Semi-supervised sequence modeling for improved behavioral segmentation[*]

Matthew R Whiteway[†] Evan S Schaffer, Anqi Wu, E Kelly Buchanan,
Omer F Onder, Neeli Mishra, Liam Paninski
Columbia University
New York, USA

Abstract

A popular approach to quantifying animal behavior from video data is through discrete behavioral segmentation, wherein video frames are labeled as containing one or more behavior classes such as walking or grooming. Sequence models learn to map behavioral features extracted from video frames to discrete behaviors, and both supervised and unsupervised methods are common. However, each approach has its drawbacks: supervised models require a time-consuming annotation step where humans must hand label the desired behaviors; unsupervised models may fail to accurately segment particular behaviors of interest. We introduce a semi-supervised approach that addresses these challenges by constructing a sequence model loss function with (1) a standard supervised loss that classifies a sparse set of hand labels; (2) a weakly supervised loss that classifies a set of easy-to-compute heuristic labels; and (3) a self-supervised loss that predicts the evolution of the behavioral features. With this approach, we show that a large number of unlabeled frames can improve supervised segmentation in the regime of sparse hand labels and also show that a small number of hand labeled frames can increase the precision of unsupervised segmentation.

1. Introduction

Behavioral segmentation is an indispensable tool for quantifying natural animal behavior as well as understanding the effects of targeted interventions [1, 35, 44]. This procedure begins with the collection of raw behavioral data during an experiment, typically with video or motion-capture sensors. In the supervised segmentation setting, the experimenter then labels a subset of frames that contain behaviors of interest, such as walking, grooming, rearing, etc. Finally, a machine learning algorithm (referred to here as a "sequence model") is trained to match each frame with the corresponding behaviors [16, 18, 20, 31, 32, 6, 33, 39, 41, 43]. As the scale of behavioral data continues to grow [11, 44], it becomes infeasible to densely label behaviors in every video. Therefore, it is crucial to develop segmentation techniques that perform well with sparsely labeled data.

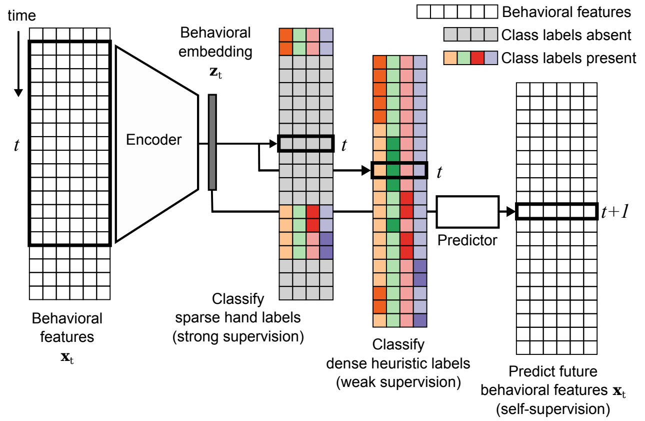

Figure 1. Temporal sequence models for behavioral classification can be augmented with a weakly supervised loss that classifies heuristic labels and a self-supervised loss that predicts future behavioral features in order to improve model performance on sparsely labeled datasets.

Fully unsupervised behavioral segmentation algorithms are a complementary approach that require no hand labels [4, 17, 47, 14, 26]. These algorithms typically reduce the dimensionality of the raw video data through various methods, then perform unsupervised clustering on the resulting low-dimensional behavioral representation [9]. These unsupervised methods tend to be more scalable than their supervised counterparts because they do not require manual input. Another benefit of these methods is their ability to discover new behaviors [9, 35]. However, there may be certain behaviors of particular interest for downstream analyses, and unsupervised methods cannot guarantee they accurately segment these behaviors.

Here we propose a semi-supervised segmentation algo-

[*] An abbreviated version of this work is to appear at the CVPR 2021 CV4Animals Workshop.

[†] Correspondence: m.whiteway@columbia.edu

rithm that combines the strengths of these two approaches while minimizing their weaknesses: we take advantage of a small number of hand labels to ensure particular behaviors are well-represented by the model and also take advantage of a large number of unlabeled frames in order to improve the behavioral representation (Fig. 1). We find that this approach improves supervised segmentation while still allowing for unsupervised behavior discovery.

We propose two simple losses that extract relevant information from unlabeled frames, which augment a standard supervised classification loss computed on labeled frames (e.g. cross entropy). The first loss is based on the observation that the behaviors of interest can often be described heuristically: for example, when tracking the paws and the nose of a mouse with pose estimation software [29], candidate bouts of grooming can be identified by finding all frames where the distance from the paws to the nose is below some threshold. This procedure may lead to false positives (e.g. when the mouse grabs a lick spout near its nose) and false negatives (when the distance threshold is set too low), but it is a straightforward and computationally efficient way to automatically label many frames in a video. We propose, then, to use these so-called "heuristic" labels as an additional supervised signal (which we refer to as *weak* supervision) for sequence models.

The second loss is based on the observation that behavior is *dynamic*; even a "still" behavior is defined in the context of what the animal was doing before and after a "still" frame. Therefore, we propose to add a self-supervised prediction loss so the model learns to map the behavioral features at time t to the behavioral features at time t+1. Note that this prediction loss can be computed for every single frame of a video, without the need for corresponding hand or heuristic labels, again allowing the sequence model to take advantage of potentially large amounts of unlabeled data.

We evaluate our proposed semi-supervised approach by conducting an empirical evaluation on a head-fixed fly dataset. We find that, individually, the weak and self-supervised losses improve supervised segmentation metrics across all behaviors; combining these losses leads to additional improvements. We also compare our approach to fully unsupervised behavioral segmentation models and show that adding a small number of hand labels can improve segmentation while still allowing for the discovery of previously unlabeled behaviors. Code is available at `https://github.com/themattinthehatt/daart`.

Related Work. We focus on behavioral segmentation using pose estimates, and therefore our work draws from the literature on skeleton-based action understanding [13]. [10] shows that motion prediction is a good auxiliary task for behavioral segmentation when using sparse hand labels; we build upon this work by including an additional set of heuristic labels that can strengthen the classifier even further. [42] introduces a set of "task program" heuristics to shape a latent behavioral embedding used by downstream behavior classifiers; while this is similar in spirit to our approach, we choose instead to provide direct heuristics for each hand-labeled behavior, and train our model end-to-end. [24, 25] use a classification and prediction loss to perform semi-supervised action recognition (different from segmentation), though their focus is on an active learning scheme for determining the most informative data points for future labeling. Several other works introduce weak supervision terms to improve segmentation [7, 15, 21, 36], though they rely on relatively strong assumptions, such as a known ordering of behaviors, that are not relevant for our use case.

If we use our model with the self-supervised loss only, we can perform fully unsupervised behavioral segmentation by performing a post-hoc clustering of the low-dimensional behavioral embeddings. This approach follows a common pipeline successfully used across many studies [9]. For example, [26] use an autoencoder RNN to produce a behavioral embedding from pose estimates, apply UMAP [30] to further reduce the dimensionality, then apply k-means clustering to perform unsupervised behavioral segmentation. Other recent works use different combinations of algorithms for embedding, dimensionality reduction, and clustering [4, 47, 17, 3, 27]. Our work expands this pipeline to include hand and heuristic labels, and we show that this semi-supervised approach can produce higher quality segmentations.

2. Methods

Most approaches for supervised behavioral segmentation from video data involve a two-step process: first, for each frame $\mathbf{f}_t$, compute a lower-dimensional feature representation $\mathbf{x}_t$ that encodes local spatiotemporal information using pose estimation [28, 12, 34, 48], Dense Trajectories [45], or two-stream network outputs [40, 6]; second, a sequence model $f(\cdot)$ (such as a recurrent neural network) maps the behavioral feature vector $\mathbf{x}_t$ (or a window of these features) to a discrete label vector $\hat{\mathbf{y}}_t$, which should match the hand labels $\mathbf{y}_t$. We assume that the hand labels are only defined on a subset of time points $\mathcal{T} \subseteq \{1, 2, ...T\}$. The cross-entropy loss function $\mathcal{L}_{\text{xent}}$ [5] then defines the supervised objective ($\mathcal{L}_{\text{super}}$) to optimize:

$$\mathcal{L}_{\text{super}} = \sum_{t\in\mathcal{T}} \mathcal{L}_{\text{xent}}\big(\mathbf{y}_t, f(\mathbf{x_t})\big). \qquad (1)$$

Weak and self-supervised loss functions. We now introduce a set of heuristic labels $\tilde{\mathbf{y}}_t$, defined at each time point. Computing the cross-entropy loss on all time points that

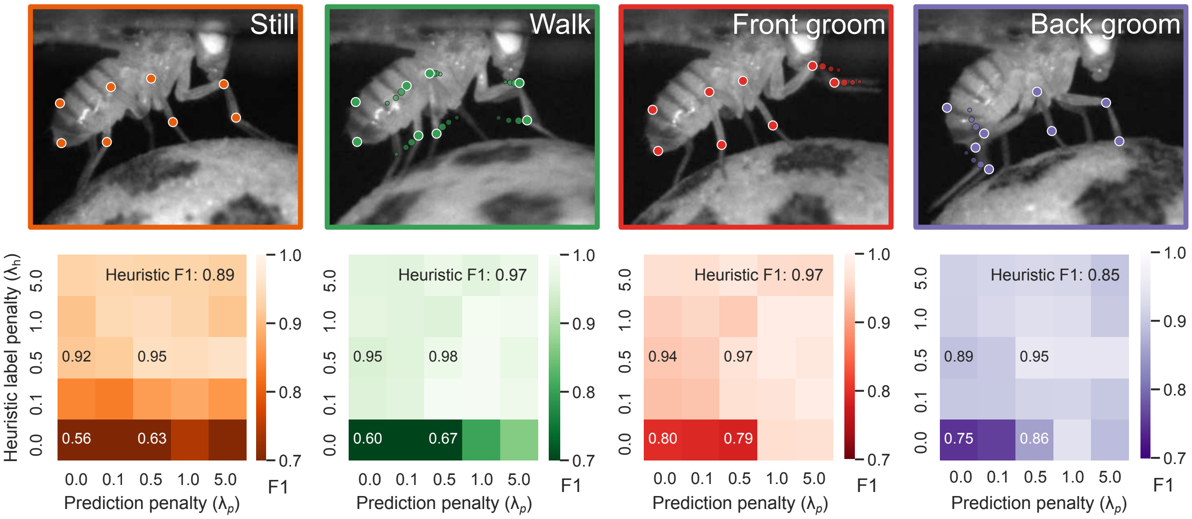

Figure 2. *Top*: Example frames from four fly behavior classes, each overlaid with the eight points tracked through pose estimation. Smaller trailing dots indicate the dynamics of each tracked point during the specified behavior. *Bottom*: F1 score for each behavior on test data as a function of the weak supervised penalty based on heuristic labels (λ_h) and the self-supervised prediction penalty (λ_p). F1 score is overlaid in text for selected hyperparameter combinations; see Fig. S2 for all values. Higher F1 scores are better, with a maximum of value of 1. Standard supervised classification corresponds to $\lambda_h = \lambda_p = 0$.

do not already have a corresponding hand label defines the heuristic objective:

$$\mathcal{L}_{\text{heur}} = \sum_{t \notin \mathcal{T}} \mathcal{L}_{\text{xent}}\big(\tilde{\mathbf{y}}_t, f(\mathbf{x_t})\big). \tag{2}$$

The self-supervised loss requires the sequence model to predict $\mathbf{x}_{t+1}$ from $\mathbf{x}_t$. To properly describe this we now expand the definition of the sequence model $f(\cdot)$ to include two components: an *encoder* $e(\cdot)$, which maps the behavioral features $\mathbf{x}_t$ to an intermediate behavioral embedding $\mathbf{z}_t$; and a (linear) *classifier* $c(\cdot)$ which maps $\mathbf{z}_t$ to the predicted discrete labels ($\hat{\mathbf{y}}_t = c(e(\mathbf{x}_t))$. We can now incorporate the self-supervised loss through the use of a *predictor* function $p(\cdot)$, which maps $\mathbf{z}_t$ to $\hat{\mathbf{x}}_{t+1}$, and match $\hat{\mathbf{x}}_{t+1}$ to the true behavioral features $\mathbf{x}_{t+1}$ through a mean square error loss $\mathcal{L}_{\text{MSE}}$ computed on all time points:

$$\mathcal{L}_{\text{pred}} = \sum_{t=1}^{T-1} \mathcal{L}_{\text{MSE}}\big(\mathbf{x}_{t+1}, p(e(\mathbf{x}_t))\big). \tag{3}$$

Finally, we combine all terms into the full semi-supervised loss function:

$$\mathcal{L}_{\text{semi}} = \lambda_s \mathcal{L}_{\text{super}} + \lambda_h \mathcal{L}_{\text{heur}} + \lambda_p \mathcal{L}_{\text{pred}}, \tag{4}$$

where the λ terms are hyperparameters that control the contributions of their respective losses. Note that setting $\lambda_h = \lambda_p = 0$ results in a fully supervised model, while $\lambda_s = \lambda_h = 0$ results in a fully unsupervised model.

Model. Our approach is agnostic to the particular architecture of the encoder and predictor networks. Here we use a standard recurrent neural network with GRU layers [8], which sees frequent use in sequence modeling [2]. Both $e(\cdot)$ and $p(\cdot)$ are modeled with two layer bidirectional GRU networks (32 cells per layer). We performed a small hyperparameter search across number of layers, cells per layer, and learning rate, and found that our results are robust across different settings (data not shown).

Data. We evaluate our approach on a head-fixed fly dataset [38] (Fig. 2). This dataset contains videos from 10 flies, with videos ranging in length from 10 to 34 minutes. We first track eight points on the fly using Deep Graph Pose [48], and use these points as our behavioral features $\mathbf{x}_t$ (see Fig. 2). The flies exhibit a small range of easily identified behaviors, including standing still, walking on an air-supported ball, and front and back grooming (Fig. 2, top). We label up to 300 frames for each of these behaviors per fly (for a total of 1.1% of all frames labeled; see Table S1). We also devise a simple set of heuristics to produce weak labels for each behavior: frames are labeled "Walk" when the ball motion energy (ME) is above a threshold; "Still" when the ME of the limb markers is below a threshold; and "Front (Back) groom" when the fly is not walking, and the ME of the forelimb (hindlimb) markers is above a threshold.

Training and evaluation. We use 5 fly videos for training and 5 for testing. All models are trained with the Adam

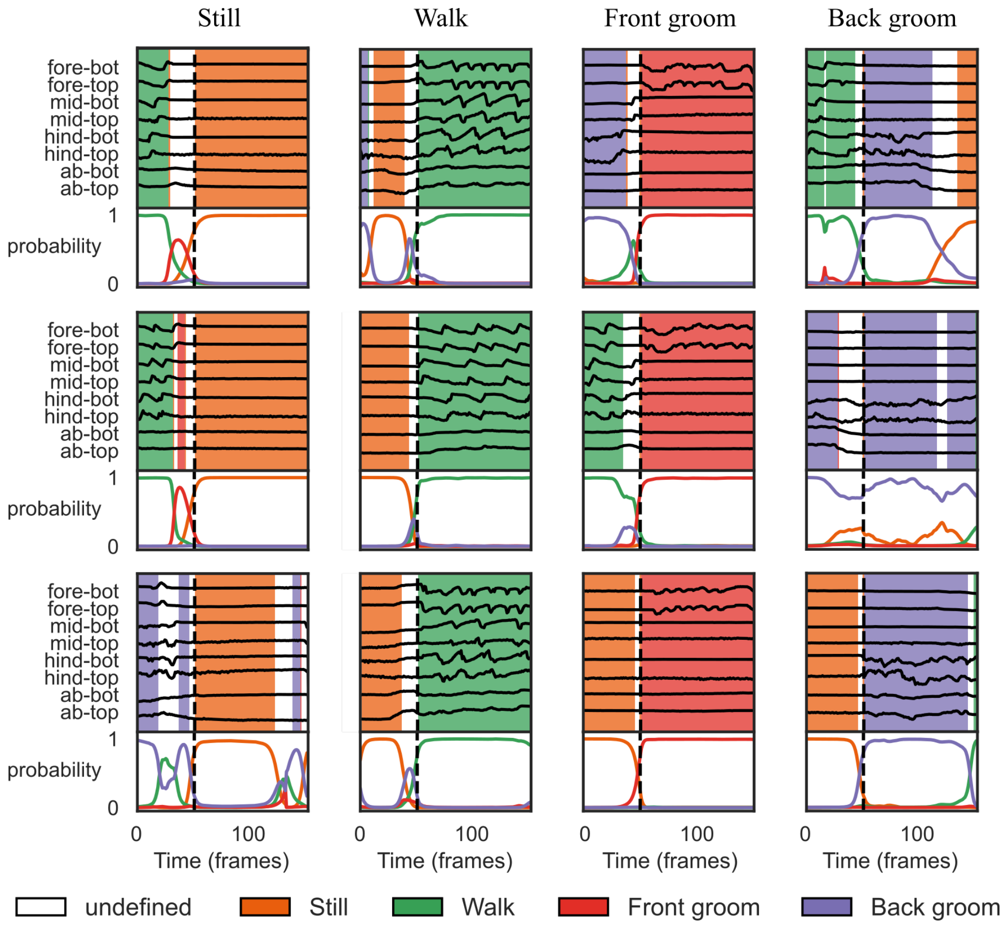

Figure 3. Sequence model outputs for sample time segments. Each panel shows the markers (top, black lines; only x-coordinates are displayed). Background color denotes the highest probability behavior class. The background color is white (”undefined” behavior) if the largest probability is less than an arbitrary threshold of 0.75. The actual class probabilities are plotted below the markers. Each column displays a set of random examples from each behavior class; behavior bout onset is indicated by the vertical black dashed line. Probabilities correspond to the GRU model with $\lambda_h = 0.5$, $\lambda_p = 1.0$, which achieved the highest F1 score averaged over all behaviors on test data. See Fig. S1 for corresponding outputs of the fully supervised sequence model.

optimizer [19] using an initial learning rate of $1\text{e}{-4}$ and a batch size of 2000 time points. For the 5 training flies, 90% of frames are used for training, 10% for validation. Training is terminated once the loss on validation data begins to rise for 20 consecutive epochs; the epoch with the lowest validation loss is used for testing. To evaluate the models, we compute the F1 score – the geometric mean of precision and accuracy – on the hand labels of the 5 held-out test flies.

3. Results

We first consider the effect of the weak and self-supervised losses on supervised segmentation (and set $\lambda_s = 1$). The weak supervision provided by the heuristic labels, controlled by λ_h, dramatically improves F1 scores across all four behavior classes, compared to the supervised baseline of $\lambda_h = \lambda_p = 0$ (Fig. 2, bottom). The presence of the high-quality hand labels also allows the model to surpass the F1 score of the lower-quality heuristic labels, especially on the more difficult behaviors (see text overlaid on heatmaps in Fig. 2). Next we consider the effect of self-supervision through the next-step-ahead prediction, controlled by λ_p. Nonzero values of λ_p also improve F1 scores across all four behavior classes. Finally, we find that combining both loss functions can slightly increase F1 yet again (for example see F1 scores for $\lambda_h = \lambda_p = 0.5$). Fig. 3 shows marker traces and their behavior class probabilities from the best performing model for several sample time segments; Fig. S1 shows the corresponding class probabilities from the fully supervised model ($\lambda_h = \lambda_p = 0$), and demonstrates how the absence of the weak and self-supervised losses leads to more errors in the segmentation.

The results above utilize a GRU architecture; to ensure

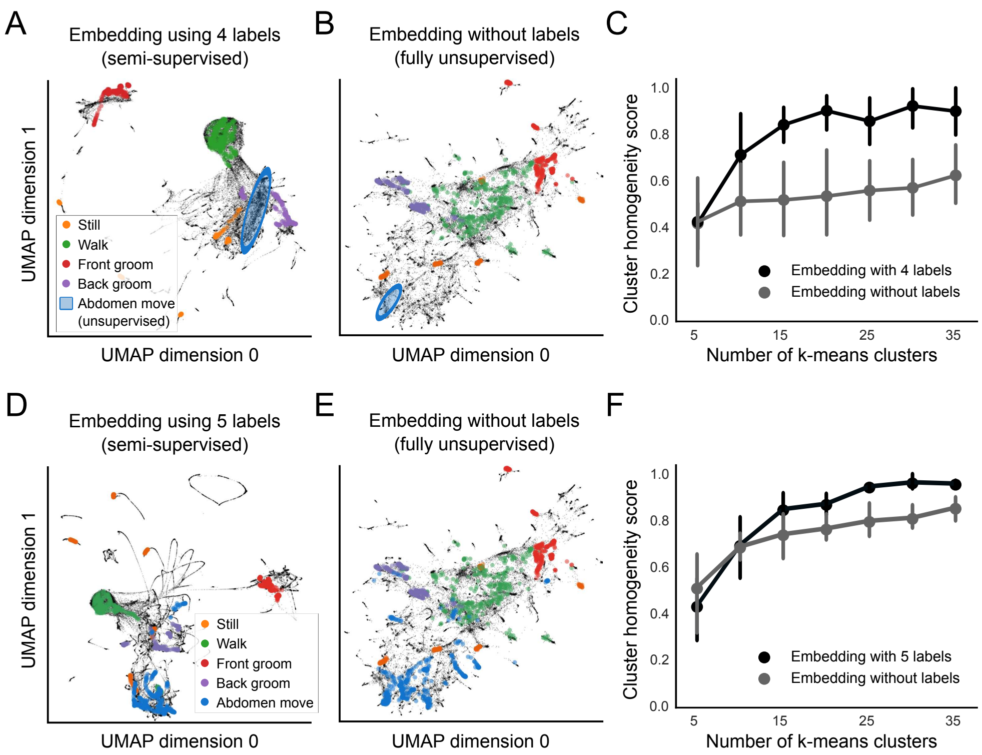

Figure 4. Hand labels improve unsupervised behavioral clustering. *A*: 2D UMAP embedding of behavior colored by hand labels; the model is trained with both hand and heuristic labels from 4 behavior classes: "Still", "Walk", "Front groom", and "Back groom". *B*: 2D UMAP embedding for the model trained without labels. *C*: The addition of hand labels produces more homogeneous clusters in the 2D space. *D*: Same as panel A, except model is trained with labels from 5 behavior classes - the previous 4 plus "Abdomen move". *E*: Same as panel B, with additional "Abdomen move" points colored by new hand labels. *F*: Same as panel C, but now cluster homogeneity score is computed with 5 behavior classes.

these performance gains are not architecture-dependent, we perform the same hyperparameter search over λ_h and λ_p using two additional architectures: a temporal convolutional network [23, 22] and an MLP neural network with an initial 1D temporal convolutional layer [3, 46] (Fig. S2). We find that for all three architectures the addition of the weak and self-supervised losses drastically improves F1 scores over standard supervised classification with the hand labels.

We next investigate the behavioral embeddings $\mathbf{z}_t$ by visualizing them for a single test fly in a 2D space through UMAP [30] (Fig. 4A). The points with corresponding hand labels are colored, revealing that similar behaviors are clustered together. In order to determine how much of this structure can be attributed to the labels, we refit the model with the self-supervised loss only ($\lambda_s = \lambda_h = 0$). The 2D visualization also shows clustered behaviors (Fig. 4B), but with more overlap of different behavior classes. Performing segmentation via clustering on this fully unsupervised behavioral embedding – a standard approach [4, 26, 27] – may therefore result in misclassified behaviors.

To quantify this observation, we next perform k-means clustering in this 2D space for both models (with and without labels). We then compute a cluster homogeneity score [37] that measures the extent to which the k-means clusters contain data points from a single behavior class. The model trained with labels achieves a higher score than the fully unsupervised model trained solely with the prediction loss (Fig. 4C). This result demonstrates how adding a small number of hand labels (and/or a larger number of heuristic labels) can improve unsupervised behavioral segmentation with little additional effort.

A primary benefit of unsupervised segmentation is the ability to discover new behaviors in an unbiased manner [9, 35]. For example, one of the k-means clusters from the unsupervised embedding (indicated by the blue ellipse in Fig. 4B) corresponds to periods where the fly moves its abdomen (see marker traces in Fig. S3), a behavior not included in our hand or heuristic labels. Because our semi-supervised model learns to predict behavior at the next time step, it too can potentially capture these unlabeled behaviors. Indeed, we find a k-means cluster in the semi-supervised embedding that also corresponds to this novel abdomen movement behavior (blue ellipse in Fig. 4A).

Next we demonstrate how to utilize this behavioral discovery to further refine the behavioral segmentation. We first hand label the abdomen movement behavior for each of the 10 flies (see Table S1 for details), as well as recompute the heuristic labels: frames are now labeled "Abdomen move" when the ME of the abdomen markers is above a threshold. We retrain the sequence models with these new labels, and find that we can accurately capture this new behavior class (Fig. S3).

We repeat our previous cluster homogeneity analysis, and again find that our semi-supervised approach produces an embedding with a representation of discrete behaviors that is more precise than the fully unsupervised embed-

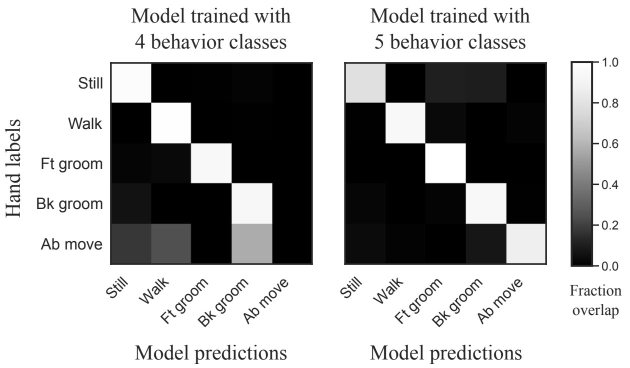

Figure 5. Inclusion of the "Abdomen move" behavior class refines behavioral segmentation. *Left*: Fraction of overlapping frames between the hand labels (y-axis) and model predictions (x-axis) for each behavior class, computed over all test data. Rows sum to 1. Overlap results are shown for best model trained using four behavior classes (same as Figs. 3, 4A, B). *Right*: Overlap results for best model trained using five behavior classes (same as Figs. 4D, E, S3).

ding (Fig. 4D-F). Furthermore, we can show that the previous model – trained *without* the "Abdomen move" labels – misclassifies these behavioral bouts as "Back grooming", "Walk" and "Still" (Fig. 5). Therefore, the addition of this new behavior class refines the previous segmentation.

4. Discussion

We presented an approach to semi-supervised behavioral segmentation that improves upon fully supervised and fully unsupervised approaches. We demonstrated that supervised segmentation metrics (F1) can be improved through the addition of a weakly supervised loss that classifies heuristic labels, as well as a self-supervised loss that predicts the evolution of the behavioral features that serve as model input (Fig. 2). Our work can also be viewed as adding a small number of labels to an unsupervised segmentation problem [10, 25], which we show increases the precision of downstream clustering (thus ensuring the model captures known behaviors of interest) while still allowing the model to discover novel behavioral phenomena (Fig. 4).

This semi-supervised approach can also serve as the foundation for an efficient active learning strategy that reduces human annotation overhead. Computing heuristic labels is a simple strategy to quickly label many frames. A model trained with the weak and self-supervised losses provides an initial embedding that can then guide the selection of frames to label. Additional behavior classes, if discovered, can be added to the set of heuristic and hand labels. This procedure can then be iterated. We demonstrated one aspect of this active learning approach, and believe this is a fruitful direction for future exploration.

References

[1] David J Anderson and Pietro Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014. 1

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 3

[3] Eleanor Batty, Matthew Whiteway, Shreya Saxena, Dan Biderman, Taiga Abe, Simon Musall, Winthrop Gillis, Jeffrey Markowitz, Anne Churchland, John P Cunningham, et al. Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. *Advances in Neural Information Processing Systems*, 32:15706–15717, 2019. 2, 5

[4] Gordon J Berman, Daniel M Choi, William Bialek, and Joshua W Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014. 1, 2, 5

[5] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. 2

[6] James P Bohnslav, Nivanthika K Wimalasena, Kelsey J Clausing, David Yarmolinsky, Tomas Cruz, Eugenia Chiappe, Lauren L Orefice, Clifford J Woolf, and Christopher D Harvey. Deepethogram: a machine learning pipeline for supervised behavior classification from raw pixels. *bioRxiv*, 2020. 1, 2, 9

[7] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 2

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 3

[9] Sandeep Robert Datta, David J Anderson, Kristin Branson, Pietro Perona, and Andrew Leifer. Computational neuroethology: a call to action. *Neuron*, 104(1):11–24, 2019. 1, 2, 5

[10] Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. *arXiv preprint arXiv:1611.00094*, 2016. 2, 6

[11] Alex Gomez-Marin, Joseph J Paton, Adam R Kampff, Rui M Costa, and Zachary F Mainen. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature neuroscience*, 17(11):1455–1462, 2014. 1

[12] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994, 2019. 2

[13] Pranay Gupta, Anirudh Thatipelli, Aditya Aggarwal, Shubh Maheshwari, Neel Trivedi, Sourav Das, and Ravi Kiran Sarvadevabhatla. Quo vadis, skeleton action recognition? *arXiv preprint arXiv:2007.02072*, 2020. 2

[14] Alexander I Hsu and Eric A Yttri. B-soid: an open source unsupervised algorithm for discovery of spontaneous behaviors. *bioRxiv*, page 770271, 2020. 1

[15] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016. 2

[16] Hueihan Jhuang, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D Steele, and Thomas Serre. Automated home-cage behavioural phenotyping of mice. *Nature communications*, 1(1):1–10, 2010. 1

[17] Matthew J Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Composing graphical models with neural networks for structured representations and fast inference. *arXiv preprint arXiv:1603.06277*, 2016. 1, 2

[18] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10(1):64, 2013. 1

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[20] Gregory Kramida, Yiannis Aloimonos, Chethan Mysore Parameshwara, Cornelia Fermüller, Nikolas Alejandro Francis, and Patrick Kanold. Automated mouse behavior recognition using vgg features and lstm networks. In *Proc. Vis. Observ. Anal. Vertebrate Insect Behav. Workshop (VAIB)*, pages 1–3, 2016. 1

[21] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 2

[22] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017. 5

[23] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54. Springer, 2016. 5

[24] Jingyuan Li and Eli Shlizerman. Iterate & cluster: Iterative semi-supervised action recognition. *arXiv preprint arXiv:2006.06911*, 2020. 2

[25] Jingyuan Li and Eli Shlizerman. Sparse semi-supervised action recognition with active learning. *arXiv preprint arXiv:2012.01740*, 2020. 2, 6

[26] Kevin Luxem, Falko Fuhrmann, Johannes Kürsch, Stefan Remy, and Pavol Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *bioRxiv*, 2020. 1, 2, 5

[27] Jesse D Marshall, Diego E Aldarondo, Timothy W Dunn, William L Wang, Gordon J Berman, and Bence P Ölveczky. Continuous whole-body 3d kinematic recordings across the rodent behavioral repertoire. *Neuron*, 109(3):420–437, 2021. 2, 5

[28] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018. 2

[29] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11, 2020. 2

[30] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2, 5

[31] Kartikeya Murari et al. Recurrent 3d convolutional network for rodent behavior recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1174–1178. IEEE, 2019. 1

[32] Ngoc G Nguyen, Dau Phan, Favorisen R Lumbanraja, Mohammad Reza Faisal, Bahriddin Abapihi, Bedy Purnama, Mera K Delimayanti, Kunti R Mahmudah, Mamoru Kubo, and Kenji Satou. Applying deep learning models to mouse behavior recognition. *Journal of Biomedical Science and Engineering*, 12(2):183–196, 2019. 1

[33] Simon RO Nilsson, Nastacia L Goodwin, Jia J Choong, Sophia Hwang, Hayden R Wright, Zane Norville, Xiaoyu Tong, Dayu Lin, Brandon S Bentzley, Neir Eshel, et al. Simple behavioral analysis (simba): an open source toolkit for computer classification of complex social behaviors in experimental animals. *BioRxiv*, 2020. 1

[34] Talmo D Pereira, Diego E Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S-H Wang, Mala Murthy, and Joshua W Shaevitz. Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1):117–125, 2019. 2

[35] Talmo D Pereira, Joshua W Shaevitz, and Mala Murthy. Quantifying behavior to understand the brain. *Nature neuroscience*, pages 1–13, 2020. 1, 5

[36] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 2

[37] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007. 5

[38] Evan Schaffer, Neeli Mishra, Wenze Li, Matthew Whiteway, Jason Freedman, Kripa Patel, Venkatakaushik Voleti, Liam Paninski, Larry Abbott, Elizabeth Hillman, and Richard Axel. Flygenvectors: large-scale dynamics of internal and behavioral states in a small animal. In *Cosyne*, 2020. 3

[39] Cristina Segalin, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J Sun, Pietro Perona, David J Anderson, and Ann Kennedy. The mouse action recognition system (mars): a software pipeline for automated analysis of social behaviors in mice. *bioRxiv*, 2020. 1

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 2

[41] Oliver Sturman, Lukas Matthias von Ziegler, Christa Schälppi, Furkan Akyol, Benjamin Friedrich Grewe, and Johannes Bohacek. Deep learning based behavioral analysis

enables high precision rodent tracking and is capable of outperforming commercial solutions. *bioRxiv*, 2020. 1

[42] Jennifer J Sun, Ann Kennedy, Eric Zhan, Yisong Yue, and Pietro Perona. Task programming: Learning data efficient behavior representations. *arXiv preprint arXiv:2011.13917*, 2020. 2

[43] Elsbeth A van Dam, Lucas PJJ Noldus, and Marcel AJ van Gerven. Deep learning improves automated rodent behavior recognition within a specific experimental setup. *Journal of neuroscience methods*, 332:108536, 2020. 1

[44] Lukas von Ziegler, Oliver Sturman, and Johannes Bohacek. Big behavior: challenges and opportunities in a new era of deep behavior profiling. *Neuropsychopharmacology*, 46(1):33–44, 2021. 1

[45] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013. 2

[46] Matthew R Whiteway, Dan Biderman, Yoni Friedman, Mario Dipoppa, E Kelly Buchanan, Anqi Wu, John Zhou, Jean-Paul R Noel, John P Cunningham, Liam Paninski, et al. Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. *bioRxiv*, 2021. 5

[47] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015. 1, 2

[48] Anqi Wu, E Kelly Buchanan, Matthew Whiteway, Michael Schartner, Guido Meijer, Jean-Paul Noel, Erica Rodriguez, Claire Everett, Amy Norovich, Evan Schaffer, et al. Deep graph pose: a semi-supervised deep graphical model for improved animal pose tracking. *bioRxiv*, 2020. 2, 3

Supplemental Figures and Tables

	Experiment ID	Total frames	Walk	Still	Front groom	Back groom	Abdomen move
Train	2019_08_07_fly2	50000	300	300	100	300	591
	2019_08_08_fly1	94960	300	300	300	150	594
	2019_08_20_fly2	65108	300	300	300	300	1067
	2019_10_10_fly3	141042	300	300	300	300	478
	2019_10_14_fly3	140929	300	300	300	300	318
Test	*2019_06_26_fly2	45000	300	300	300	300	706
	2019_08_14_fly1	124144	300	300	300	300	693
	2019_08_20_fly3	73055	300	300	300	300	405
	2019_10_14_fly2	139925	300	300	300	300	101
	2019_10_21_fly1	142554	300	300	300	300	0
	Totals	1016717	3000	3000	2800	2850	4953

Table S1. The number of labeled frames per behavior for each fly. Initial behaviors ("Walk", "Still", "Front groom", "Back groom") were labeled in chunks of 50 contiguous time points. Video frame rate is 70 Hz. Flies often engage in behaviors for longer than 50 frames, so the selected 50-frame chunks did not contain any transitions from one behavior to another. The "Abdomen move" behavior was added during a second round of labeling. We labeled longer contiguous chunks in order to capture the full range of the behavior during each bout, which usually consists of raising the abdomen, a brief hold, and then lowering the abdomen. Labeling was performed using the DeepEthogram GUI [6]. The asterisk (*) denotes the test experiment visualized in Figs. 3, 4, S1 and S3.

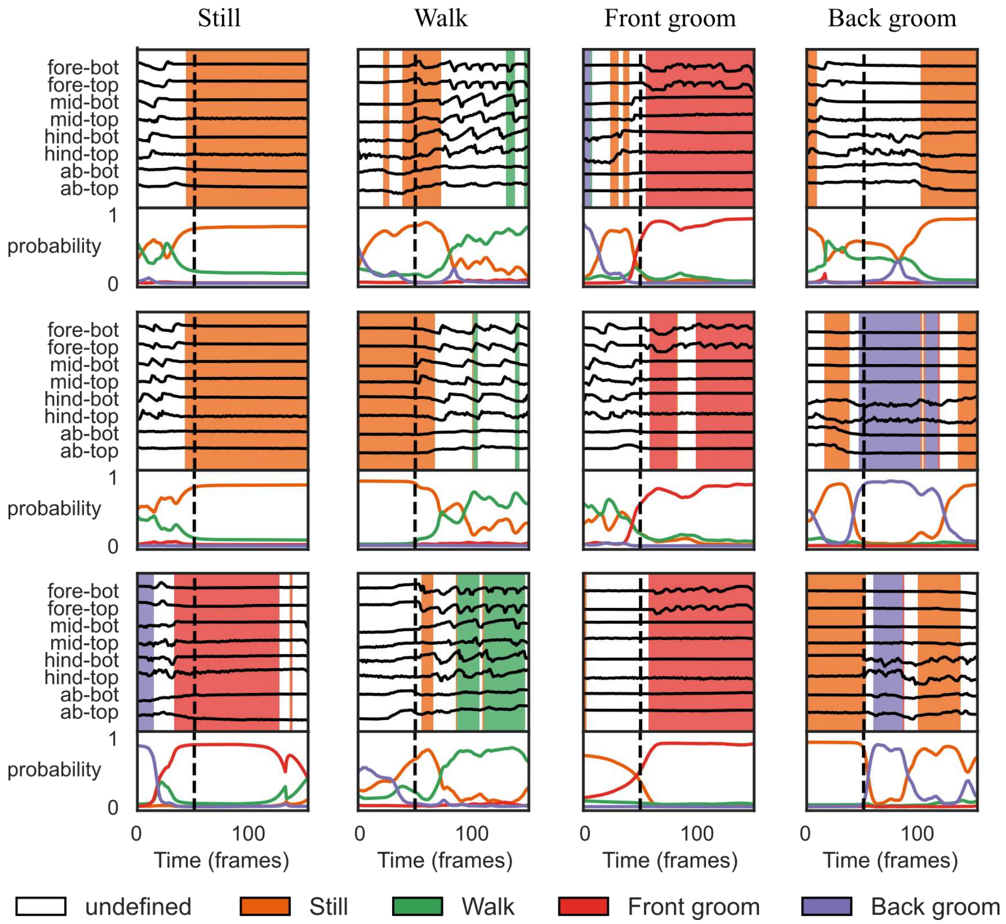

Figure S1. The absence of the weak and self-supervised losses leads to more errors in the segmentation. Panels show the outputs of the fully supervised sequence model (GRU with $\lambda_h = \lambda_p = 0$) for the same sample time segments as in Fig. 3. Note that for some bouts (e.g. those in the "Walk" column) the highest probability belongs to the correct behavior class, but the model is less confident. For other bouts, the model is confident but incorrect (e.g. the final "Still" bout is misclassified as "Front groom").

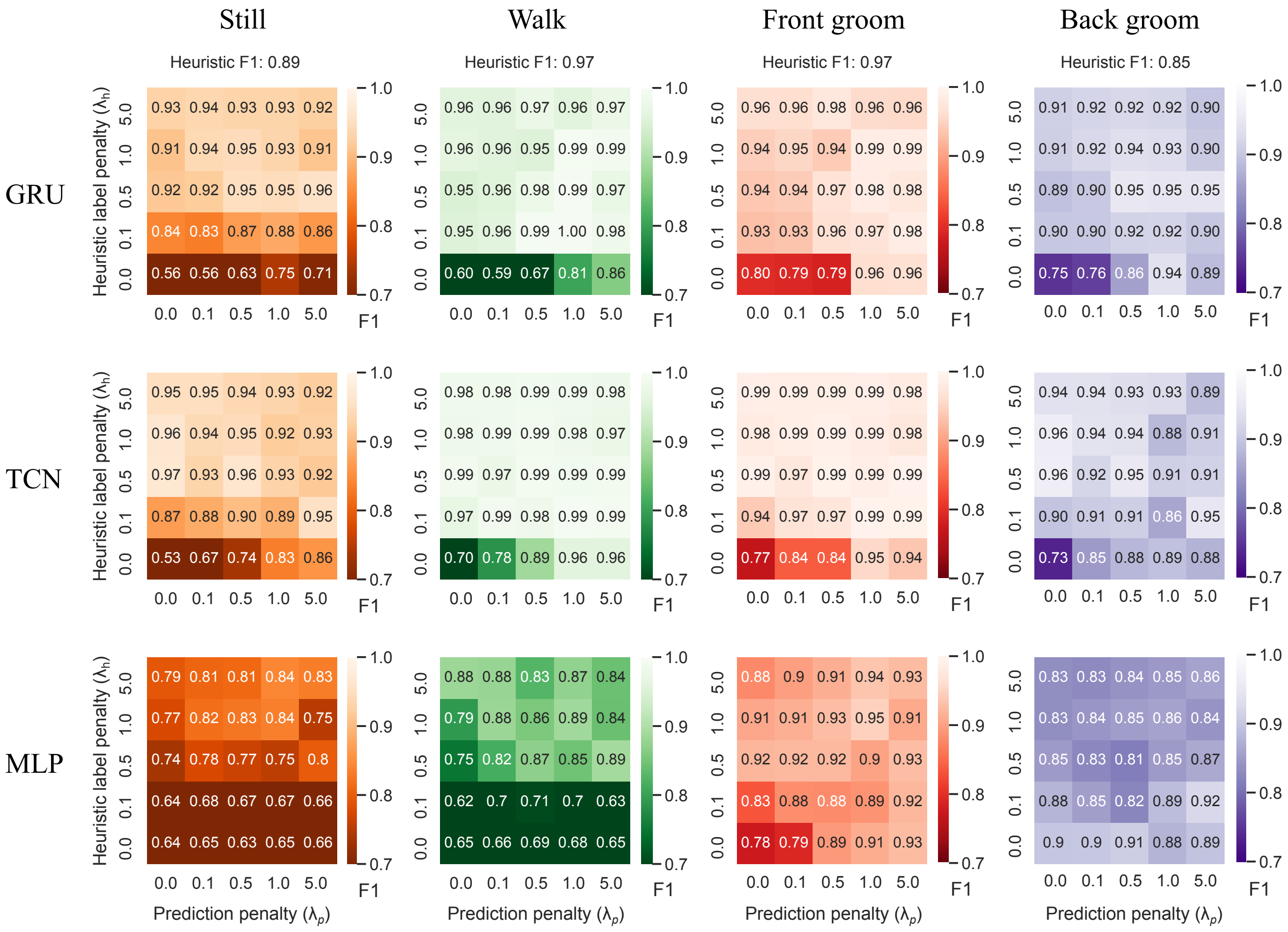

Figure S2. The performance improvement in F1 scores due to weak and self-supervised losses is not architecture dependent. *Top*: F1 results from a GRU where both the encoder and decoder are modeled with two layer bidirectional GRU networks with 32 cells per layer (same models as Fig. 2). *Middle*: F1 results from a temporal convolutional network (TCN) where both the encoder and decoder are modeled with two layers of 1D temporal convolutions (filter size of 17), each followed by a temporal downsampling (encoder) or upsampling (decoder) step. *Bottom*: F1 results from an MLP network where the first layer of the encoder is a 1D temporal convolution (filter size of 17) and the second layer is fully connected (32 hidden units). The decoder is modeled as a two layer MLP (no convolutions) with 32 hidden units per layer. For each architecture we performed a small hyperparameter search across number of layers, cells/units per layer, learning rate, and temporal filter sizes; we found that our results are robust across different settings (data not shown).

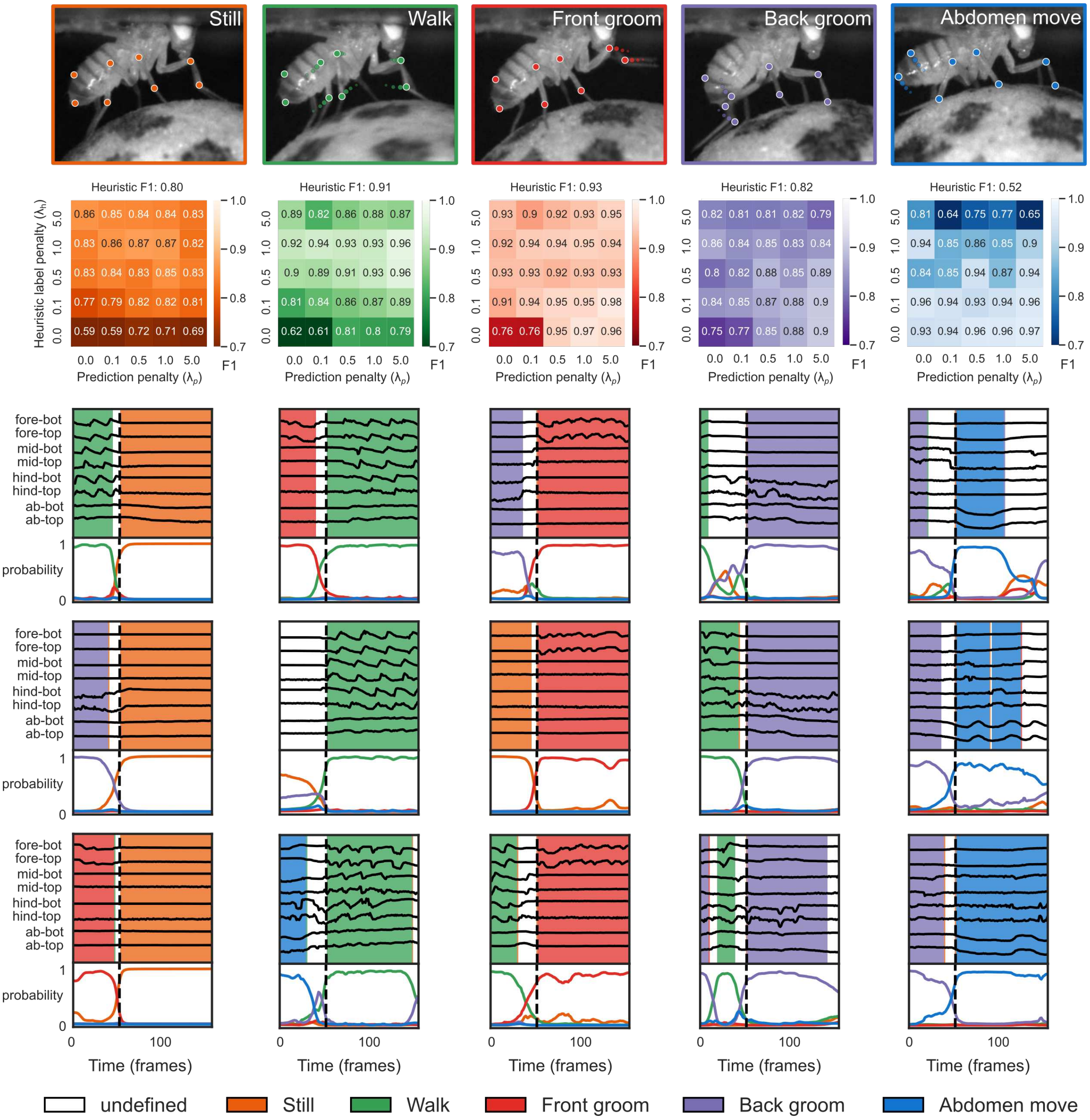

Figure S3. Sequence models can be fit to new behavior classes. Each column represents model fits for a single behavior class. *Top row*: Example frames from five fly behavior classes, overlaid with pose estimates. *Second row*: F1 scores for each behavior on test data. *Bottom rows*: Model outputs for sample time segments. Same conventions as Fig. 3. These probabilities correspond to the GRU model with $\lambda_h = 0.5$, $\lambda_p = 5.0$, which achieved the highest F1 score averaged over all behaviors on test data.