Pierce N Mullen[1], Beatrice Bowlby[1], Holly C Armstrong[1], and Maarten F Zwart[1]

[1]School of Psychology and Neuroscience, Centre for Biophotonics, University of St Andrews, St Andrews, KY16 9JP, United Kingdom

PoseR - A deep learning toolbox for decoding animal behavior

Abstract

The actions of animals provide a window into how their minds work. Recent advances in deep learning are providing powerful approaches to recognize patterns of animal movement from video recordings, including markerless pose estimation models. However, tools to efficiently parse coordinates of animal position and pose into meaningful semantic behavioral labels are lacking. Here, we present PoseRecognition (PoseR), a behavioral decoder leveraging state-of-the-art action recognition models using spatio-temporal graph convolutional networks. We show that it can be used to decode animal behavior quickly and accurately from pose estimations, using zebrafish larvae and mice as model organisms. PoseR can be accessed using a Napari plugin, which facilitates efficient behavioral extraction, annotation, model training and deployment. We have simplified the workflow of behavioral analysis after pose estimation, transforming coordinates of animal position and pose into meaningful semantic behavioral labels, using methods designed for fast and accurate behavioral extraction, annotation, model training and deployment. Furthermore, we contribute a novel method for unsupervised clustering of behaviors and provide open-source access to our zebrafish datasets and models. The design of our tool ensures scalability and versatility for use across multiple species and contexts, improving the efficiency of behavioral analysis across fields.

Introduction

Decoding animal behavior from video recordings allows us to understand its neural underpinnings. Identifying where an animal is, i.e., its position and pose, during video

recordings has largely been solved by advances in deep learning, but recognizing animal movements or sequences of poses as meaningful behavior remains a more difficult problem for neuroscientists (Mathis et al., 2018; Pereira et al., 2022). Existing analysis workflows fall into two categories: unsupervised behavioral discovery or supervised behavioral classification. Unsupervised discovery leverages machine learning models to identify distinct actions of animals from either raw videos or pose estimations requiring no prior knowledge (Berman et al., 2014; Hsu & Yttri, 2021; Weinreb et al., 2023; Wiltschko et al., 2015, 2020). This approach is useful for researchers looking to discover new behaviors but requires thorough post-hoc analysis and is sensitive to pre-processing of the data. Alternatively, supervised classification applies prior knowledge of the behaviors that a researcher would like extract from video recordings and teaches them to machine and deep learning models to improve the efficiency of analysis (Bohnslav et al., 2021; Dankert et al., 2009; Kabra et al., 2013; Ro et al., 2020; Segalin et al., 2021; Sturman et al., 2020). These previous approaches have typically focused on specific contexts of behavior of mice or fruit flies; more general solutions to classifying animal behavior across contexts, backgrounds and species should be pursued.

To this end, we incorporated skeleton-based action recognition deep learning architectures that are state-of-the art in human action recognition to the classification of animal behavior (Yan et al., 2018). These architectures use graph neural networks to learn both the spatial and temporal components of pose estimations, treating them as nodes of a graph upon which convolution can be applied. Graph neural networks are revolutionizing our ability to model complex relationships between connected components with breakthroughs in solving the protein folding problem, recommendation systems, and drug discovery (Hamilton et al., 2017; Jiménez-Luna et al., 2020; Jumper et al., 2021). A graph consists of nodes and edges where edges describe the relationship between nodes. Nodes can contain features, for instance x, y coordinates and confidence interval of a pose estimation. Mathematically, the pose graph is described as $G = (V, E)$, where V represents the body part nodes and E the edges corresponding to the anatomical relationships between body parts. Convolutional operations

on graphs involve aggregating feature information for each node from neighbor nodes to produce a new feature representation of the pose graph where knowledge of the state of nearby nodes contributes to the state of each node. For every time point in a behavior, the pose graph can be transformed in this way and the temporal features of these aggregated nodes can then be learned to classify behaviors.

Our strategy to develop a new behavioral decoder based on these models was to simplify and accelerate three main steps in the behavioral analysis pipeline: 1) extraction, 2) annotation and finally 3) decoding of behavior. We first applied signal processing methods to identify windows in which behavior is occurring, enabling rapid annotation of thousands of behaviors. The coordinates of the points on an animal body extracted using pose estimation by DeepLabCut (Mathis et al., 2018) are then used in space *and* time in spatial-temporal graph convolutional networks. We included easy-to-use functions and released PoseR as an open-source plugin for the popular multi-dimensional data viewer Napari (*Napari: A Multi-Dimensional Image Viewer for Python | Zenodo*, n.d.; Sofroniew et al., 2022) to make training and deploying these deep learning models more accessible to a wider audience, and for the performance and visualization benefits it offers.

We used zebrafish larvae to develop an efficient behavioral analysis workflow and demonstrate its general use when applied to other species and when including environmental context. Zebrafish exhibit a large repertoire of behaviors encompassing environmental exploration, escape, and predation (Kalueff et al., 2013). It is possible to elicit these behaviors in an experimental environment with visual projection of choice stimuli such as the looming shadow of a predator, the random walk of small prey or ebb and flow in a natural scene of riverbed (Johnson et al., 2020; Marques, Lackner, Felix, et al., 2018). We acquired high speed videos (330 fps) of freely swimming zebrafish larvae under these conditions and tested the effectiveness of our end-to-end behavioral analysis toolbox, PoseR in two scenarios: generating and decoding a small manually curated dataset and generating and decoding a large novel unsupervised-generated dataset of zebrafish behaviors. We also tested PoseR on

a mouse open field dataset (Sturman et al., 2020) and a pre-clustered zebrafish dataset containing tail angles (Marques, Lackner, Félix, et al., 2018) and demonstrate the applicability of this approach to the analysis of other species, context-dependent behaviors, and data without pose estimations.

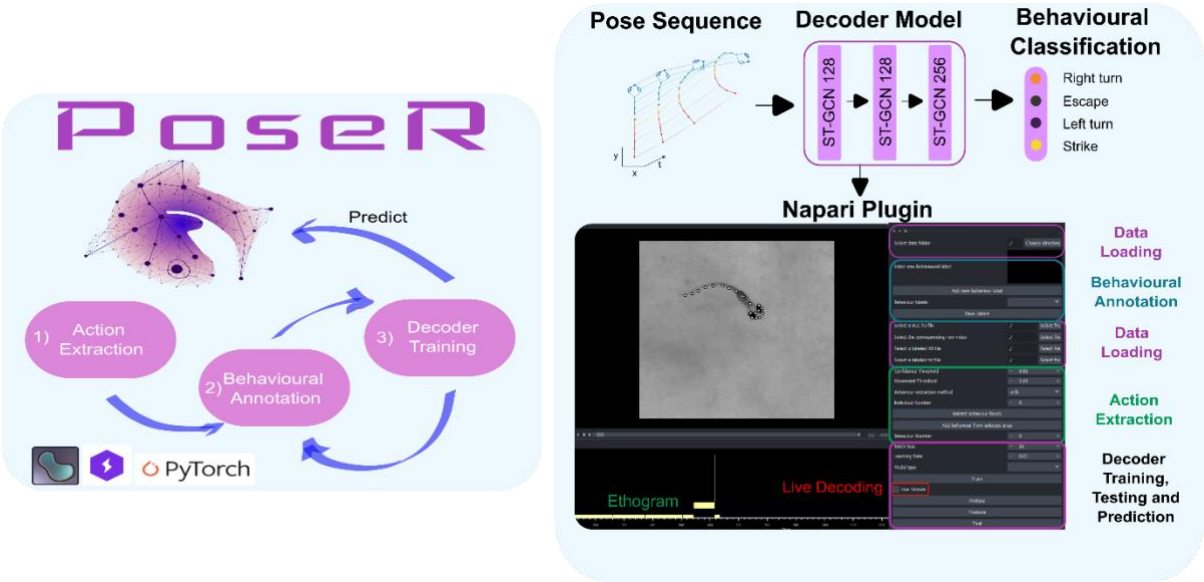


Figure 1. PoseR – A deep learning behavioral classification toolbox. A napari plugin designed to accelerate extraction, annotation, and training deep learning classifiers to predict animal behaviors. It combines popular deep learning packages Pytorch (Paszke et al., 2019) and Pytorch Lightning (*Lightning-AI/Lightning: Deep Learning Framework to Train, Deploy, and Ship AI Products Lightning Fast.*, n.d.) to simplify the process of training deep learning behavioral decoders for a wide range of applications.

Methods

Behavioral setup. All procedures were carried out according to the UK Animals (Scientific Procedures) Act 1986 and approved by the UK Home Office. Zebrafish larvae (4-7 days post fertilization (dpf)) were placed in an acrylic recording chamber (25 x 25 x 25 mm) containing system water. Visual projections were displayed onto diffuse acrylic beneath the recording chamber using a cold mirror (Edmund Optics, 45° AOI, 50.0mm Square, Cold Mirror, #64 451) and a projector (Epson EF-11 3LCD, #0011131458). The zebrafish larvae were illuminated using a custom infra-red LED (850nm) array beneath the chamber and recorded at 330 frames

per second using a Mikrotron camera (MC1362) and a high-speed frame grabber (National Instruments, PCIe-1433). Images were acquired and dynamically cropped in Bonsai (Lopes et al., 2015) and zebrafish larval positions were extracted using background subtraction and thresholding. This allowed for closed-loop presentation of stimuli based on the position and orientation of the larvae using the BonZeb package (Guilbeault et al., 2021) and reduced file size to permit continuous recording to disk of long-duration videos. In some experiments, live low-saline rotifers were added to the imaging chamber to record zebrafish larval swims in the presence of prey.

Pose estimation. A ResNet50 neural network was trained using DeepLabCut (Mathis et al., 2018) to estimate the position of 19 points on the zebrafish body. Each eye was represented by 4 points and the remaining 11 points were positioned at equal intervals along the zebrafish midline from nose to tail fin (see Fig. 1). The neural network was trained and videos were analyzed using a Tesla V100 Nvidia GPU at the Kennedy High Performance Computing Cluster, St Andrews.

Swim bout extraction

Due to the discrete bout-like nature of zebrafish swimming behavior it was relatively straightforward to define and extract periods in which behavior was occurring. Leveraging the lateral movement of the tail during swimming, the side-to-side motion of each body part was calculated, and a peak finding algorithm (Virtanen et al., 2020) was used to identify peak lateral movement and define the start and end of that swim bout. The side-to-side motion was calculated by first subtracting all coordinates by a center node (node 13) to get an egocentric representation of the zebrafish larval pose. The Euclidean trajectory and orthogonal trajectory for each node for each frame was calculated and future trajectories were projected onto the preceding orthogonal axis by dot product. This quantified the degree of perpendicular (side-to-side) motion of each node relative to the nodes' previous position. The median side-to-side motion was smoothed with a gaussian filter (width = frames per seconds / 10). This representation of lateral motion was then thresholded using a median absolute deviation of 2

to extract peaks and windows around putative swim bouts using the Scipy find peaks function. Post-processing of these windows ensured swim bouts did not overlap and that the pose estimation confidence scores during that window were greater than 0.8.

Manual labelling of swim bouts

All videos and DeepLabCut pose estimation were loaded into the PoseR napari plugin to facilitate easy swim bout extraction and manual labelling of behaviors. Swim bouts were extracted within the plugin as described above and video clips of those swim bouts could be cycled through in PoseR with pose estimations overlayed. Labels are assigned in the plugin from a drop-down menu and saved to a h5 file format to store the individual id, pose estimation, swim bout number, behavioral label and confidence scores. For initial validation, swim bouts were labelled as either left, right or forward swims, resulting in a manually classified dataset of 4368 swim bouts.

Swim bout unsupervised clustering

34015 swim bout pose estimations were extracted from behavioral recordings, resulting in an N x C x T x V array, where N is the number of swim bouts, C is the number of channels (X coordinate, Y coordinate and confidence interval of estimation), T is the number of timepoints in the swim bout, and V is the number of nodes on the zebrafish larvae body that were estimated. Swim bouts were aligned to the vertical axis ensuring all larvae were orientated facing north and modified to an egocentric coordinate system by subtracting the coordinates of a central node. The change in angle from the central node of each node during the swim bouts was computed, resulting in an N x T x V array containing angle changes. The dimensionality of this array was reduced using tensor decomposition (Kolda & Bader, 2009; *Neurostatslab/Tensortools: A Very Simple and Barebones Tensor Decomposition Library for CP Decomposition a.k.a. PARAFAC a.k.a. TCA*, n.d.; Williams et al., 2018), where the data is approximated by a model consisting of a sum of components, with each component described by the outer product of three rank-1 tensors in the swim bout (N), time (T) and node (V)

direction. This decomposition results in three matrices; a swim bout (N) x components factor matrix, a time (T) x components factor matrix and a node (V) x components matrix. The swim bout factor matrix contains a description of each swim bout according to the 10 tensor components. We used 10 components as this resulted in a low reconstruction error of 0.17, where the sum of components approximated the original dataset with an accuracy of 83%, whilst retaining stability in multiple replicates. Hierarchical agglomerative clustering (Scikit-Learn) was subsequently performed on this matrix where 30 distinct swim bout types resulted in a relatively low Davies-Bouldin score (Davies & Bouldin, 1979) and high silhouette score (Rousseeuw, 1987) in cluster evaluation.

Spatial temporal graph convolutional network. A spatial temporal graph convolutional network (ST-GCN)(*GitHub - Open-Mmlab/Mmskeleton: A OpenMMLAB Toolbox for Human Pose Estimation, Skeleton-Based Action Recognition, and Action Synthesis.*, n.d.; Yan et al., 2018) was modified into a Pytorch-Lightning module and the final architecture optimised using the Pytorch ecosystem across multiple Nvidia Tesla V100 GPUs (Kennedy HPC, St Andrews). The network was further modified to be shallower and wider consisting of three spatial temporal hidden layers of width 48, 256, 256, trained using a cross-entropy loss function and ADAM optimizer. All datasets were split into a training set (70 %), validation set (15 %) and testing set (15 %) using sklearn, and accuracy, precision, F1 score, and recall were calculated to evaluate the performance of each model.

Code and data availability

PoseR is installable via pypi https://pypi.org/project/PoseR-napari/ and hosted on GitHub at https://github.com/pnm4sfix/PoseR. All data is available at the Zenodo repository at https://doi.org/10.5281/zenodo.7807968.

Results

PoseR enables fast coding and decoding of a small set of behaviors.

We initially aimed to develop and validate our toolbox by rapidly coding a small subset of trivial zebrafish behaviors (left, right and forward swims) and designing an ST-GCN model capable of correctly decoding these. Whilst in practice decoding left vs right swims is easily solved using classic signal processing techniques, it provided an appropriate proof-of-concept to test our analysis pipeline and application of ST-GCNs to zebrafish behavior. Zebrafish larval poses, consisting of 19 coordinates on the larval head, trunk, and tail, were extracted from video recordings using a DeepLabCut ResNet50 model and ~4500 swim bouts were manually labelled in the plugin according to the direction of the swim bout by a trained observer. The graph in ST-GCN models is a spatial and temporal representation of the animal upon which graph convolution can be applied to represent complex interactions between nodes in a pose. To examine how the spatial graph representation of pose changes over time during the frames of a video recording, each node extends and connects to its corresponding node for each video frame through time (Figure 2A). These abstract spatial and temporal representations of an animal's pose can be learned using a spatial temporal graph convolutional network and the subsequently trained model can be used to classify behavior in an experimental setting (Yan et al., 2018). We trained an ST-GCN network consisting of three spatial-temporal graph convolution layers for 26 epochs using early stopping to prevent overfitting to training data (Fig 2A). Testing the model on the validation dataset resulted in a high average accuracy of 90 % across swim types (Fig 2B). We calculated a value of 0.9 for precision, recall and F1-score of the model's predictions versus ground truth. Precision quantifies the ratio of true positives to total positive predictions whereas recall quantifies the ratio of true positives to the total true positives and false negatives. The F1-score is the harmonic mean of precision (does the model detect all the true positives) and recall (does the model detect the positive cases and only the positive cases). Applying the model to unseen swim bouts and plotting the distribution of heading angle changes during the bouts revealed tight distributions in the appropriate direction for left and right swims (Fig 2C). No trajectory information was included with our dataset, which relied only on egocentric coordinates; however, our model was able to accurately classify forwards swims and resulted in a heading angle change distribution

centered on zero degrees. This distribution was symmetric but wider and bimodal, suggesting sub-groups of forward swims. This initial left-right ST-GCN model provided a promising initial validation for our toolbox in predicting a small subset of manually labelled behaviors. We took this approach further to test the limits of the toolbox and produce a model capable of accurately classifying a wider, more diverse, and challenging range of zebrafish behaviors.

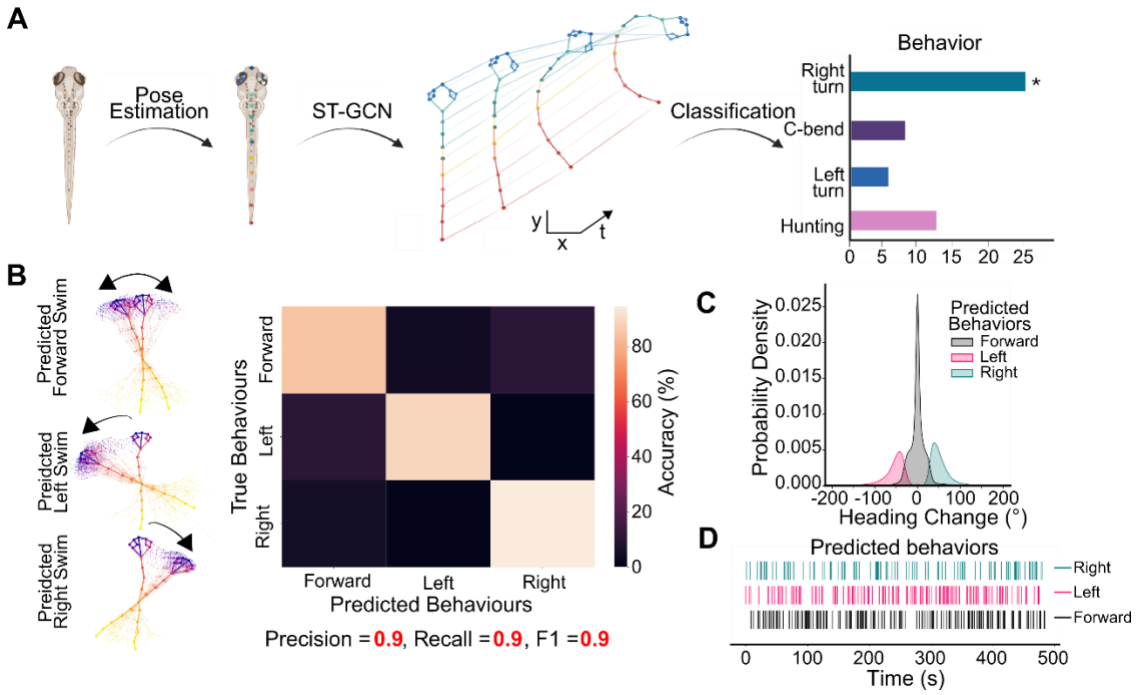


Figure 2. Deep Learning Behavior Decoding Strategy. A) Zebrafish larval poses were extracted from videos using DeepLabCut. Sequences of poses can be represented as graphs with edges connecting body parts in space and time. **B)** These sequences were used to train a spatial-temporal graph convolutional neural network (ST-GCN) to classify swim bouts manually labelled as either left, right or forwards swims. Examples of correctly classified swim bouts are shown. **C)** Heading changes of unseen swim bouts classified by the trained left-right ST-GCN model. **D)** An example ethogram showing classified left, right, forward swims during a recording session.

Generating a comprehensive zebrafish behavior dataset using tensor decomposition and agglomerative clustering

We next generated a larger dataset of zebrafish larval behaviors. We recorded long duration (40 minutes) high frame rate (330 fps) videos of zebrafish larvae behaving and responding to a wide range of closed-loop visual stimuli (Guilbeault et al., 2021). These stimuli were chosen to elicit natural behavior such as escape reflexes, phototaxis and optomotor responses, resulting in a dataset of ~30,000 swim bouts at high temporal resolution. We took a novel approach to clustering these swim bouts by first using tensor decomposition (Kolda & Bader, 2009; Williams et al., 2018) to reduce the complexity of the dataset to 10 tensor components with each component containing a swim bout factor, body part factor and time factor (Fig. 3A). This resulted in a matrix describing the contribution of each swim bout according to the 10 tensor components, to which hierarchical agglomerative clustering could then be applied (Fig. 3B). This approach resulted in 30 distinct behaviors, which contained swim bouts that were homogenous within each cluster and showed similarity to swim bout types that have been previously described (Fig S1). Symmetrical tail beats in clusters 2, 14 and 23 represented forward swims with increasing power from slow scoots to forward bursts. Broadly, clusters appeared to be initially separable by changes in heading direction with broad categories assigned left, right, forward, and large angle turns and mapped to a low dimension behavioral projection of each swim using uniform manifold approximation and projection (McInnes et al., 2018)(Fig 3C, S1A). Within these classes, the temporal dynamics varied depending on the vigor, amplitude, and number of tail oscillations and whether changes in directions occurred early or late within the swim bout. Clusters 9, 11, 15, 18, 22 and 27 appeared to show similarity with routine-turns described in the literature involving a change in orientation of about 40 degrees with no scoot (Fero et al., 2011). Burst-like swims were identified in clusters 1, 4, 6, 7, 12, 13, 23 and 28, where sustained large-amplitude cyclical tail oscillations were observed. Putative O-bend swims could be mapped to clusters 20 and 30 where an almost complete inversion of heading direction occurred with little evidence of large tail beats after the heading change. J-turn-like orientating swims showed similarity to cluster 3, 5 and 29 where the heading change is accompanied by small amplitude tail beats. Large amplitude escape-like swims were identifiable in clusters 16 and 24 where fish rapidly changed and swam with vigor

in almost the opposite direction resembling slow-latency C-bend swim (SLC)(Fero et al., 2011). Long-latency C-bends (LLC), where the heading change and swim vigor were less extreme than SLC swims were seen in clusters 8, 10, 17, 19. Quantifying the presence of each swim type by visual stimuli highlighted the expected preference for left turns and right turns during the presentation of R-L and L-R optomotor gratings, respectively, as larvae aligned themselves with the direction of visual flow. Increases in swim activity across types were seen during the presentation of visual prey, with increases in forward swims when the prey was presented ahead of the larvae. Large amplitude swims were most prevalent in phototaxis, optomotor and prey presentation. In the presence of live rotifers, swim types 1, 3, 4, 5, 12, 14, and 23 dominated, representing a combination of burst swims and J-turns prevalent in zebrafish predation (Fig 3D, (Borla et al., 2002; Budick & O'Malley, 2000; Marques, Lackner, Félix, et al., 2018; Patterson et al., 2013)). Using this novel approach, we succeeded in extracting and generating a large, complex dataset of a range of swim bout behaviors that zebrafish larvae employ during different visual contexts.

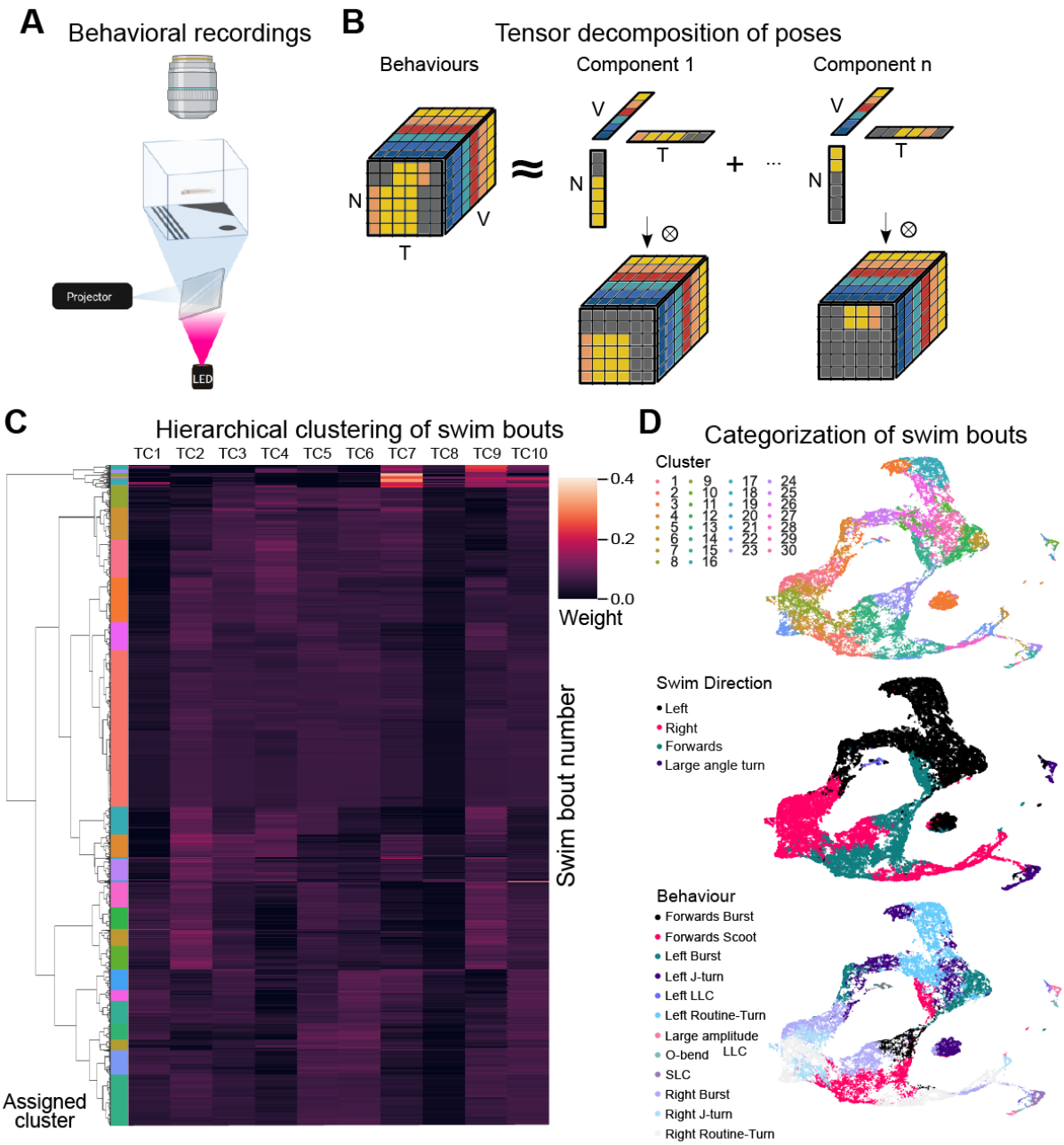


Figure 3. Clustering decomposed swim bouts. A) Tensor decomposition (PARAFAC/CANDECOMP) was applied to a 3d (rank-3 tensor) dataset containing egocentric node positions (in radians) of swim bouts, where the N dimension represented the swim bout number, the T dimension represented time and the V dimension represented the node/body part of the zebrafish. **B)** Agglomerative hierarchical clustering heatmap of the N x TC factor matrix showing how swim bouts could be grouped into similar clusters. **C)** A uniform manifold approximation and projection of the N x TC factor matrix color-coded by swim type cluster and broad swim class (left, right, forward and large angles swim).

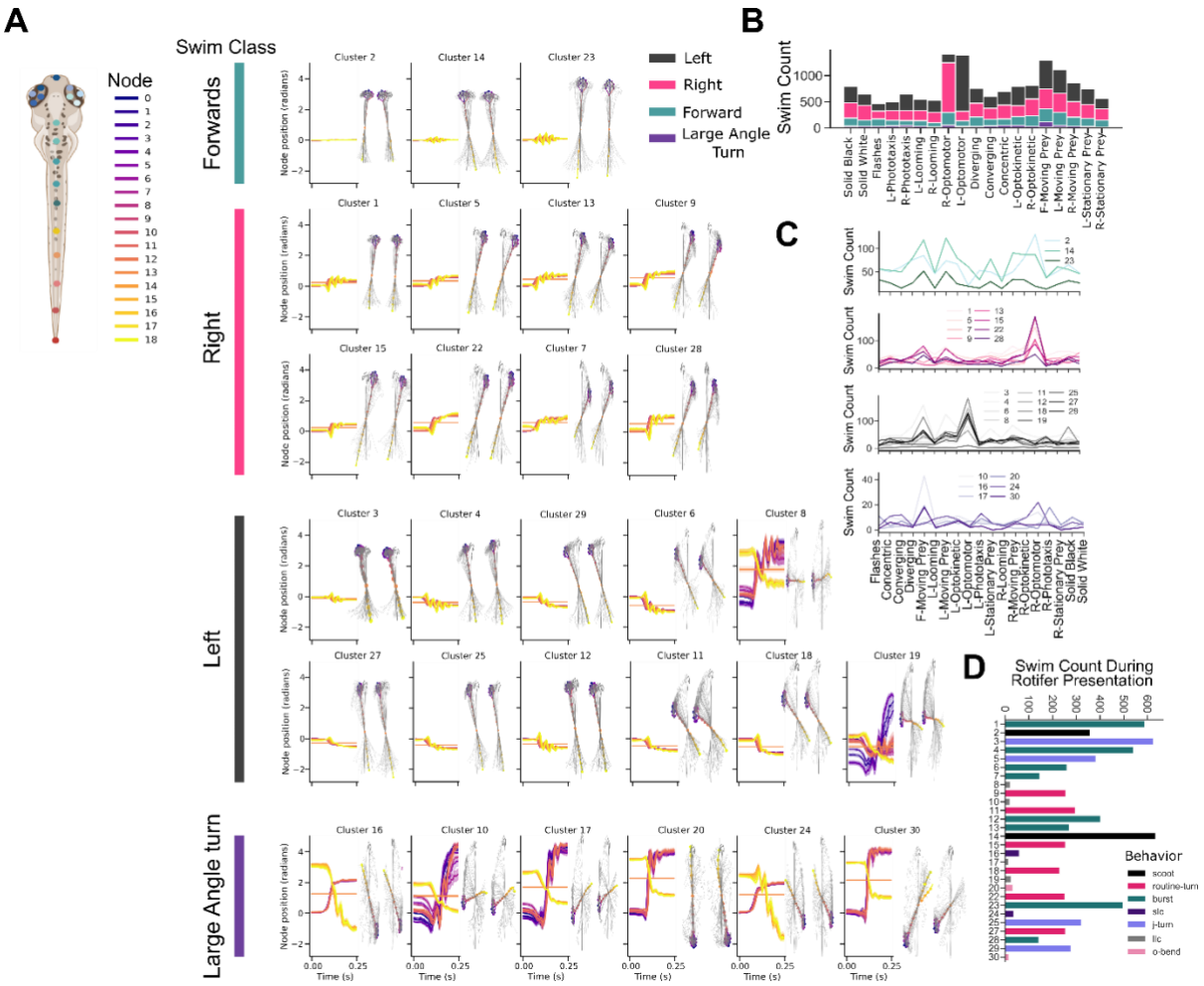
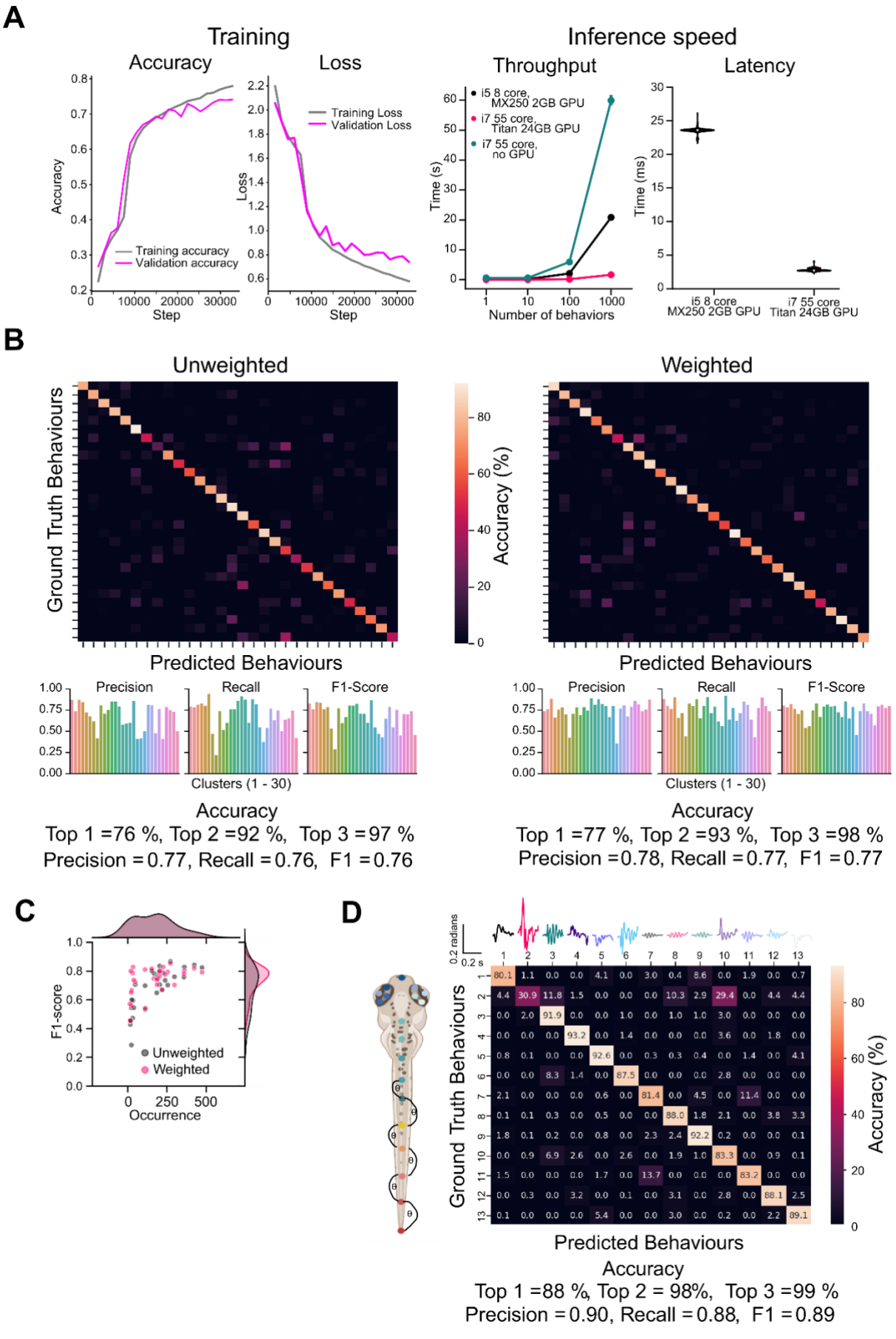


Figure S1. Swim dynamic and cluster examples. A) A schematic displaying the location of key point nodes along the zebrafish larval body. Each cluster (excluding 21 and 26 due to pose estimation noise) is represented by the average position of each color-coded node in radians during swimming. Error bands are shown as standard error of the mean position. **B)** A stacked bar chart quantifying numbers of each broad class (left, right, forwards and large amplitude swim type) according to the presentation of visual stimuli. **C)** Parallel plots showing the number of each swim type cluster across visual stimuli. **D)** Swim counts of each swim cluster during presence of live rotifers.

PoseR can be used to rapidly decode many complex behaviors

We next used PoseR to train an ST-GCN to recognize the more complex clustered swim behaviors with the aim of developing a universal zebrafish decoder that could be applied across a range of experiments. We found a high top-1 accuracy of 76 % and top-3 accuracy of 97 % for correctly classifying swim bout types in the test dataset, achieved on a first run

using PoseR's built-in helper functions to optimize initial model hyperparameters. Model accuracy increased and loss decreased quickly during training with a batch size of 16, cross entropy loss function and ADAM optimizer before stopping early when the loss plateaued (Fig 4A). The accuracy of the model's initial predictions was evident by the bright band along the diagonal axis of the confusion matrix and a precision, recall and f1-score of 0.77, 0.76, 0.76, respectively (Fig 4A, B). In addition to fast optimized training, we endeavored to evaluate the speed at predicting and analyzing new behaviors on different systems accelerated by either a GPU or a CPU. As expected, we found faster inference speeds on GPU based systems compared to CPU, with decoding of 1000 behaviors taking approximately 20 seconds with a batch size of 10 on GPU systems. The latency to analyze one behavior on GPU systems was 23.5 ± 0.0004 ms (Fig 4B). The different frequencies with which swim types occur resulted in an imbalanced dataset where some swim types, in particular the large amplitude turns, were rarer. To address this issue, we included a weight calculation function in PoseR during training to estimate the best weights for each swim type derived from the occurrence of that swim type relative to the total dataset size. This produced a model with a higher accuracy of 77%, a more balanced precision, recall and F1-score for all swim types including less frequent swim types demonstrating the ability of PoseR to accurately predict behaviors in unbalanced datasets (Fig. 4B, C, D), Graphs are very flexible data structures and graph nodes can be assigned multiple types of data from x, y pose estimations to precalculated joint angles, or local video features. To demonstrate this flexibility, we trained an ST-GCN model on a large pre-clustered zebrafish dataset that contained only angle information of a zebrafish larval tail with no x, y pose estimations (Marques, Lackner, Félix, et al., 2018). This dataset contained 13 swim types recorded at 700 fps, and angle information for 8 nodes along the tail and tail nodes in the ST-GCN model were connected in a chain to represent the tail. Using PoseR, the trained ST-GCN model achieved top 1, 2, 3 accuracies of 88%, 98%, and 99% with an overall precision, recall and f1-score of 0.9, 0.88, 0.89, respectively (Fig 4 E), demonstrating the powerful application of this approach to decoding behavior from a variety of different data types.


A
Training
Accuracy
Loss
Training accuracy
Validation accuracy
Training Loss
Validation Loss
Step
Inference speed
Throughput
Latency
i5 8 core, MX250 2GB GPU
i7 55 core, Titan 24GB GPU
i7 55 core, no GPU
Time (s)
Number of behaviors
Time (ms)
i5 8 core MX250 2GB GPU
i7 55 core Titan 24GB GPU
B
Unweighted
Weighted
Ground Truth Behaviours
Predicted Behaviours
Accuracy (%)
Precision
Recall
F1-Score
Clusters (1 - 30)
Accuracy
Top 1 =76 %, Top 2 =92 %, Top 3 =97 %
Precision =0.77, Recall =0.76, F1 =0.76
Accuracy
Top 1 =77 %, Top 2 =93 %, Top 3 =98 %
Precision =0.78, Recall =0.77, F1 =0.77
C
F1-score
Occurrence
Unweighted
Weighted
D
0.2 radians
0.2 s
Ground Truth Behaviours
Predicted Behaviours
Accuracy (%)
Accuracy
Top 1 =88 %, Top 2 = 98%, Top 3 =99 %
Precision =0.90, Recall =0.88, F1 =0.89

314

Figure 4. Accurate decoding of zebrafish behavior using a trained ST-GCN model. A) Training and validation accuracy and loss reported during each training step. Latency for one sample and inference speed for batch sizes 1, 10, 100, 1000 on different systems: a CPU-only system, a system with 2GB MX250 GPU, and one with a 24GB Titan GPU. **B)** A confusion matrix highlighting the percent of correct predictions compared to ground truth behavioral labels for unweighted and weighted behavioral training data. **C)** Precision, recall and F1-score for unweighted and weighted trained models. **D)** A scatterplot of F1-score versus the occurrence of a behavioral label to show weighting labels improves the overall F1-score. E) An ST-GCN model and subsequent confusion matrix trained on the ZebRep dataset (Marques, Lackner, Félix, et al., 2018; Orger, 2018) where only angle information is included in the feature matrix of the pose graph.

PoseR can classify behaviors of other species and understand environmental context

PoseR can be extended to understand the patterns of movements of other animals, and within their environmental context. We trained two ST-GCN networks to classify three mouse behaviors recorded in an open field test (Sturman et al., 2020), with one network using only pose information from the mouse body, and the other using pose information from the mouse body and an additional five points demarcating the corners and center of the arena (Fig 5A). The dataset contains manually labelled rearing and grooming behaviors, with the rearing behaviors sub-divided into supported, where the mouse leans on the arena to rear, and unsupported, where it does not (Fig 5B). Neural networks trained on body-only pose information excelled at identifying groom behaviors (92% accuracy) from rearing behaviors (100% accuracy), however due to the lack of environmental context within the pose estimation unsupported rearing was more often confused with supported rearing (Fig 5C). Including information about the arena as nodes within the pose graph led to enhanced performance in recognizing and distinguishing supported and unsupported rearing and grooming producing a more balanced behavioral model with accuracies of 89 %, 75 %, 86 % for support and unsupported rearing and grooming respectively. These results demonstrated the versatility of ST-GCN models to learn behaviors of other animal models, as well as their ability to use information about the environment to make accurate predictions (Fig 5D).

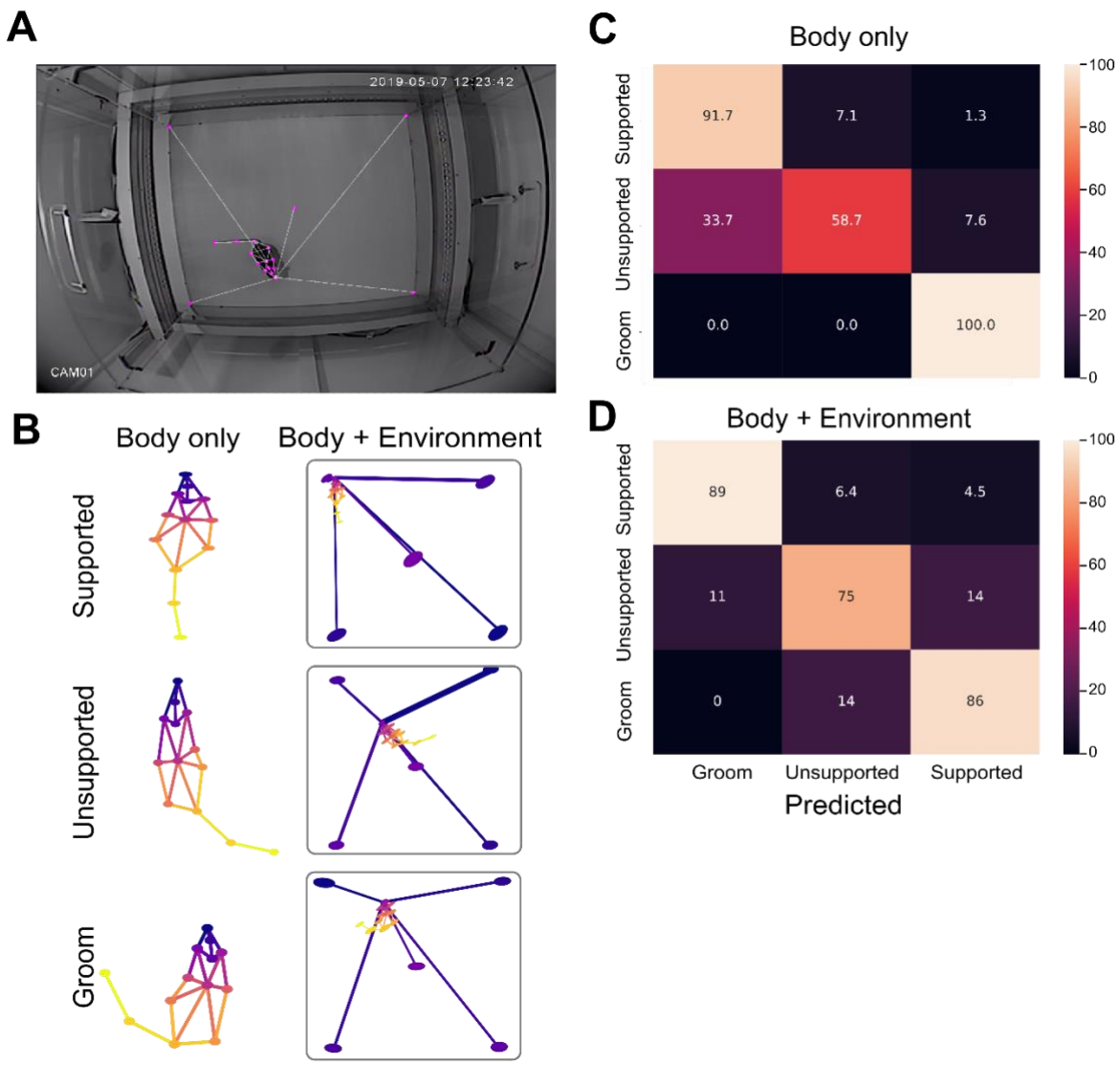


Figure 5. Accurate decoding of mouse behavior in the open field test. A) The open field arena with tracked key points of the mouse body as well as the arena overlayed in magenta. **B)** Dataset contains three behaviors: supported and unsupported rearing, and grooming; these were modelled with a body only or body plus environment spatial temporal pose graph. **C, D)** Confusion matrices showing the performance of an ST-GCN network trained on body-only pose information and body plus environment pose information, respectively.

Dataset	Paper	Accuracy	F1	Recall	Precision
ZebLR	This	0.90	0.90	0.90	0.90
ZebTensor	This	0.76	0.76	0.76	0.77
ZebTensor (weighted)	This	0.77	0.77	0.77	0.78
ZebRep	(Marques, Lackner, Félix, et al., 2018)	0.88	0.89	0.88	0.90

Open field test (body only)	(Sturman et al., 2020)	0.8	0.8	0.8	0.81
Open field test (body + environment)	(Sturman et al., 2020)	0.84	0.86	0.84	0.89

Table 1. Accuracy metrics on datasets used to train models in PoseR

Discussion

Here, we presented a deep learning toolbox, PoseR, with the aim to accelerate the understanding of animal behavior. Using the versatile zebrafish larva as an animal model, we demonstrated the flexibility of PoseR in extracting and annotating behaviors from pose estimations and training deep learning models to recognize a range of these behaviors. We also highlight the versatility of applying tensor decomposition to behavioral data as a powerful method for dimensionality reduction preceding unsupervised behavior discovery. We designed PoseR to be fast and accessible: we leveraged Python libraries that offer user-friendly solutions to training deep learning models; use shallow networks that learn classification boundaries without adding computational overhead; and combined these tools into an established and responsive data viewer, Napari. Our approach is validated by the rapid inference speeds and accurate models on initial first-pass training. Models can be further refined within PoseR using finetuning functions to freeze and unfreeze select layers during training. This offers the ability to modify our pretrained models for animal behavior to new behavioral classes by, for example, training only the last classification layer of the model. We have demonstrated that PoseR is also inherently agnostic to animal species, relying only on pose estimations. Furthermore, researchers studying the interactions of animals with their environment can include key point coordinates representing important features in the environment within our framework to understand how animals behave in a context-specific manner.

Whilst PoseR can be used with CPU-only systems, optimal throughput is achieved with GPU-based systems, drastically cutting inference speed and the time required to train a model. The current framework is built upon pose estimation outputs from the popular DeepLabCut Python

package; however, future integration with other animal pose estimation software is planned. The classification models presented here are modified spatial-temporal graph convolutional networks. These networks have several advantages in that they require only pose information and confidence intervals of those estimations, and can simultaneously study the spatial and temporal components of these poses. Application of these networks to human action recognition datasets greatly improved classification accuracy in these challenges, and subsequently these types of models are beginning to be adopted in the study of animal behavior (Zhao et al., 2022). As the field of action recognition advances, further improvements in neural network architecture will be implemented within PoseR. These strategies typically involve including more context of pose estimations, converting them to multidimensional heatmaps as input for three-dimensional convolutional neural networks, and including an RGB video layer (Duan et al., 2021). Viewing pose estimations as graph structures is a powerful approach and can be further extended by expanding the number of features associated with each body part node. For example, local video features at the location of each pose key point can be assigned to each node and included in the feature matrix for convolution. This would enable the network to learn the local video context in addition to spatial representation and estimation confidence of behaviors. Further, research into graph to text conversion and the use of language models to extract information from graphs is a very promising direction towards semantically describing behavior in an unsupervised way using only information contained in a pose graph. By including an individual and its environment in the open field test pose graph, we created a basic scene graph, where objects, individuals and their environment can be modelled as connected components. This could be further extended to more complex environments with multiple individuals and objects, and provide more utility for researchers performing complex behavioral experiments involving interaction with other individuals and the environment.

Applying tensor decomposition provided a powerful way to extract distinct swim types and create a large behavioral dataset to train and test our supervised ST-GCN models. Previous

attempts to extract zebrafish larvae behavior have used a range of techniques from t-SNE embedding, density-based clustering to FuzzyArt algorithms to discover and describe zebrafish behaviors in a range of experimental conditions (Johnson et al., 2020; Marques, Lackner, Félix, et al., 2018; Yang et al., 2021). However, unsupervised clustering, whilst a powerful and useful technique, does not represent an absolute truth of the number or separation of behaviors. We report a division of our swim bouts derived from visual stimuli and predation assays into 30 clusters. In similar experimental setups, others have reported a range of distinct swim types from 13 (Marques, Lackner, Félix, et al., 2018) to 36 (Johnson et al., 2020). We created an ST-GCN model to accurately decode the 13 behaviors from (Marques, Lackner, Felix, et al., 2018) using only the angle information of the tail as no x, y pose estimation data was evident in this dataset. The ability of the ST-GCN model to learn these different behaviors from one feature alone was testament to the versatility of graph neural networks and the well-clustered swim types in this dataset. However, our main aim was to teach ST-GCN models to recognize behavior from pose estimations, and we were unable to train an adequate model to learn the 36 behaviors from Johnson et al.. We do not know whether this was due to the larger number of clusters, or as a consequence of the low temporal resolution (60 fps) of this dataset compared to our own (330 fps) and Marques et al. (700 fps). This provided the rationale for creating our own dataset, presented and openly accessible here, with high temporal resolution pose estimations of larval behavior evoked by a variety of visual stimuli and prey.

The PoseR toolbox developed here sits within a healthy ecosystem of emerging and maturing behavioral analysis tools that aim to discover behaviors in an unsupervised way or classify behaviors manually. Our approach of skeleton-based action recognition performs well on datasets such as the open field test as an alternative to tools such as DeepEthogram that use video information only (Bohnslav et al., 2021). An advantage of skeleton-based action recognition over video-based action recognition is that it is invariant to background in the video. In our case, we acquired high frame rate (>300 fps) videos required for capturing

zebrafish larvae behavior by dynamically cropping the region of interest around the zebrafish during acquisition to reduce image size and to save to disk in real time. This led to a dynamically shifting non-uniform background flow that would interfere with video flow-based methods of behavioral analysis. Where a camera's field-of-view is not stable, whether during recording in the wild or, as in our case, where a camera's field of view is cropped to track a fast-moving animal, we view our skeleton-based approach as the optimal one. An important aim of neuroscience is to understand how neural activity underlies behavior. Crucially, the latencies of the decoders presented here are sufficient to enable real-time decoding of behavior. Our tool, combined with fast pose estimation (Kane et al., 2020) and neural activity recording, therefore offers an exciting opportunity to directly correlate *in vivo* neural activity with behavior, and trigger experimental manipulation with behavioral cues, advancing the effort to understand how the brain produces behavior.

Funding

This work was supported by the Biotechnology and Biological Sciences Research Council Grant BB/T006560/1, and an RS Macdonald Charitable Trust Grant.

Acknowledgements

We would like to acknowledge David Harris-Birtill for his invaluable advice, Cat Hobaiter and Stefan Pulver for their input and data-sharing during the conception and early stages of the project, and Jacqueline MacPherson, Michael Kinnear, James Lewis-Cheetham and Angus Aitken from the Psychology Workshop for their technical support. We'd also like to thank Joe Chapman and technical staff from the Scottish Ocean Institute for zebrafish husbandry support.

Berman, G. J., Choi, D. M., Bialek, W., & Shaevitz, J. W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, *11*(99). https://doi.org/10.1098/RSIF.2014.0672

Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., Kashlan, A. D., Chiappe, M. E., Orefice, L. L., Woolf, C. J., & Harvey, C. D. (2021). DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *ELife*, *10*. https://doi.org/10.7554/ELIFE.63377

Borla, M. A., Palecek, B., Budick, S., & O'Malley, D. M. (2002). Prey capture by larval zebrafish: Evidence for fine axial motor control. *Brain, Behavior and Evolution*, *60*(4), 207–229. https://doi.org/10.1159/000066699

Budick, S. A., & O'Malley, D. M. (2000). Locomotor repertoire of the larval zebrafish: Swimming, turning and prey capture. *Journal of Experimental Biology*, *203*(17), 2565–2579. https://doi.org/10.1242/JEB.203.17.2565

Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J., & Perona, P. (2009). Automated monitoring and analysis of social behavior in Drosophila. *Nature Methods 2009 6:4*, *6*(4), 297–303. https://doi.org/10.1038/nmeth.1310

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

Duan, H., Zhao, Y., Chen, K., Lin, D., & Dai, B. (2021). Revisiting Skeleton-based Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 2959–2968. https://doi.org/10.48550/arxiv.2104.13586

Fero, K., Yokogawa, T., & Burgess, H. A. (2011). The behavioral repertoire of larval zebrafish. *Neuromethods*, *52*, 249–291. https://doi.org/10.1007/978-1-60761-922-2_12/FIGURES/4

GitHub - open-mmlab/mmskeleton: A OpenMMLAB toolbox for human pose estimation, skeleton-based action recognition, and action synthesis. (n.d.). Retrieved February 17, 2023, from https://github.com/open-mmlab/mmskeleton

Guilbeault, N. C., Guerguiev, J., Martin, M., Tate, I., & Thiele, T. R. (2021). BonZeb: open-source, modular software tools for high-resolution zebrafish tracking and analysis. *Scientific Reports 2021 11:1*, *11*(1), 1–21. https://doi.org/10.1038/s41598-021-85896-x

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, *2017-December*, 1025–1035. https://arxiv.org/abs/1706.02216v4

Hsu, A. I., & Yttri, E. A. (2021). B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications 2021 12:1*, *12*(1), 1–13. https://doi.org/10.1038/s41467-021-25420-x

Jiménez-Luna, J., Grisoni, F., & Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence 2020 2:10*, *2*(10), 573–584. https://doi.org/10.1038/s42256-020-00236-4

Johnson, R. E., Linderman, S., Panier, T., Wee, C. L., Song, E., Herrera, K. J., Miller, A., & Engert, F. (2020). Probabilistic Models of Larval Zebrafish Behavior Reveal Structure on Many Scales. *Current Biology*, *30*(1), 70-82.e4. https://doi.org/10.1016/j.cub.2019.11.026

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R.,

Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature 2021 596:7873*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S., & Branson, K. (2013). JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods 2012 10:1*, *10*(1), 64–67. https://doi.org/10.1038/nmeth.2281

Kalueff, A. V., Gebhardt, M., Stewart, A. M., Cachat, J. M., Brimmer, M., Chawla, J. S., Craddock, C., Kyzar, E. J., Roth, A., Landsman, S., Gaikwad, S., Robinson, K., Baatrup, E., Tierney, K., Shamchuk, A., Norton, W., Miller, N., Nicolson, T., Braubach, O., … Schneider, H. (2013). Towards a Comprehensive Catalog of Zebrafish Behavior 1.0 and Beyond. *Zebrafish*, *10*(1), 70. https://doi.org/10.1089/ZEB.2012.0861

Kane, G. A., Lopes, G., Saunders, J. L., Mathis, A., & Mathis, M. W. (2020). Real-time, low-latency closed-loop feedback using markerless posture tracking. *ELife*, *9*, 1–29. https://doi.org/10.7554/ELIFE.61909

Kolda, T. G., & Bader, B. W. (2009). Tensor Decompositions and Applications. *SIAM Rev.*, *51*(3), 455–500. https://doi.org/10.1137/07070111X

Lightning-AI/lightning: Deep learning framework to train, deploy, and ship AI products Lightning fast. (n.d.). Retrieved April 7, 2023, from https://github.com/Lightning-AI/lightning

Lopes, G., Bonacchi, N., Frazão, J., Neto, J. P., Atallah, B. V., Soares, S., Moreira, L., Matias, S., Itskov, P. M., Correia, P. A., Medina, R. E., Calcaterra, L., Dreosti, E., Paton, J. J., & Kampff, A. R. (2015). Bonsai: An event-based framework for processing and controlling data streams. *Frontiers in Neuroinformatics*, *9*(APR). https://doi.org/10.3389/fninf.2015.00007

Marques, J. C., Lackner, S., Felix, R., & Orger, M. B. (2018). *Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering*. 181–195. https://doi.org/10.1016/j.cub.2017.12.002

Marques, J. C., Lackner, S., Félix, R., & Orger, M. B. (2018). Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering. *Current Biology*, *28*(2), 181-195.e5. https://doi.org/10.1016/j.cub.2017.12.002

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*. https://doi.org/10.1038/s41593-018-0209-y

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. https://arxiv.org/abs/1802.03426v3

napari: a multi-dimensional image viewer for Python | Zenodo. (n.d.). Retrieved February 1, 2023, from https://zenodo.org/record/7276432#.Y9nbhq3P1EY

neurostatslab/tensortools: A very simple and barebones tensor decomposition library for CP decomposition a.k.a. PARAFAC a.k.a. TCA. (n.d.). Retrieved March 28, 2023, from https://github.com/neurostatslab/tensortools

Orger, M. (2018). *1. Structure of the zebrafish locomotor repertoire revealed with unsupervised behavioural clustering Marques*. *1*. https://doi.org/10.17632/R9VN7X287R.1

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1912.01703v1

Patterson, B. W., Abraham, A. O., MacIver, M. A., & McLean, D. L. (2013). Visually guided gradation of prey capture movements in larval zebrafish. *Journal of Experimental Biology*, *216*(16), 3071–3083. https://doi.org/10.1242/jeb.087742

Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., Papadoyannis, E. S., Normand, E., Deutsch, D. S., Wang, Z. Y., McKenzie-Smith, G. C., Mitelut, C. C., Castro, M. D., D'Uva, J., Kislin, M., Sanes, D. H., Kocher, S. D., Wang, S. S. H., Falkner, A. L., … Murthy, M. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods 2022 19:4*, *19*(4), 486–495. https://doi.org/10.1038/s41592-022-01426-1

Ro, S., †1, N., Goodwin, N. L., Choong, J. J., Hwang, S., Wright, H. R., Norville, Z. C., Tong, X., Lin, D., Bentzley, B. S., Eshel, N., Mclaughlin, R. J., & Golden, S. A. (2020). Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals. *BioRxiv*, 2020.04.19.049452. https://doi.org/10.1101/2020.04.19.049452

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(C), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J. J., Perona, P., Anderson, D. J., & Kennedy, A. (2021). The mouse action recognition system (MARS) software pipeline for automated analysis of social behaviors in mice. *ELife*, *10*. https://doi.org/10.7554/ELIFE.63720

Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Bokota, G., Winston, P., Peña-Castellanos, G., Yamauchi, K., Bussonnier, M., Doncila Pop, D., Can Solak, A., Liu, Z., Wadhwa, P., Burt, A., Buckley, G., Sweet, A., Migas, L., Hilsenstein, V., Gaifas, L., … McGovern, A. (2022). *napari: a multi-dimensional image viewer for Python*. https://doi.org/10.5281/ZENODO.7276432

Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., Grimm, C., Thieren, L., Zerbi, V., Grewe, B., & Bohacek, J. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology 2020 45:11*, *45*(11), 1942–1952. https://doi.org/10.1038/s41386-020-0776-y

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods 2020 17:3*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Weinreb, C., Osman, M. A. M., Zhang, L., Lin, S., Pearl, J., Annapragada, S., Conlin, E., Gillis, W. F., Jay, M., Ye, S., Mathis, A., Mathis, M. W., Pereira, T., Linderman, S. W., & Datta, S. R. (2023). Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *BioRxiv*, 2023.03.16.532307. https://doi.org/10.1101/2023.03.16.532307

Williams, A. H., Kim, T. H., Wang, F., Vyas, S., Ryu, S. I., Shenoy, K. V., Schnitzer, M., Kolda, T. G., & Ganguli, S. (2018). Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. *Neuron*, *98*(6), 1099-1115.e8. https://doi.org/10.1016/J.NEURON.2018.05.015

Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., Abraira, V. E., Adams, R. P., & Datta, S. R. (2015). Mapping Sub-Second Structure in Mouse Behavior. *Neuron*, *88*(6), 1121–1135. https://doi.org/10.1016/j.neuron.2015.11.031

Wiltschko, A. B., Tsukahara, T., Zeine, A., Anyoha, R., Gillis, W. F., Markowitz, J. E., Peterson, R. E., Katon, J., Johnson, M. J., & Datta, S. R. (2020). Revealing the

structure of pharmacobehavioral space through motion sequencing. *Nature Neuroscience 2020 23:11*, *23*(11), 1433–1443. https://doi.org/10.1038/s41593-020-00706-3

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 7444–7452.

Yang, P., Takahashi, H., Murase, M., & Itoh, M. (2021). Zebrafish behavior feature recognition using three-dimensional tracking and machine learning. *Scientific Reports*, *11*(1), 1–9. https://doi.org/10.1038/s41598-021-92854-0

Zhao, Y., Feng, L., Tang, J., Zhao, W., Ding, Z., Li, A., & Zheng, Z. (2022). Automatically recognizing four-legged animal behaviors to enhance welfare using spatial temporal graph convolutional networks. *Applied Animal Behaviour Science*, *249*, 105594. https://doi.org/10.1016/J.APPLANIM.2022.105594