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

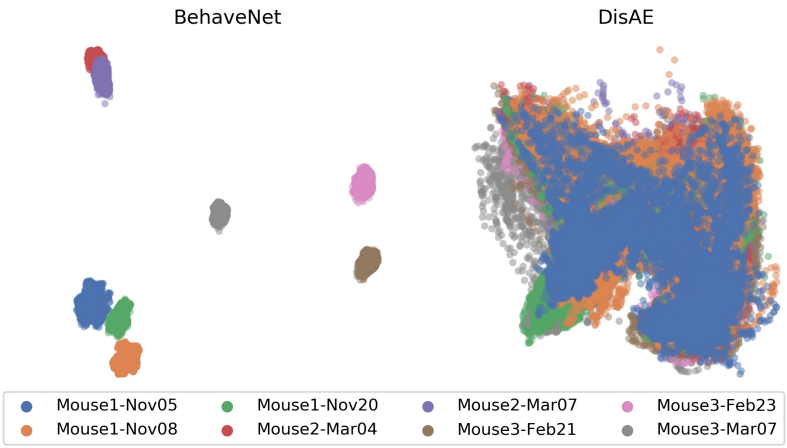


Figure 9: A comparison between BehaveNet and DBE embeddings for the Hand-reach dataset.

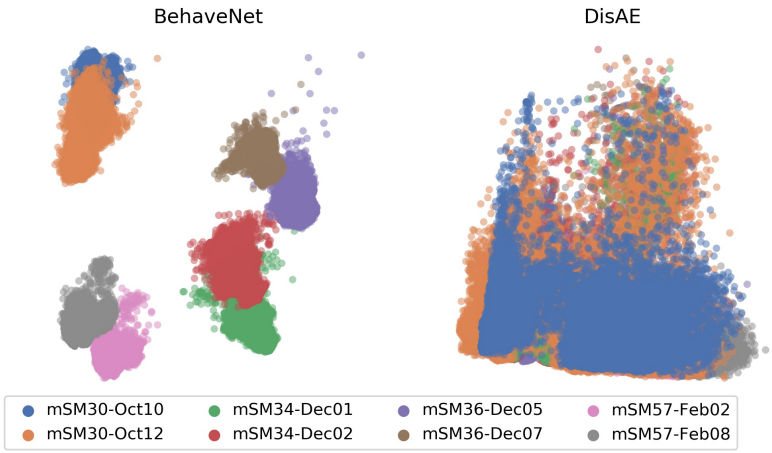


Figure 10: A comparison between BehaveNet and DBE embeddings for the WFCI dataset.

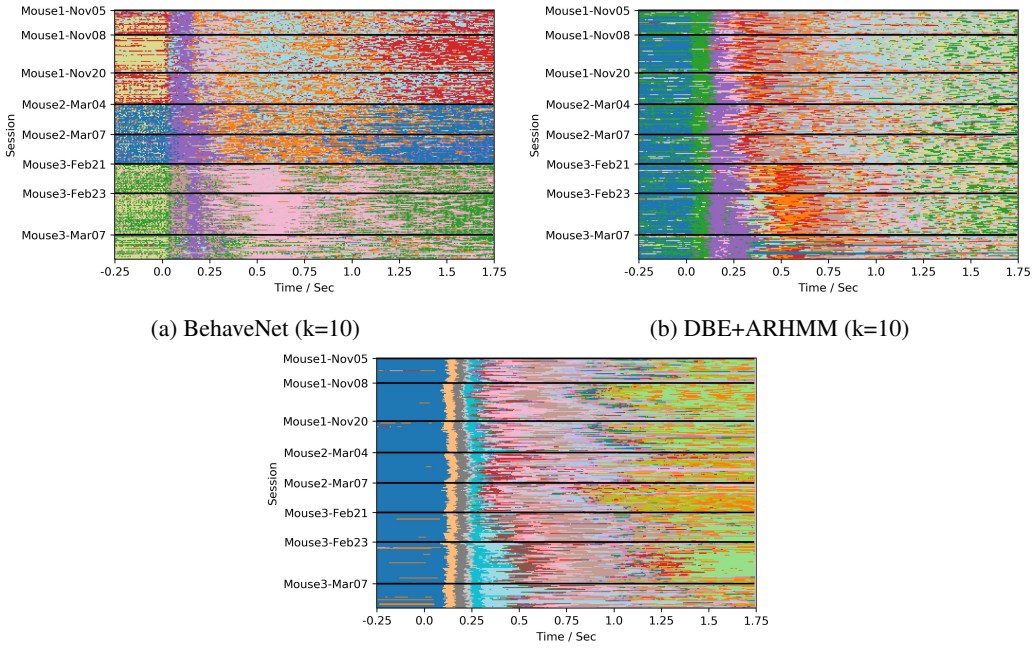
Behavenet relies on, that leads to animal-specific motifs. On the other hand, although video-wise normalization helps with different sessions of the same animals, it fails when the differences between context nuisances are more severe. Compared to the ARHMM-based segmentation, the VDBE motifs are more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

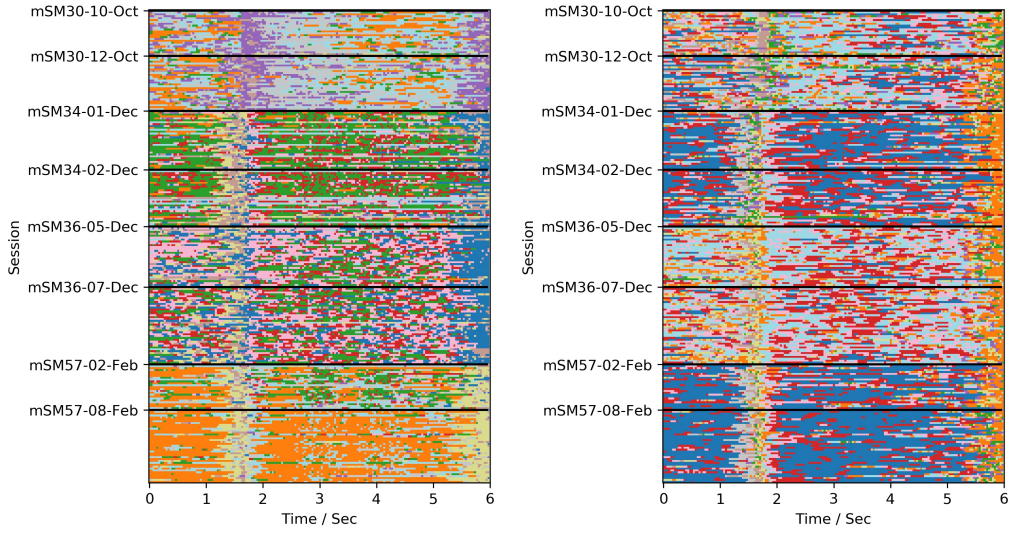
Figure 14b, motifs that are mapped to the same label are well-clustered. The transition graph of motifs corresponds to the global sequence of behavior: On-Perch→Lift→Reach→Grab→To-Mouth→At-Mouth→Back-To-Perch→On-Perch.

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.

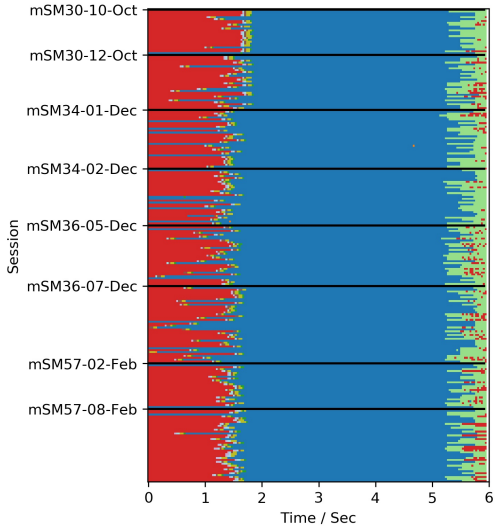
J Behavioral video generation

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



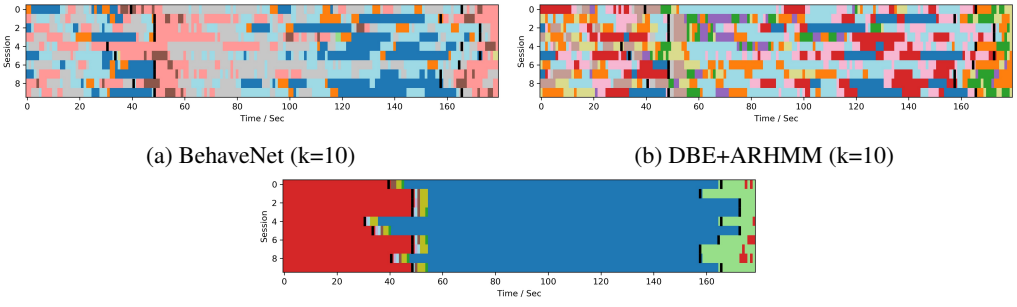
(a) BehaveNet (k=10)

(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.

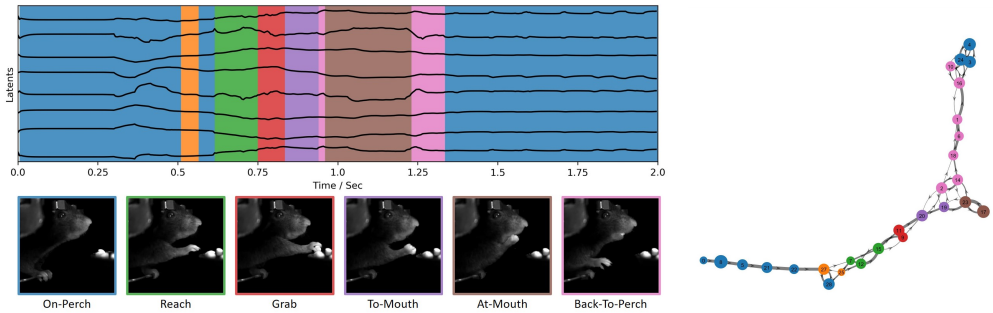


(a) BehaveNet (k=10)

(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).