
DLC	21.19	-
DBE	28.21	35.49
VDBE	17.94	19.54
DBE-context	88.59	100.00
VDBE-context	96.19	100.00

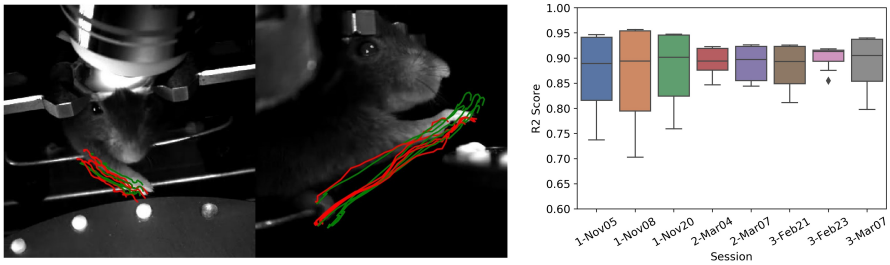
Table 1: Session classification

4.2 Trajectory decoding

DLC is a robust method used to track manually selected landmarks in behavioral videos. A good behavioral embedding should contain the information encoded by DLC. To illustrate that our approach learns meaningful behavior embeddings, we perform a simple linear regression task from the DBE

embeddings to the DLC coordinates in the Hand-reach dataset. A good fit of the linear regression model suggests that our learned embedding contains the DLC coordinate information.

We place 5 markers on the center and 4 digits of the right paw of the mouse in each view of the videos, and train a DLC model to track the coordinates of those 10 markers. Due to the differences in spatial layout of the sessions (e.g., table location), we fit separate linear regression models to each marker for each individual experimental session. Only DLC coordinates that have a confidence score larger than 0.95 are used for the training of linear regression.



(a) DLC trajectories prediction (b) Goodness-of-fit of linear regression

Figure 5: Predicting DLC trajectories from disentangled embeddings on the Hand-reach dataset. (a) Example video overlaid with true DLC trajectories (red) and the linear regression model predictions (green). (b) R^2 averaged across each coordinate for each individual session.

An example of DLC predictions and the average R^2 of each individual linear regression model are shown in Figure 5. In Figure 5a, the predictions are well aligned with the true DLC trajectories, capturing the right paw of the mouse moving from the perch to the pellet on the table. The high R^2 scores in Figure 5b show that the linear regression models are well-fitted to the data, which indicates that the DBE embeddings manifest meaningful information about the selected markers.

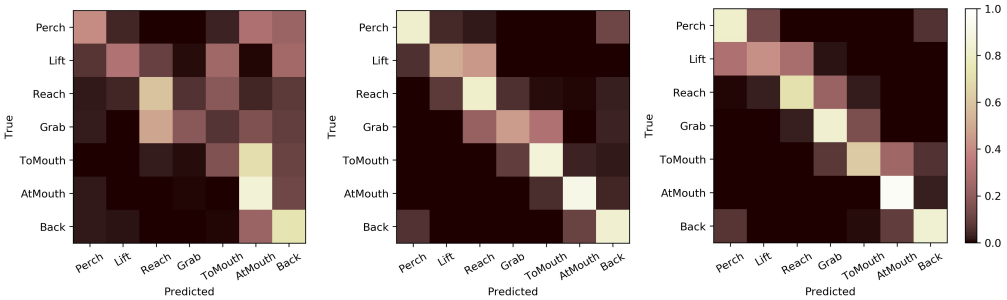
4.3 Motif segmentation

Learning fine-grained behavior motifs from videos is an essential but challenging task for analyzing animal behavior, especially when no supervision is available. In this section, we compare our approach with other competing methods for motif segmentation on the Hand-reach dataset. To evaluate the quality of the generated motifs, we select 10% of the videos (25 videos) and perform expert manual labeling of the selected videos with a set of seven behaviors of interest: On-Perch, Lift, Reach, Grab, To-mouth, At-mouth, and Back-To-Perch (4730 labeled frames). The rest of the videos are used as the training set for motif segmentation methods, i.e. clustering. For each method, we assign the learned motifs to the labeled set of behavior, where the mapping is selected to maximize the clustering accuracy of holdout videos. Thus, each motif corresponds to a "mega-cluster" that may or may not contain more than one inferred discrete state. Note that we set the number of the discrete states k to be larger than the number of target labels for two reasons. The first reason is that only part of the frames are labeled and the target labels do not cover all the possible behavior of the mice. Extra clusters are necessary to capture these additional behavior. The second reason is that setting a larger number of motifs rather than the exact number of labels accounts for inter-class versus intra-class variability. For example, this enables separating a highly variable behavioral label like Reach into more than one motif without having to compensate for that by merging At-mouth and To-mouth into a single motif. We evaluate the performance of motif segmentation on three clustering metrics, accuracy, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI).

We use the DBE embeddings in two variations for motif segmentation: DBE+ARHMM which replaces the CAE embeddings of the BehaveNet model with DBE embeddings, and DBE+VAME, which replaces the DLC trajectories with DBE embeddings as inputs to VAME. As shown in Table 2 and Figure 6, disentanglement is the key to achieve good behavioral clusters across different animals and multiple sessions. Combining DBE for frame embedding and 30-cluster VAME (DBE+VAME) for temporal dynamics and segmentation achieves the best accuracy, NMI and ARI. Our full VDBE model has very similar performance, with both models exhibiting $\sim 30\%$ higher accuracy than BehaveNet and over 20% higher accuracy than the original VAME framework (relying on DLC inputs). These results also show that using a more powerful temporal model such as VAME can

Table 2: Multi-session motif segmentation for the Hand-reach dataset

Model	Unsup.	Dis.	End2End	Accuracy	NMI	ARI
BehaveNet (k=15)	✓	✗	✗	50.52	30.89	20.98
BehaveNet (k=30)	✓	✗	✗	53.09	33.44	25.09
DBE + ARHMM (k=15)	✓	✓	✗	64.04	42.06	36.08
DBE + ARHMM (k=30)	✓	✓	✗	71.49	48.96	45.76
DLC + VAME (k=30)	✗	✗	✗	57.17	42.07	44.09
DBE + VAME (k=15)	✓	✓	✗	71.31	59.47	51.46
DBE + VAME (k=30)	✓	✓	✗	81.25	67.31	67.33
VDBE (k=30)	✓	✓	✓	79.53	64.85	66.94



(a) BehaveNet (k=30) (b) DBE+VAME (k=30) (c) VDBE (k=30)

Figure 6: Confusion matrices of BehaveNet, DBE+VAME and VDBE for the Hand-reach dataset.

benefit motif segmentation. With disentanglement, the confusion matrices (Fig. 6) are more diagonal which indicates better clustering. Figure 7 shows examples of random segments from VDBE motifs corresponding to 4 different labels. Note that VDBE successfully clusters the behaviors even when the context is varying. Additional results can be found in supplementary materials.

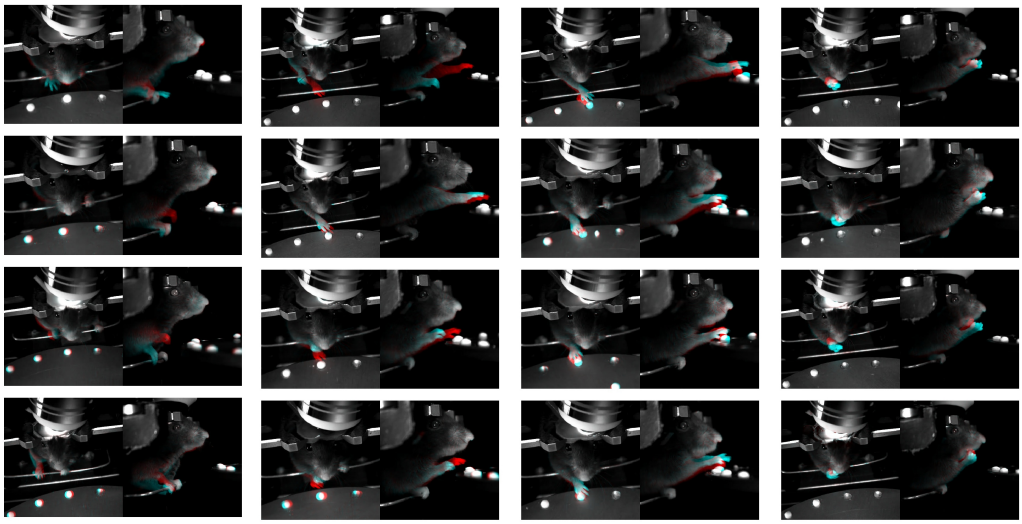
4.4 Controllable behavior generation

One advantage of VDBE and ARHMM-based models compared to VAME is that we can decode synthetic videos from behavior embeddings, which demonstrates better interpretability and allows potential applications to neural decoding. In this section, we further investigate behavior decoding by manipulating the context components. In order to generate synthetic videos that replicate source behavior with a new context, we use the inferred discrete behavior representations $c_{1:T}$ from the source video and the context embedding from the target session, while $g_{1:T}$ and z_1 are sampled from their prior distributions. If the target session is the same as the source session, the synthetic video should be similar to the source video. The results are shown in Figure 8. Note that the behavior in the source video is replicated in both generated videos with the same and different contexts. This indicates that our model successfully learns the dynamics of the behavior.

5 Conclusions

In this paper, we tackle the problem of unsupervised behavior representation learning from multi-session videos. A disentanglement mechanism is designed to separate session-wise non-behavioral factors from the useful behavioral information, and a stochastic dynamic model is jointly trained to model the temporal dynamics of behaviors. Compared with competing methods, our method shows superior performance on multiple tasks such as fine-grained motif segmentation, meanwhile also enjoys the advantages of being unsupervised, end-to-end trainable, and more interpretable.

Despite the above improvements, a couple of open questions still remain and will be explored in future work. Firstly, the current model which relies on a predefined number of states is limited in generalization of the motif assignment to videos which include novel unseen behavior. Secondly, our disentanglement assumes the non-behavioral nuisances are stationary through a video, which is not always true. Future work should benefit from allowing slow dynamics in context embeddings. Thirdly,



(a) Lift (b) Reach (c) Grab (d) At-Mouth

Figure 7: Motifs learned by VDBE. The blue channel indicates the start of the motif and the red channel indicates the end. Note that in each column the behavior is consistent but the context is varying (e.g., location of the table and identity of the mouse).

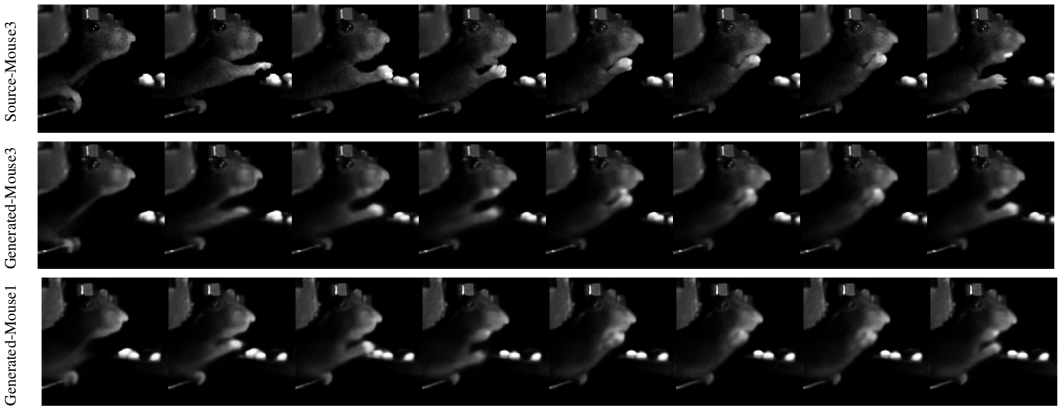


Figure 8: Source (top row) and generated videos of the same discrete behavior sequence, with the same (middle row) and different (bottom row) contexts. Each row is plotted at intervals of 20 frames.

with the distributional shift in videos confirmed, it would be interesting to study disentangling other visual behavior representations such as DLC for multi-session analysis. Finally, with the learned behavioral embeddings consistent across sessions, future works can leverage this representation to relate behavior to neural activity [23].

Regarding societal impacts of our study as well as behavior analysis research in general, although there is a significant gap between the head-fixed animal behavior studied here and naturalistic human behavior, we should always be careful about the negative consequences for human monitoring.

Code is available at https://github.com/Mishne-Lab/Disentangled-Behavior-Embedding.

Acknowledgement

This work was supported by in part by the National Institutes of Health grant No. R01EB026936. We thank Simon Musall and Matthew Whiteway for providing the WFCI dataset and their generous help.

References

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

[2] E. Batty, M. Whiteway, S. Saxena, D. Biderman, T. Abe, S. Musall, W. Gillis, J. Markowitz, A. Churchland, J. P. Cunningham, et al. Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. *Advances in Neural Information Processing Systems*, 32: 15706–15717, 2019.

[3] J. Bayer and C. Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[4] G. J. Berman, D. M. Choi, W. Bialek, and J. W. Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014.

[5] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4417–4426, 2017.

[6] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR, 2018.

[7] Z. Dong, B. Seybold, K. Murphy, and H. Bui. Collapsed amortized variational inference for switching nonlinear dynamical systems. In *International Conference on Machine Learning*, pages 2638–2647. PMLR, 2020.

[8] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *arXiv preprint arXiv:1710.05741*, 2017.

[9] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari. Stochastic latent residual video prediction. In *International Conference on Machine Learning*, pages 3233–3246. PMLR, 2020.

[10] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe, and I. D. Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8: e47994, 2019.

[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019.

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

[13] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 515–524, 2018.

[14] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[15] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10(1):64, 2013.

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

[17] R. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

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

[19] S. Levy, M. Lavzin, H. Benisty, A. Ghanayim, U. Dubin, S. Achvat, Z. Brosh, F. Aeed, B. D. Mensh, Y. Schiller, et al. Cell-type-specific outcome representation in the primary motor cortex. *Neuron*, 107(5):954–971, 2020.

[20] S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922. PMLR, 2017.

[21] K. Luxem, F. Fuhrmann, J. Kürsch, S. Remy, and P. Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *bioRxiv*, 2020.

[22] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[23] G. Mishne, R. Talmon, R. Meir, J. Schiller, M. Lavzin, U. Dubin, and R. R. Coifman. Hierarchical coupled-geometry analysis for neuronal structure and activity pattern discovery. *IEEE Journal of Selected Topics in Signal Processing*, 10(7):1238–1253, 2016.

[24] S. Musall, M. T. Kaufman, S. Gluf, and A. K. Churchland. Movement-related activity dominates cortex during sensory-guided decision making. *BioRxiv*, page 308288, 2018.

[25] S. Musall, M. T. Kaufman, A. L. Juavinett, S. Gluf, and A. K. Churchland. Single-trial neural dynamics are dominated by richly varied movements. *Nature neuroscience*, 22(10):1677–1686, 2019.

[26] T. D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz. Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1): 117–125, 2019.

[27] R. Shu, J. Brofos, F. Zhang, H. H. Bui, M. Ghavamzadeh, and M. Kochenderfer. Stochastic video prediction with conditional density estimation. In *ECCV Workshop on Action and Anticipation for Visual Learning*, volume 2, page 2, 2016.

[28] G. J. Stephens, B. Johnson-Kerner, W. Bialek, and W. S. Ryu. Dimensionality and dynamics in the behavior of c. elegans. *PLoS Comput Biol*, 4(4):e1000028, 2008.

[29] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[30] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017*. International Conference on Learning Representations, ICLR, 2017.

[31] A. B. Wiltschko, M. J. Johnson, G. Iurilli, R. E. Peterson, J. M. Katon, S. L. Pashkovski, V. E. Abraira, R. P. Adams, and S. R. Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.

[32] A. Wu, E. K. Buchanan, M. Whiteway, M. Schartner, G. Meijer, J.-P. Noel, E. Rodriguez, C. Everett, A. Norovich, E. Schaffer, et al. Deep graph pose: a semi-supervised deep graphical model for improved animal pose tracking. *bioRxiv*, 2020.

[33] L. Yingzhen and S. Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pages 5670–5679. PMLR, 2018.

[34] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

A ELBO derivation

The ELBO (Eq. 4) is derived as follow:

$$
\begin{aligned}
\log p(x_{1:T}) &= \log \iiint_{c,g,z} p(x_{1:T}, c_{1:T}, g_{1:T}, z_1)\, dc\, dg\, dz \\
&= \log \iiint_{c,g,z} q(c_{1:T}, g_{1:T}, z_1|x_{1:T}) \frac{p(x_{1:T}, c_{1:T}, g_{1:T}, z_1)}{q(c_{1:T}, g_{1:T}, z_1)|x_{1:T}}\, dc\, dg\, dz \\
&= \log \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1)|x_{1:T}} \frac{p(x_{1:T}, c_{1:T}, g_{1:T}, z_1)}{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \\
&= \log \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \frac{p(x_{1:T}|c_{1:T}, g_{1:T}, z_1)p(c_{1:T}, g_{1:T}, z_1)}{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \\
&\geq \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \log \frac{p(x_{1:T}|c_{1:T}, g_{1:T}, z_1)p(c_{1:T}, g_{1:T}, z_1)}{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \\
&= \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \log \frac{p(x_{1:T}|c_{1:T}, g_{1:T}, z_1)p(c_{1:T}, g_{1:T}, z_1)}{q(c_{1:T}|x_{1:T})q(g_{1:T}|x_{1:T})q(z_1|x_{1:T})} \\
&= \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \Bigg[\log p(x_{1:T}|c_{1:T}, g_{1:T}, z_1) + \log \frac{p(c_{1:T}|g_{1:T})}{q(c_{1:T}|x_{1:T})} \\
&\quad + \log \frac{p(g_{1:T})}{q(g_{1:T}|x_{1:T})} + \log \frac{p(z_1)}{q(z_1|x_{1:C})} \Bigg] \\
&= \mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1|x_{1:T})} \Bigg[\log \sum_{t=1}^{T} p(x_t|z_t) + \log \sum_{t=1}^{T} \frac{p(c_t|z_t)}{q(c_t|x_{1:T})} + \log \sum_{t=1}^{T} \frac{p(g_t)}{q(g_t|x_{1:T})} \\
&\quad + \log \frac{p(z_1)}{q(z_1|x_{1:C})} \Bigg] \\
&= \sum_{t=1}^{T} \Bigg[\mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1)} \log p(x_t|z_t) - \mathcal{D}_{KL}(q(c_t|x_{1:T})||p(c_t|z_{t-1})) \\
&\quad - \mathcal{D}_{KL}(q(g_t|x_{1:T})||p(g_t)) \Bigg] - \mathcal{D}_{KL}(q(z_1|x_{1:C})||p(z_1)).
\end{aligned}
$$

The posterior is factorized using mean-field approximation. We get the final objective after applying trade-off parameters to each term:

$$
\begin{aligned}
&\sum_{t=1}^{T} \Big[\mathbb{E}_{q(c_{1:T}, g_{1:T}, z_1)} \log p(x_t|z_t) - \alpha \mathcal{D}_{KL}(q(c_t|x_{1:T})||p(c_t|z_{t-1})) - \beta \mathcal{D}_{KL}(q(g_t|x_{1:T})||p(g_t)) \Big] \\
&- \gamma \mathcal{D}_{KL}(q(z_1|x_{1:C})||p(z_1)).
\end{aligned}
$$

B Architectures

We use an architecture similar to the convolutional autoencoder used in BehaveNet. The kernel size of all convolutional layers is 5×5. 'Cat' denotes concatenation operation which combines the embeddings of different views. We use a pose dimension of 6 and 4 for the Hand-reach dataset and the WFCI dataset respectively.

For the stochastic dynamic model, we use a single layer bidirectional RNN with a hidden dimension of 128 for inference. For the Hand-reach dataset, the dimension of g_t and z_t is set to 4 and 8 respectively. For the WFCI dataset, we use 3 and 4 for each variable. Note that for the Hand-reach dataset we slightly increase the dimension of pose embedding from 6 in DBE to 8 in full VDBE, in order to allow more capacity for the temporal history. The bottleneck of VDBE is placed before the dynamical model as we set the output dimension of the pose encoder to be 6.

Table 3: Pose encoder

Layer type	Stride	Output shape
Conv2D	(1, 1)	(128, 128, 32)
Maxpool2D	(2, 2)	(64, 64, 32)
Conv2D	(1, 1)	(64, 64, 64)
Maxpool2D	(2, 2)	(32, 32, 64)
Conv2D	(1, 1)	(32, 32, 128)
Maxpool2D	(2, 2)	(16, 16, 128)
Conv2D	(1, 1)	(16, 16, 128)
Maxpool2D	(2, 2)	(8, 8, 256)
Cat	N/A	(8, 8, 512)
Conv2D	(8, 8)	(1, 1, pose dim)

Table 4: Context encoder

Layer type	Stride	Output shape
Conv2D	(2, 2)	(64, 64, 32)
Conv2D	(2, 2)	(32, 32, 64)
Conv2D	(2, 2)	(16, 16, 128)
Conv2D	(2, 2)	(8, 8, 256)
Cat	N/A	(8, 8, 512)

Table 5: Decoder

Layer type	Stride	Output shape
Conv2D	(1, 1)	(8, 8, 256)
Upsample2D	(2, 2)	(16, 16, 256)
Conv2D	(1, 1)	(16, 16, 128)
Upsample2D	(2, 2)	(32, 32, 128)
Conv2D	(1, 1)	(32, 32, 64)
Upsample2D	(2, 2)	(64, 64, 64)
Conv2D	(1, 1)	(64, 64, 1)
Upsample2D	(2, 2)	(128, 128, 1)

C Training details

For the Hand-reach dataset, we train our model for 2000 epochs with a learning rate of 0.002 using the Adam optimizer. At each iteration, we extract a batch of 20 random short clips of 50 consecutive frames from the full video clips and train our model on these cropped video clips. The trade-off parameters α, β and γ are set to 1e-5, 5e-5 and 5e-5 respectively. To avoid KL-divergence vanishing, we apply an annealing strategy to the KL-divergence terms, which linearly increases the trade-off parameters from 0 to their full values at the 200th epoch.

To train an ARHMM model, we use the SSM package. The latent dimension of the ARHMM model is set to 4, and the models are trained for 100 epochs using EM algorithms. Our experiments show that increasing the latent dimension has no significant effects on the performance of ARHMM. To train VAME on the Hand-reach dataset, we use the implementation released by the authors. We set the window size to 10, prediction steps to 5, and the hidden dimension of the VAME embeddings to 20.

D Mono-view learning

While we demonstrate our results on multi-view videos, our methods are also compatible with mono-view videos as mentioned in Section 3.1. In this section we evaluate our methods using only the side view of the Hand-reach dataset. We also compare our methods with Yingzhen and Mandt [33] which first proposed the dimension bottleneck technique for disentanglement, however it was not designed for multi-view data. For the [33] model, we use the same dimension of 6 for the pose embeddings, while keeping the other dimensions the same as its default values. The model is trained on 20 frames-length videos for 500 epochs with a batch size of 10. The video crops are selected in the same fashion as other experiments. The results in Table 6 show that our methods consistently outperforms BehaveNet with either mono-view videos or multi-view videos. On mono-view videos our methods also outperforms [33] which indicates the importance of proper adaptations of the disentanglement to the behavioral behavioral neuroscience settings.

Table 6: Mono-View ablations

Model	Accuracy	NMI	ARI
BehaveNet (k=15)	52.39	37.65	29.99
BehaveNet (k=30)	58.32	44.03	35.40
DBE + ARHMM (k=15)	51.90	31.17	23.46
DBE + ARHMM (k=30)	71.01	50.69	43.32
DBE + VAME (k=15)	69.47	59.90	52.36
DBE + VAME (k=30)	73.93	64.16	54.27
VDBE (k=30)	72.84	56.86	51.06
Yingzhen & Mandt + ARHMM (k=15)	54.26	33.99	22.85
Yingzhen & Mandt + ARHMM (k=30)	65.95	44.68	36.29

Table 7: Fusion ablations

Model	Accuracy	NMI	ARI
Concatenating (k=15)	64.04	42.06	36.08
Concatenating (k=30)	71.49	48.96	45.76
Averaging (k=15)	59.59	38.56	29.44
Averaging (k=30)	62.42	42.88	35.11
Multi-channel (k=15)	55.10	34.17	25.39
Multi-channel (k=30)	68.11	46.75	41.29

E View fusion

In this section, we perform ablation studies on different fusion approaches for multiple views. We compare three different view fusions on the DBE model with ARHMM: concatenating, averaging, and multi-channel fusion. Concatenation (used in DBE and VDBE), the outputs of the separate encoders of multiple views are concatenated, while for averaging the outputs are averaged together. Multi-channel fusion refers to the fusion used by BehaveNet which treats different views at a timestamp as different channels of a single input image to the convolutional network. We use concatenation in the main paper. As shown in Table 7, the concatenating leads to the best results on all 3 metrics, demonstrating the importance of proper design of multi-view fusion.

F Disentangled embeddings

We use PCA decomposition to visualize the learned embeddings from BehaveNet and DBE. In Figure 9 we visualize embedded videos from the Hand-reach dataset. BehaveNet embeds videos from different sessions into different clusters, while our disentanglement mechanism helps DBE to learn consistent behavior representations across all sessions. Note that for BehaveNet, embeddings from the same animals are closer to each other, as the contextual differences are smaller than those between different animals. In Figure 10 we visualize embedded videos from the WFCI dataset, with similar results.

G Ethograms

To further demonstrate the efficacy of disentanglement, we compare the ethograms of ARHMM estimation on both BehaveNet and DBE in comparison to VDBE. The comparison on the Hand-reach dataset is shown in Figure 11a and the comparison on the WFCI dataset is shown in Figure 12a. To rule out the potential of over-fitting, we use a smaller number of discrete states of 10 as a demonstration. Note that the same color in different figures does not indicate the same motif. In favor of BehaveNet, we apply video-wise normalization before fitting the ARHMM model to the learned embeddings. This normalization can be seen as a naive method to remove the context information from BehaveNet embeddings. For DBE embeddings, we simply apply a global normalization to all videos. As shown in Figures 11-12, our disentanglement mechanism helps ARHMM to learn consistent states across different animals and sessions, as opposed to the CAE embeddings which

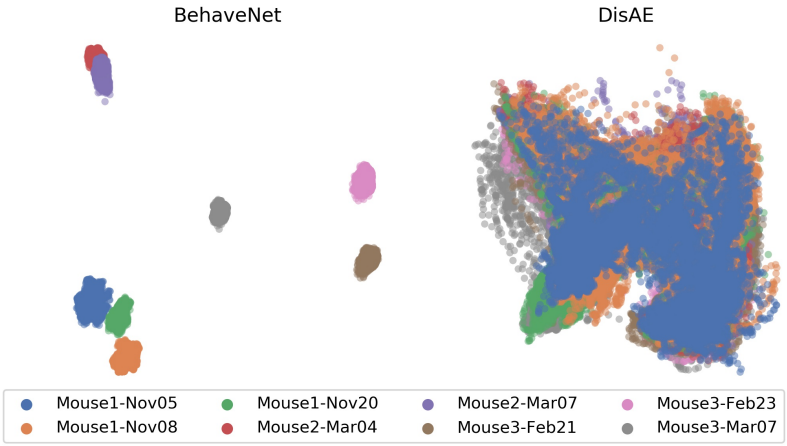


Figure 9: A comparison between BehaveNet and DBE embeddings for the Hand-reach dataset.

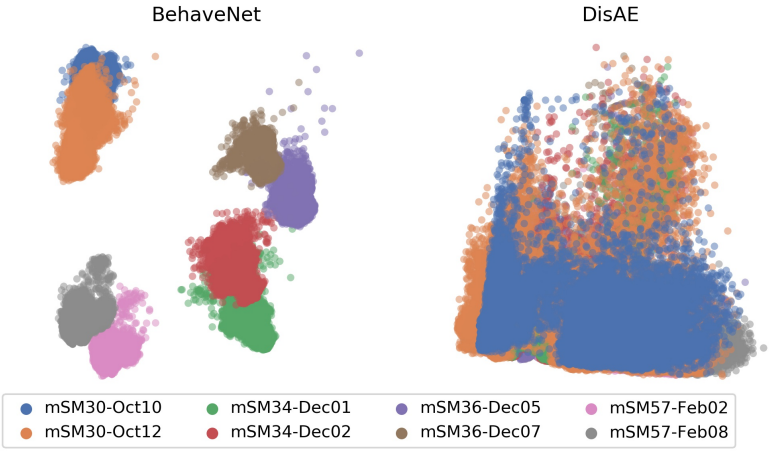


Figure 10: A comparison between BehaveNet and DBE embeddings for the WFCI dataset.

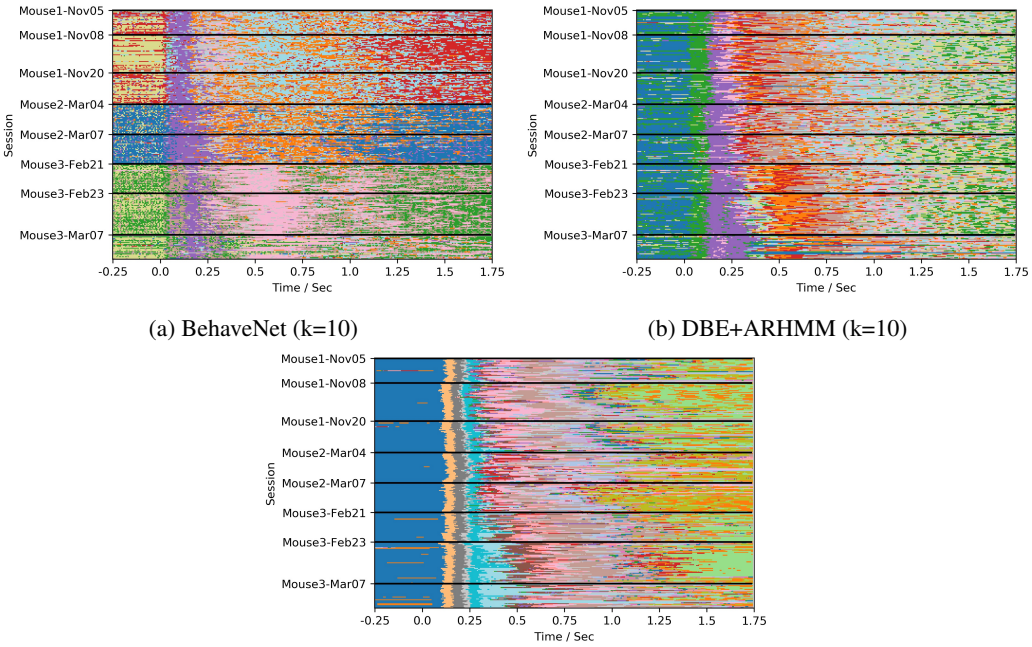
Behavenet relies on, that leads to animal-specific motifs. On the other hand, although video-wise normalization helps with different sessions of the same animals, it fails when the differences between context nuisances are more severe. Compared to the ARHMM-based segmentation, the VDBE motifs are more consistent and smoother across animals and sessions.

Note that although we use a large number (30) of VDBE clusters, not all clusters are activated. For the Hand-reach dataset, 28 out of 30 clusters are activated; and for the WFCI dataset, only 9 out of 30 clusters are activated. This demonstrates that VDBE can automatically determine the number of clusters that explains the variability of behavior.

A zoomed-in VDBE estimation on the WFCI dataset overlaid with key-point annotations is shown in Figure 13. In each trial the first vertical black bar indicates the time of Lever-In and the second vertical black bar indicates the time of Spout-In. As shown in the figure, VDBE motifs successfully identify the transitions corresponding to these events.

H Motif transitions

In this section we present the motif transition graph of the estimated states of our full VDBE model on the Hand-reach dataset. The size of each node indicates the overall number of the frames assigned to that motif, the color of each node indicates the label mapping of that motif, and the width of an edge indicates the transition probability from the source node to the target node. We filter out the edges that have probability lower than 0.1 to better visualize the main transitions. As shown in



(a) BehaveNet (k=10)

(b) DBE+ARHMM (k=10)

(c) VDBE (k=30)

Figure 11: A comparison of generated ethograms between ARHMM estimation on BehaveNet and DBE on the Hand-reach dataset. The x-axis denotes time and the y-axis denotes trials which are clustered by sessions and animals. Colors indicate unmerged motif estimations.

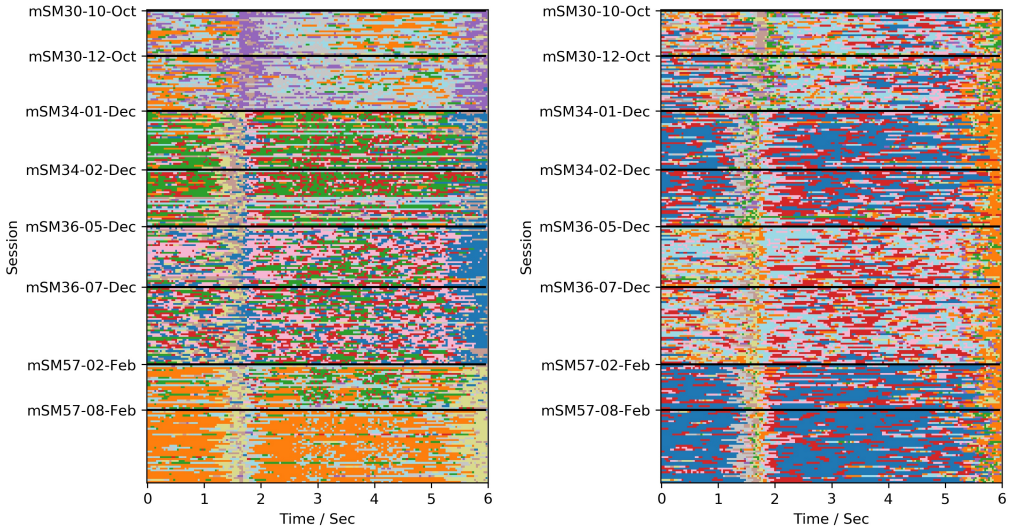
Figure 14b, motifs that are mapped to the same label are well-clustered. The transition graph of motifs corresponds to the global sequence of behavior: On-Perch→Lift→Reach→Grab→To-Mouth→At-Mouth→Back-To-Perch→On-Perch.

I VDBE latent embeddings

An example of VDBE latent factors and the estimated behavioral states (merged motifs) is shown in Figure 14a. Each color indicates one of the 7 behavioral states, and sampled frames of selected states are shown below the latent factors. As we can see, the video is segmented in a meaningful way with the motor activity happening in natural order. Note that the human labeling was based on the motion of the right fore-paw. However, our motifs take into account all motion information in the video. For example, manual inspection of the motifs that correspond to the On-perch (blue clusters) reveals that motifs are separated based on stationary (no movement), sequences of the table turning (before the cue; motifs 0, 8, 5, 21, 22, 28 in Fig. 14b) and chewing of the food pellet (after back-to-perch; motifs 24, 3, 4 in Fig. 14b). The chewing motion is also evident in the periodicity of some of the latent factors in Figure 14a. Thus, VDBE latents and motifs reveal finer resolution behavior beyond human labels.

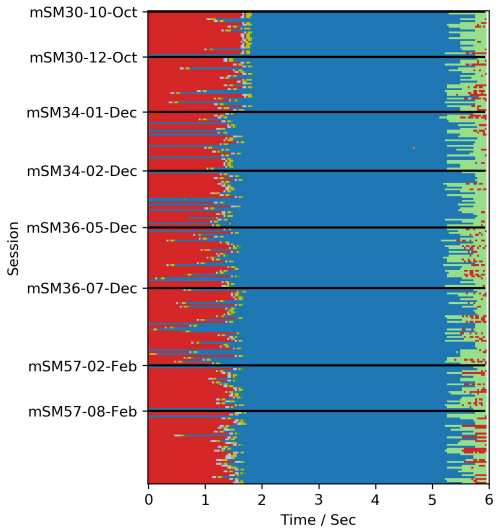
J Behavioral video generation

The source and generated videos shown in Section 4.4 are included in supplementary materials.



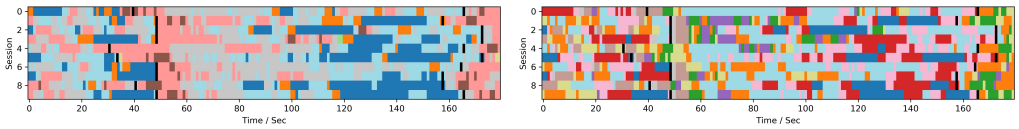
(a) BehaveNet (k=10)

(b) DBE+ARHMM (k=10)



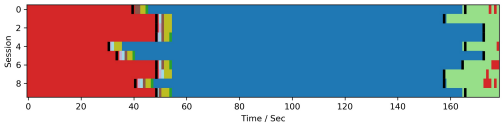
(c) VDBE (k=30)

Figure 12: A comparison of generated ethograms between ARHMM estimation on BehaveNet and DBE on the WFCI dataset. The x-axis denotes time and the y-axis denotes trials which are clustered by sessions and animals. Colors indicate motif estimations.



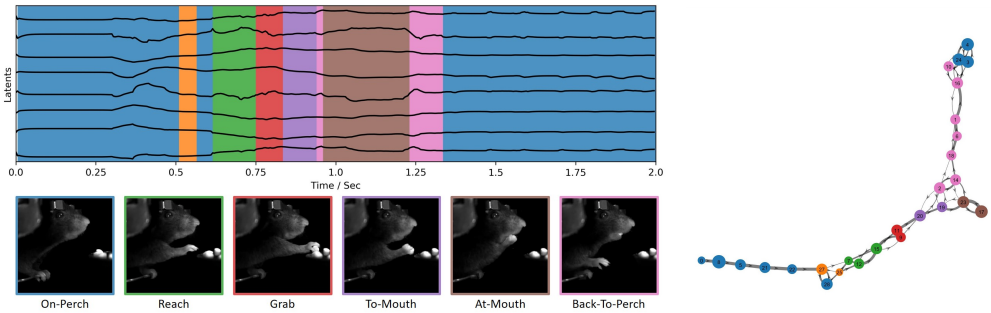
(a) BehaveNet (k=10)

(b) DBE+ARHMM (k=10)



(c) VDBE (k=30)

Figure 13: Ethograms of sampled trials from the WFCI dataset overlaid with key-point annotations in black (Lever-In and Spout-In). Colors indicate the same motifs as in Figure 12.



(a) VDBE latents.

(b) Transition graph.

Figure 14: (a) VDBE latent factors (both continuous and discrete pose representations) of a video example from the Hand-reach dataset and (b) the transition graph of VDBE motifs. Colors in (b) correspond to labels, and the same colors in (a) indicate corresponding merged states. Frames sampled from the merged states are also shown in (a).