Learning Disentangled Behavior Embeddings

Anonymous Author(s)
Affiliation
Address
email

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 a Disentangled Autoencoder (DisAE) to learn robust behavioral embeddings from unlabeled, multi-view, high-resolution behavioral videos across different animals and multiple sessions. We further combine DisAE with a stochastic temporal model to propose Disentangled Behavior Embedding (DBE), an end-to-end approach that learns meaningful discrete behavior representation 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, DisAE and DBE demonstrate better interpretability and 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 progress 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, 25, 30] and autoencoder-based dimensionality reduction methods [2]. Pose estimation methods characterize animal behaviors with the trajectories of body-part landmarks. These methods require manual labelling of landmarks within the video, then train a model (typically a deep CNN) to predict these landmarks in a supervised fashion. While this approach has been highly successful in behavior analysis, these methods 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 behaviors that are hard to track, such as facial movements.

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. Existing approaches currently suffer from certain drawbacks. First of all, canonical autoencoders can be easily biased by the varying visual nuisance 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

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

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 Autoencoder (DisAE) to learn robust behavior embedding from large, unlabeled, multi-session videos. DisAE 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. DisAE latents can be used in conjunction with temporal analysis models, e.g., VAME, to perform downstream analysis. We also propose DBE, a full end-to-end model trainable with DisAE, which further exploits the temporal structures of the pose components using a stochastic dynamical 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 and more interpretable, but also enjoy superior performances on downstream tasks such as behavioral state estimation and fine-grained behavioral motif generation.

2 Related Works

2.1 Video disentanglement

Disentanglement of the pose and the context in videos 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 [28] is an end-to-end framework that models motion (pose) and content 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 [31] proposes a disentangled VAE and shows controllable video generation on the synthetic Sprite dataset. In our approach we build a disentanglement mechanism in a similar paradigm as [31], by fixing context embeddings across time and constraining the dimension of pose embeddings.

2.2 Stochastic dynamic modeling

Latent state-space models are widely used to model the temporal structures for sequential data [3, 8, 17, 20, 7]. While these models have achieved success in many areas such as text and speech, adapting these models to high-dimensional, non-synthetic videos is still challenging to this date. Hafner et al. [11] proposes 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] proposes a latent dynamical model enhanced by residual connections for stochastic video prediction. Although other papers also tackle stochastic video generation and prediction on low-dimensional frame embeddings [1, 6, 18], these papers tackle the problems in an auto-regressive fashion and thus offers no single latent state vector that captures all 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 [22], DeepPoseKit [10] and LEAP [25] train deep neural networks to produce confidence maps for predefined landmark annotations, all in a

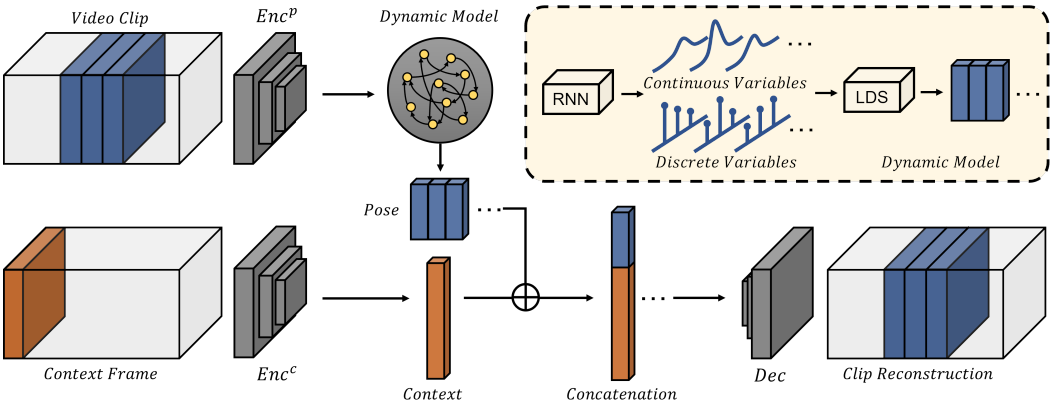


Figure 1: An overview of DBE. The pose encoder encodes individual frames into a sequence of frame embeddings, which are then passed through a stochastic dynamical model to generate pose embeddings. The context embeddings are generated from a conditional context frame by the context encoder. Both embeddings are concatenated at each timestamp as input to the decoder which reconstructs the input clip. The dynamical model (top right) uses an RNN to infer both continuous and discrete latent states that captures the transition of the frame embeddings, and use a linear state-space model for generation. When the dynamical model is not used, the full DBE model becomes DisAE.

similar fashion. Leveraging the success of DeepLabCut (DLC), Wu et al. [30] improves the quality of pose estimation by imposing additional spatial and temporal constraints on multiple landmarks during training. Luxem et al. [21] further develop 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 annotates animal behaviors based on user-defined labeling.

On the other hand, unsupervised dimensionality reduction methods like PCA have been used to analyze behaviors for decades [27, 4, 29]. 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 ARHMM model to exploits the temporal structures of the learned frame embeddings. However, BehaveNet is 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 pose from context.

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 nuisance in the video in an unsupervised fashion. Our goal is to learn a meaningful embedding $z^p_{1:T}$ that captures 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, 23], and not freely moving. Additionally, in behavior experiments, videos $(x_a, x_b)_{1:T}$ are often recorded from different views, where a and b are two views of the behavior. We treat each view as a different image and fuse the embeddings of each view as the representation of this timestamp. For the rest of this paper, we will slightly abuse the notations and omit the view subscripts, e.g. we will use $x_{1:T}$ to refer to the multi-view data $(x_a, x_b)_{1:T}$.

3.2 Disentanglement

In order to factor out the non-behavioral factors, our approach utilizes two separate 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. The context z^c of each view is concatenated into a single context embedding, 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 $x_{1:T}$ for each view, which are then concatenated into the pose embeddings $z^p_{1:T}$. The temporal structures of the pose component are exploited by a stochastic dynamical 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 these concatenated embeddings are further decoded into frame reconstructions by two convolutional decoders D for each view. The whole architecture can also be seen as a hybrid model with a deterministic path context embedding and a probabilistic path of pose embedding.

Because the context embedding z^c is time-invariant and 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 Yingzhen and Mandt [31]. 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 pose embeddings of different sessions form distinct clusters. The disentangled context embedding, on the other hand, capture these nuisances and form into clusters by sessions as expected.

3.3 Stochastic dynamic modeling

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. 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 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 the behaviors. 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 and for the purpose of motif segmentation because it offers no discrete states.

A more suitable choice for capturing such multi-modality is a Gaussian mixture model [26]. Shu et al. [26] 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.

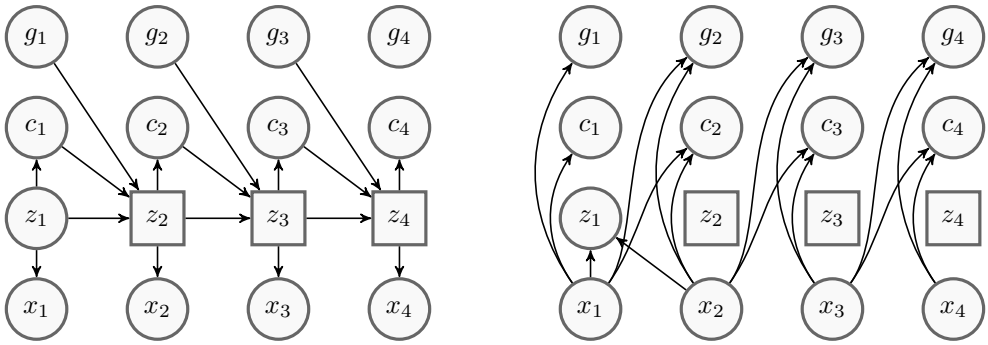
Generation The generative procedure is shown in Figure 2a. We model the transition between the previous z_{t-1} and the current state z_t using a Gaussian mixture. The generative model is as follow:

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

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

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

where π_θ and f_θ are linear projection layers. At each timestamp, stochastic variables c_t and g_t models the discrete state and the additional stochasticity, then the continuous state z_t is generated



(a) Generation (b) Inference

Figure 2: The generation (a) and inference (b) process of our stochastic dynamical 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 purpose of clarity.

with a deterministic path from z_{t-1}, c_t and g_t. Note that although both c_t and g_t capture the stochasticy 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. We assume the covariance matrix of each Gaussian component is diagonal and $\odot$ stands for 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 a 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 representation 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 method on two multi-session video datasets: a hand reach to pellet motor behavior dataset we collected (Hand-reach), and the behavioral videos from a decision making dataset, the Wide-Field Calcium Imaging dataset (WFCI) [23, 24].

The hand reach dataset contains videos of 263 successful trials of 3 expert animals (who have learned the motor task) in 8 sessions, and includes fine-grained behavior. An illustration of the hand reach dataset is shown in Figure 3. Head-fixed mice are trained to reach from their perch to a food pellet. Videos are acquired on multiple dates, within which the experimental layout is potentially different, e.g., location of the table. Two cameras simultaneously record a front view and a side view. Each video is 12 seconds long, with a frame rate of 200Hz. A 0.2 second cue is given 4 seconds after the start of the trial, which we set as the origin (0) of our time axis; shortly after table-turning brings the food in front of the mouse. 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. As the mice in this dataset are experts, and thus mostly stationary before the cue and chewing 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 frames are sampled at one second before the cue, when most of the subjects are stationary.

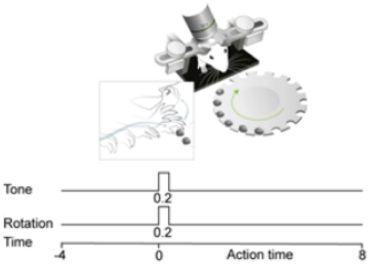


Figure 3: Hand-reach dataset.

In the WFCI dataset (previously used and partially released in [2]) head-fixed mice perform a visual decision making task of drinking from spouts using levers. 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 compare our methods against two behavior embedding methods: BehaveNet [2] and VAME [21], with implementation details of each provided in Appendix C and Appendix D, respectively.

We also train DisAE without the stochastic dynamical modeling component. We then use the DisAE embeddings in two variations: DisAE+ARHMM which replaces the CAE embeddings with DisAE embeddings in the BehaveNet model, and DisAE+VAME, which replaces the DLC trajectories with DisAE embeddings in VAME. In Section 4.1 and 4.2, we use the DisAE embeddings to demonstrate the efficacy of our disentanglement component. In Section 4.3, we show both variations relying on DisAE achieve superior results 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. DisAE as well as our full DBE model generates more session-agnostic pose embeddings than the plain CAE autoencoder. 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).

We further visualize the advantage of disentanglement in Figure 4. We plot a 2D PCA applied to the BehaveNet and DisAE 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 there are new sessions, and learn a mapping across animals to relate the embeddings of the same behavior. In contrast to BehaveNet, our DisAE nicely aligns the two videos in the pose embedding space. As we will show in the later section, this consistency is crucial to state estimation and behavior

decoding. More results comparing BehaveNet and DisAE embeddings, as well as ethograms of both Hand-reach and WFCI datasets, are provided in Appendix E and F.

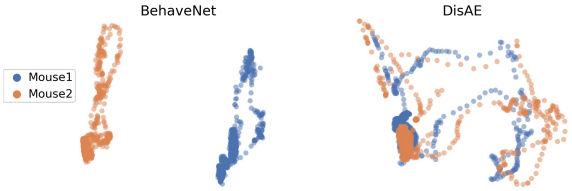


Figure 4: Embeddings of 2 similar videos (hand-reach)

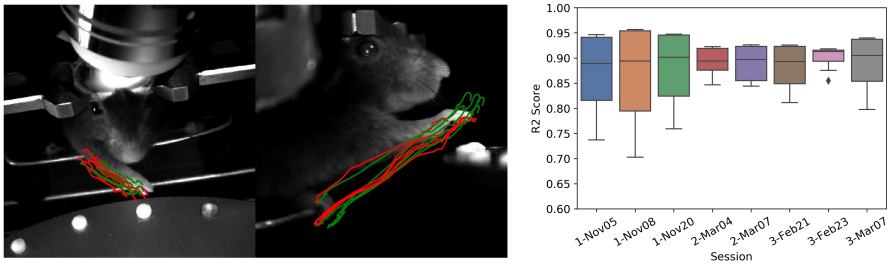
Model	Hand-reach	WFCI
BehaveNet	100.00	99.99
DLC	21.19	-
DisAE	28.21	35.49
DBE	17.94	19.54

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 a meaningful behavior embedding, we perform a simple linear regression task from the DisAE embedding 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 training.



(a) DLC trajectories prediction

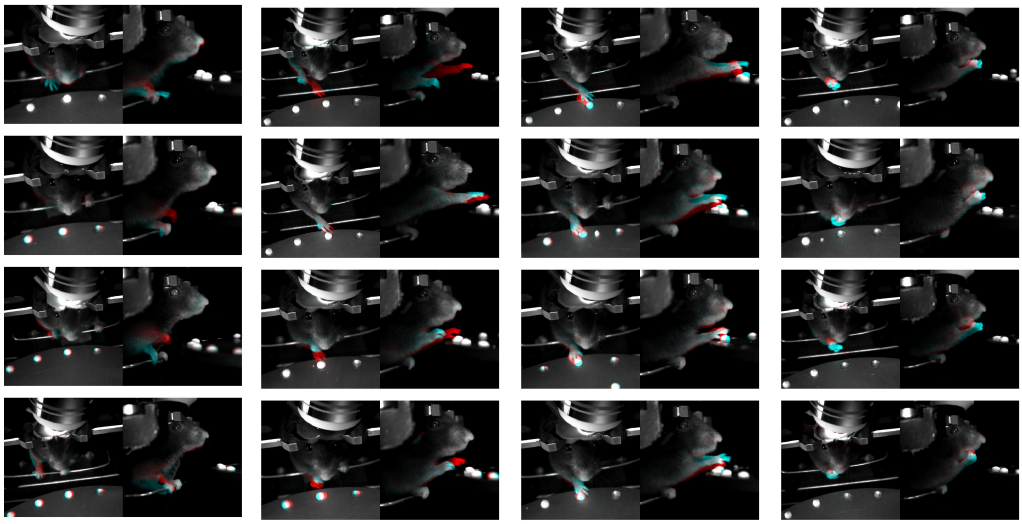
(b) Goodness-of-fit of linear regression

Figure 5: Predicting DLC trajectories from disentangled embeddings. (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 prediction are well aligned with the true DLC trajectories, capturing the right paw of the mouse moving from the perch to the food 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 DisAE embeddings contain 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 behaviors, especially when no supervision is available. In this section, we compare our approach with other competing methods for motif generation. 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 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 generation methods, i.e. clustering. We then assign these motifs to the labeled set of behaviors, 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 to be larger than the number of target motifs for



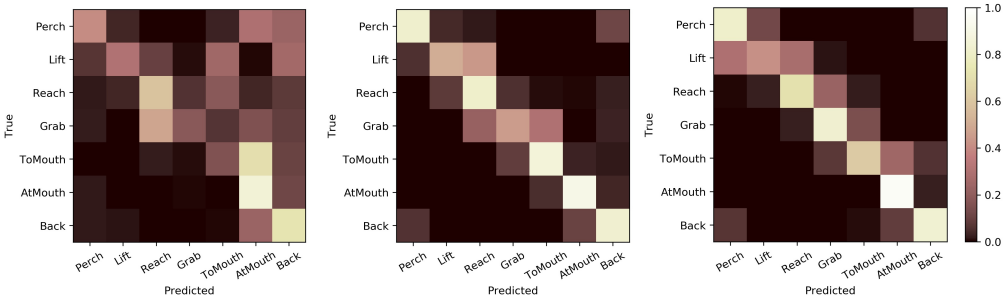
(a) Lift (b) Reach (c) Grab (d) At-Mouth

Figure 6: DBE motifs images. The blue channel indicates the start of the motif and the red channel indicates the end. Note that in each columns the behavior is consistent but the context is varying (location of the table and identity of the mouse).

Table 2: Multi-session motif segmentation

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

two reasons. The first reason is that only part of the frames are labeled and the target motifs do not cover all the possible behaviors of the mice. Extra clusters are necessary to capture these additional behaviors. 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.



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

Figure 7: Confusion matrices of BehaveNet (no disentanglement), DisAE+VAME (with disentanglement) and DBE (with disentanglement).

We evaluate the performance of motif segmentation on three clustering metrics, accuracy, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). As shown in Table 2 and Figure 7, disentanglement is the key to achieve good behavioral clusters across different animals and multiple sessions. With disentanglement, the confusion matrices are more diagonal which indicates better clustering. Combining the DisAE for frame embedding and 30-cluster VAME (DisAE+VAME) for temporal dynamics and segmentation achieves the best accuracy, NMI and ARI. Our full DBE 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 benefit motif segmentation. Figure 6 shows examples of random segments from DBE motifs corresponding to 4 different labels. Note that DBE 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 DBE and ARHMM-based models against VAME is that we can decode synthetic videos from behavior embeddings (Table 2), which demonstrates better interpretability and allows potential application to neural decoding. In this section, we further investigate behavior decoding as shown in Figure 8. To generate synthetic videos, we use the inferred discrete behavior representation $c_{1:T}$ from the source video and the context embedding from the target sessions, while $g_{1:T}$ and z_1 are sampled from their prior distributions. Note that the behaviors in the source video are replicated in both generated videos with the same and different contexts. This indicates that our model successfully learns the dynamics of the behaviors. Future works will explore using DBE for neural decoding.

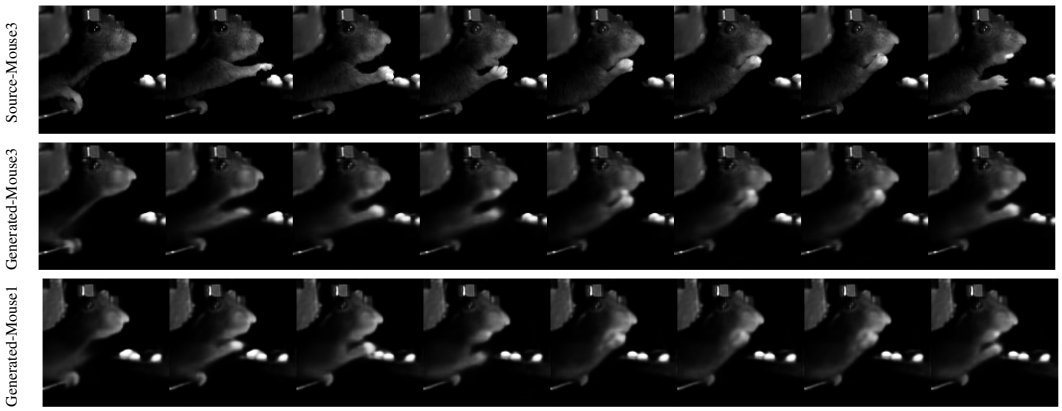


Figure 8: Source (top row) and generated videos of the same discrete behavior sequence, with the same (middle row) and different (bottom row) contexts. Each row is plotted at intervals of 20 frames.

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 variational sequential autoencoder 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. First, 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. Second, our disentanglement assumes the non-behavioral nuisance is stationary through a video, which is not always the case. Future works should benefit from allowing slow dynamics in context embeddings. Regarding societal impact of our study 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.

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] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The hand-reach dataset will not be released.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code will be provided in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Previous works are properly cited in Section 4. We will make clear unblinded acknowledgement if accepted.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]