OPEN BIOLOGY

royalsocietypublishing.org/journal/rsob

Methods and Techniques

Check for updates

Cite this article: Mullen PN, Bowlby B, Armstrong HC, Gray A, Zwart MF. 2026 PoseR: a deep learning toolbox for classifying animal behaviour. *Open Biol.* **15**: 250322.
https://doi.org/10.1098/rsob.250322

Received: 25 August 2025
Accepted: 13 November 2025

Subject Areas:
neuroscience

Keywords:
deep learning, behaviour classification, computer vision

Author for correspondence:
Maarten F. Zwart
e-mail: mfz@st-andrews.ac.uk

Supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.8168694.

THE ROYAL SOCIETY PUBLISHING

PoseR: a deep learning toolbox for classifying animal behaviour

Pierce N. Mullen, Beatrice Bowlby, Holly C. Armstrong, Angus Gray and Maarten F. Zwart

School of Psychology and Neuroscience, Centre of Biophotonics, University of St Andrews, St Andrews, UK

MFZ, 0000-0002-5073-8631

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 using markerless pose estimation models. Current methods for classifying animal behaviour using the outputs of these models often rely on species and task-specific feature engineering of trajectories, kinematics and task programming. Generalized solutions that use only pose estimations and the inherent structure of animals and their environment provide an opportunity to develop foundational, contextual and, importantly, standardized animal behaviour models for efficient and reproducible behavioural analysis. Here, we present PoseRecognition (PoseR), a behavioural classifier using spatio-temporal graph convolutional networks. We show that it can be used to classify animal behaviour quickly and accurately from pose estimations, using zebrafish larvae, *Drosophila melanogaster*, mice and rats as model organisms. Our easily accessible tool simplifies the behavioural analysis workflow by transforming coordinates of animal position and pose into semantic labels with speed and precision. The design of our tool ensures scalability and versatility for use across multiple species and contexts, improving the efficiency of behavioural analysis across fields.

1. Introduction

Decoding animal behaviour 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 behaviour remains a more difficult problem for neuroscientists [1,2]. Existing analysis workflows fall into two categories: unsupervised behavioural discovery or supervised behavioural classification. Unsupervised discovery leverages machine learning models to identify distinct actions of animals from either raw videos or pose estimations requiring no prior knowledge [3–9]. This approach is useful for researchers looking to discover new behaviours but requires thorough *post hoc* analysis and is sensitive to parameter selection and subjective interpretation of the number of clusters. As a result, it may lead to inter-lab discrepancies within the same dataset. Alternatively, supervised classification applies prior knowledge of the behaviours that a researcher would like to extract from video recordings and teaches them to machine and deep learning models to improve the efficiency of analysis [10–16]. The advantage of a generalized classifier is that there is no requirement for generating new embedding spaces of pose and features or aligning new data to pre-existing latent spaces; classifiers relying only on pose can be used in a plug-and-play fashion making them more accessible to the research community. Previous classifier approaches have typically focused

on specific contexts of behaviour of mice, fruit flies and, to a limited extent, zebrafish, often requiring the pre-computation of species-specific features or task programming [16,17]. To produce a classifier architecture that does not require extensive feature or task engineering and generalizes across contexts, backgrounds and species, we propose representing animal pose as a skeleton-like graph upon which temporal and spatial relationships between nodes can be learnt to accurately predict behaviour.

To this end of classifying diverse animal behaviour, we utilized skeleton-based action recognition deep learning architectures that have demonstrated success in human action recognition [18]. 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 [19–21]. 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 neighbour 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 behaviour, the pose graph can be transformed in this way and the temporal features of these aggregated nodes can then be learnt to classify behaviours.

2. Design and implementation

Our strategy to develop a new behavioural classifier based on these models was to simplify and accelerate three main steps in the behavioural analysis pipeline: (i) extraction, (ii) annotation and finally (iii) classification of behaviour. We first applied signal processing methods to identify windows in which behaviour is occurring, enabling rapid annotation of thousands of behaviours. The coordinates of the points on an animal body extracted using pose estimation, for example by DeepLabCut [1] or YOLO [22], are then used in space *and* time in spatial–temporal graph convolutional networks (ST-GCN). We included easy-to-use functions and an accompanying tool, called PoseRecognition (PoseR), as an open-source plugin for the popular multidimensional data viewer napari [23] 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 first develop an efficient behavioural analysis workflow and then demonstrate its general use when applied to other species, multi-animal social contexts, and when including environmental context. Zebrafish exhibit a large repertoire of behaviours encompassing environmental exploration,23 escape and predation [24]. Previous work [25,26] to develop classifiers of zebrafish behaviour has focused on binary classification of prey-capture swims versus non-prey capture swims in a fixed-head preparation [27,28]. Several studies have developed unsupervised methods to uncover zebrafish swim types of freely swimming zebrafish; K-means clustering was used to identify 15 swim types from swim kinematics [29], density-based clustering was used to identify 13 swim types from kinematics [26], the FuzzyART algorithm was used to reveal around 50 clusters from trajectories of adult zebrafish [30], hierarchical clustering of four kinematic parameters was used to reveal three prey-response swims [31], autoencoder latent-embedding coupled with principal component analysis and dimensionality-reduction of spectrograms was used to obtain 22 clusters from raw video [32], tSNE dimensionality reduction of 220 swim bout features was used to obtain 36 mirrored-swim types from videos recorded at 60 frames per second (fps) [25] and, finally, independent-component analysis was used to obtain four swim types [33]. These approaches further highlight the variability in the number of swim types produced from unsupervised methods. Further work has also elegantly examined fundamental principles, motifs and sequential organization of zebrafish behaviour [34–36]. Despite being exceedingly rich datasets, none satisfied all the criteria needed for efficiently developing an end-to-end behavioural classification pipeline of (i) being openly accessible, (ii) containing raw pose estimates and more than two behavioural classes, and (iii) having high framerates (>60 fps). We therefore sought to produce and open source our own large, high-frame-rate dataset of zebrafish larval poses and swim types using novel unsupervised methods to test the accuracy of graph convolutional classifiers in learning zebrafish behaviour. It is possible to elicit a wide range of zebrafish behaviours 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 a riverbed [25,26]. We acquired high-speed videos (330 fps) of freely swimming zebrafish larvae under these conditions and tested the effectiveness of our end-to-end behavioural analysis toolbox PoseR in two scenarios: generating and classifying a small manually curated dataset and classifying a large novel dataset generated through unsupervised clustering of zebrafish behaviours. We also tested PoseR on a mouse open field dataset [14], a pre-clustered zebrafish dataset containing tail angles [26] and mouse, fly and rat datasets of social behaviour [37–39] to demonstrate the applicability of this approach to the analysis of other species and species-specific social and context-dependent behaviours.

2.1. Behavioural 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 × 25 × 25 mm) containing system water. Visual projections were displayed onto diffuse acrylic beneath the recording chamber using a cold mirror (Edmund Optics, 45° AOI, 50.0 mm Square, Cold Mirror, no. 64 451) and a projector (Epson EF-11 3LCD, no. 0011131458). The zebrafish larvae were illuminated using a custom infra-red LED (850 nm) array beneath the chamber and recorded at 330 fps using a Mikrotron camera (MC1362) and a high-speed frame grabber (National Instruments, PCIe-1433).

Images were acquired and dynamically cropped in Bonsai [40] 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 [31] 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.

2.2. Pose estimation

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

2.3. Swim bout extraction

Due to the discrete bout-like nature of zebrafish swimming behaviour it was relatively straightforward to define and extract periods in which behaviour 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 [43] 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 centre 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 = fps/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.

2.4. 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 behaviours. 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 an hdf5 file format to store the individual identity, pose estimation, swim bout number, behavioural 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.

2.5. Swim bout unsupervised clustering

From the behavioural recordings,34 015 swim bout pose estimations were extracted, resulting in an $N \times C \times T \times V$ 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 with respect to the central node during the swim bouts was computed for each node, resulting in an $N \times T \times V$ array containing angle changes. The dimensionality of this array was reduced using tensor decomposition [44–46], where the data are 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) × components factor matrix, a time (T) × components factor matrix and a node (V) × 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%, while 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 [47] and high silhouette score [48] in cluster evaluation.

2.6. Spatial temporal graph convolutional network

An ST-GCN [18,49] was modified into a Pytorch-Lightning module and the final architecture optimized 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 where train, validation and test splits were not already provided in open-source datasets, and accuracy, precision, F1 score and recall were calculated to evaluate the performance of each model.

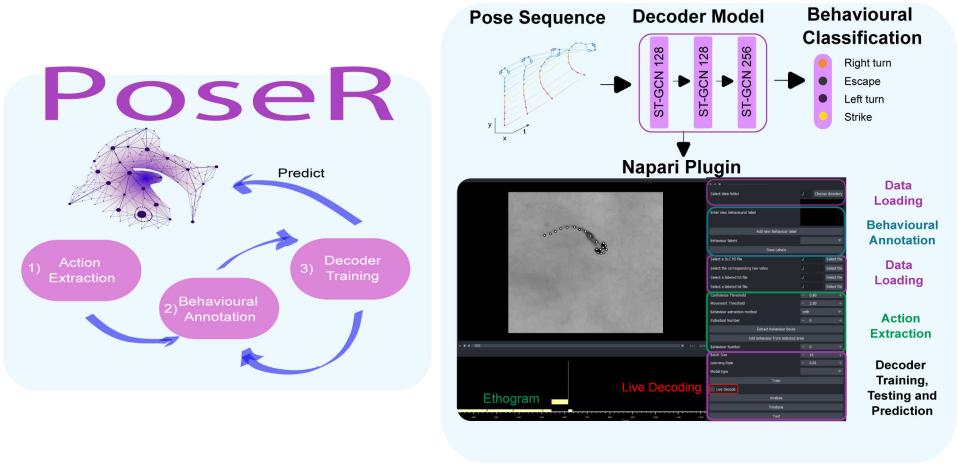

Figure 1. PoseR: a deep learning behavioural classification toolbox. A napari plugin designed to accelerate extraction, annotation, and training deep learning classifiers to predict animal behaviours. It combines popular deep learning packages Pytorch [41] and Pytorch Lightning [42] to simplify the process of training deep learning behavioural classifiers for a wide range of applications.

2.7. Installation

PoseR is installable via pypi https://pypi.org/project/PoseR-napari/ or via GitHub https://github.com/pnm4sfix/PoseR.

3. Results

3.1. PoseR enables fast coding and decoding of a small set of behaviours

We initially aimed to develop and validate our toolbox by rapidly coding a small subset of trivial zebrafish behaviours (left, right and forward swims) and designing an ST-GCN model capable of correctly decoding these. While in practice classifying left versus 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 behaviour. 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 approximately 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 learnt using a ST-GCN and the subsequently trained model can be used to classify behaviour in an experimental setting [18]. 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 (figure 2A). Testing the model on the validation dataset resulted in a high average accuracy of 90% across swim types (figure 2B). We calculated a value of 0.9 for precision, recall and F1-score of the model's classifications versus ground truth. Precision quantifies the ratio of true positives to total positive classifications 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 (figure 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 centred 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 behaviours. 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 behaviours.

3.2. Generating a comprehensive zebrafish behaviour dataset using tensor decomposition and agglomerative clustering

We next generated a larger dataset of zebrafish larval behaviours. We recorded long duration (40 min), high frame rate (330 fps) videos of zebrafish larvae behaving and responding to a wide range of closed-loop visual stimuli [31]. These stimuli were chosen to elicit natural behaviour such as escape reflexes, phototaxis and optomotor responses, resulting in a dataset of approximately 30 000 swim bouts at high temporal resolution. We took a novel approach to clustering these swim bouts by first using tensor decomposition [44,45] 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 (figure 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

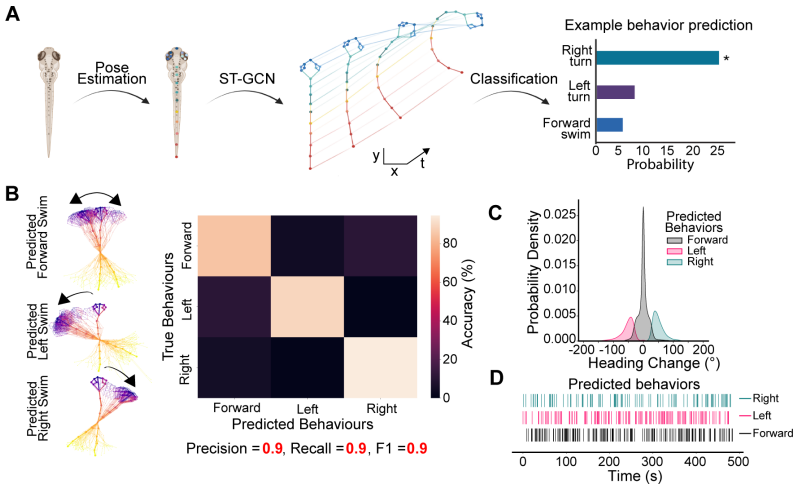

Figure 2. Deep learning behaviour classification strategy. (A) Graphical abstract created in BioRender (DJ27IPKD7A) demonstrating the workflow of classifying sequences of poses as zebrafish behaviour. Zebrafish larval poses were extracted from videos using DeepLabCut. Sequences of poses can be represented 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.

(figure 3B). To create a challenging dataset, we defined optimal clustering criteria as having a minimum cluster number of 15 and well-separated clusters with a silhouette score of >0.2. This resulted in 30 distinct behaviours, which contained swim bouts that were homogeneous within each cluster and showed similarity to swim bout types that have been previously described [26,50] (electronic supplementary material, figure 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, which we mapped to a low dimension behavioural projection of each swim using uniform manifold approximation and projection [51] (figure 3C, electronic supplementary material, figure S1A). Within these classes, the temporal dynamics varied depending on the vigour, amplitude and number of tail oscillations and whether changes in direction 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° with no scoot [50]. More vigorous 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 vigour in almost the opposite direction resembling short-latency C-bend swim (SLC) [50]. Long-latency C-bends (LLC), where the heading change and swim vigour 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 more vigorous forward swims and turns alongside J-turns that are in particular prevalent in zebrafish predation (figure 3D) [26,52–54]. Using this novel approach, we succeeded in extracting and generating a large, complex dataset of a range of swim bout behaviours that zebrafish larvae employ during different visual contexts.

3.3. PoseR can be used to rapidly classify many complex behaviours

We next used PoseR to train an ST-GCN to recognize these more complex clustered swim behaviours with the aim of developing a universal zebrafish classifier that could be applied across a range of experiments. We found a high top-1 and top-3 unweighted-average accuracy (76% and 97%, respectively) for correctly classifying all behaviours in the test dataset. and this was achieved on a first run using PoseR's built-in helper functions to optimize initial model hyperparameters. Top-1 and top-3 accuracy is often used to report the presence of correct behavioural classification in the top guess of the model (top-1) and in the top-3 guesses. 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 (figure 4A). The accuracy of the model's initial classifications 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 (figure 4A,B). In addition to fast optimized training, we endeavoured to evaluate the speed at predicting and analysing new behaviours on different systems accelerated by either a GPU or a CPU. As expected, we found faster inference speeds on GPU based systems compared with CPU; with classifying of 1000 swim bouts taking approximately 20 s with a batch size of 10 on GPU systems. The latency to analyse one bout on an Nvidia Titan RTX GPU system was 2.85 ± 0.22 ms (figure 4B). The different frequencies with which swim types occur resulted in an imbalanced dataset where some swim

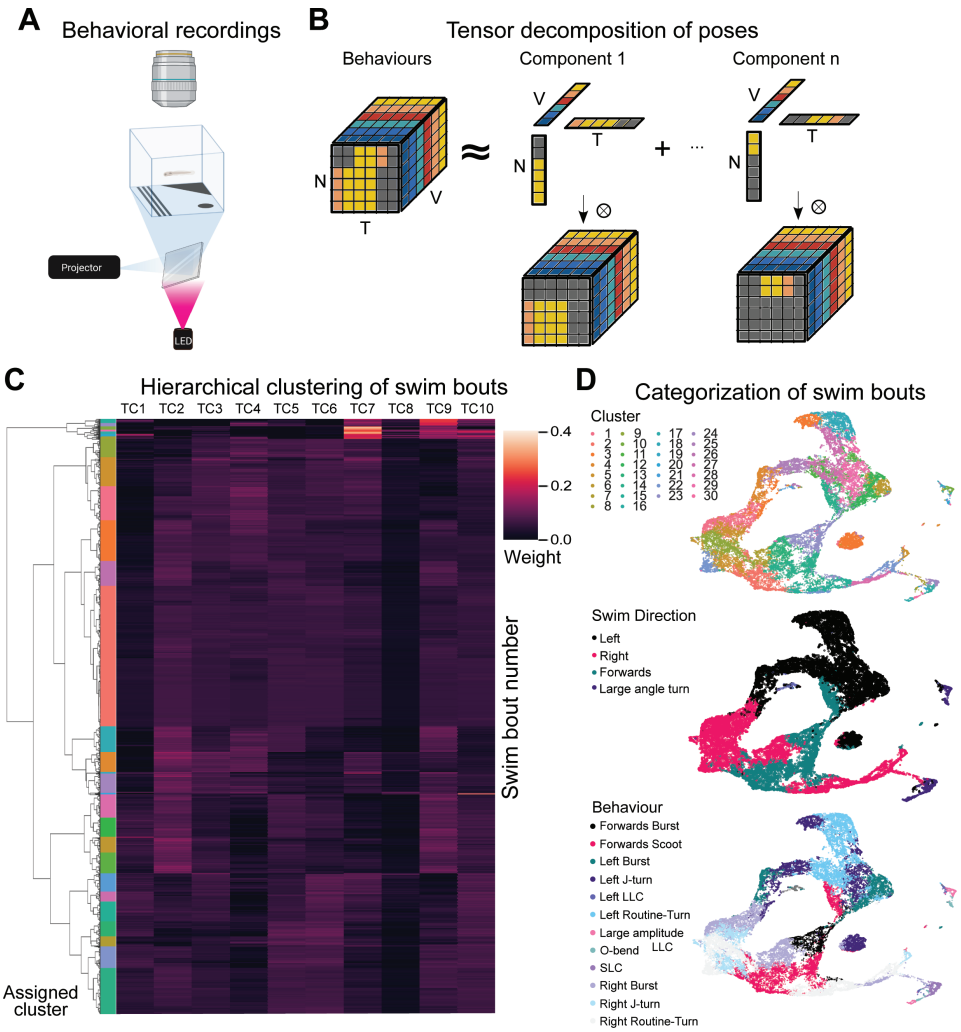

Figure 3. Clustering decomposed swim bouts. (A) Graphic of the behavioural experimental set-up for high-speed video recording and visual presentation, made in BioRender (II27IPKPNA). (B) Tensor decomposition (PARAFAC/CANDECOMP) was applied to a 3D (rank-3 tensor) dataset containing egocentric node positions (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. (C) Agglomerative hierarchical clustering heatmap of the N × TC factor matrix showing how swim bouts could be grouped into similar clusters. (D) A uniform manifold approximation and projection of the N × TC factor matrix colour-coded by swim type cluster and broad swim class (left, right, forward and large angles 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 rate of occurrence of that swim type relative to the total dataset size. This produced a model with a slightly 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 develop models to accurately predict behaviours in unbalanced datasets (figure 4B–D). Graphs are 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 instead of x and y pose estimation coordinates of a zebrafish larval tail [26]. This dataset contained 13 swim types recorded at 700 fps, and angle information for eight 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 (figure 4E), demonstrating the powerful application of this approach to classifying behaviour from different types of data from a variety of sources.

3.4. PoseR can classify behaviours 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. To demonstrate the utility of PoseR and the skeleton approach for training accurate behavioural models in this context, two ST-GCN networks were trained to classify three mouse behaviours recorded in an open field test [14], with one network trained in PoseR using pose information from the mouse body, and the other trained in PoseR using pose information from the mouse body and an additional five points demarcating the corners and centre of the arena (figure 5A). This pre-existing dataset already contained manually labelled rearing and grooming behaviours, with the rearing behaviours sub-divided into supported, where the mouse leans on the arena to rear, and unsupported, where it does not (figure 5B). We excluded very rare

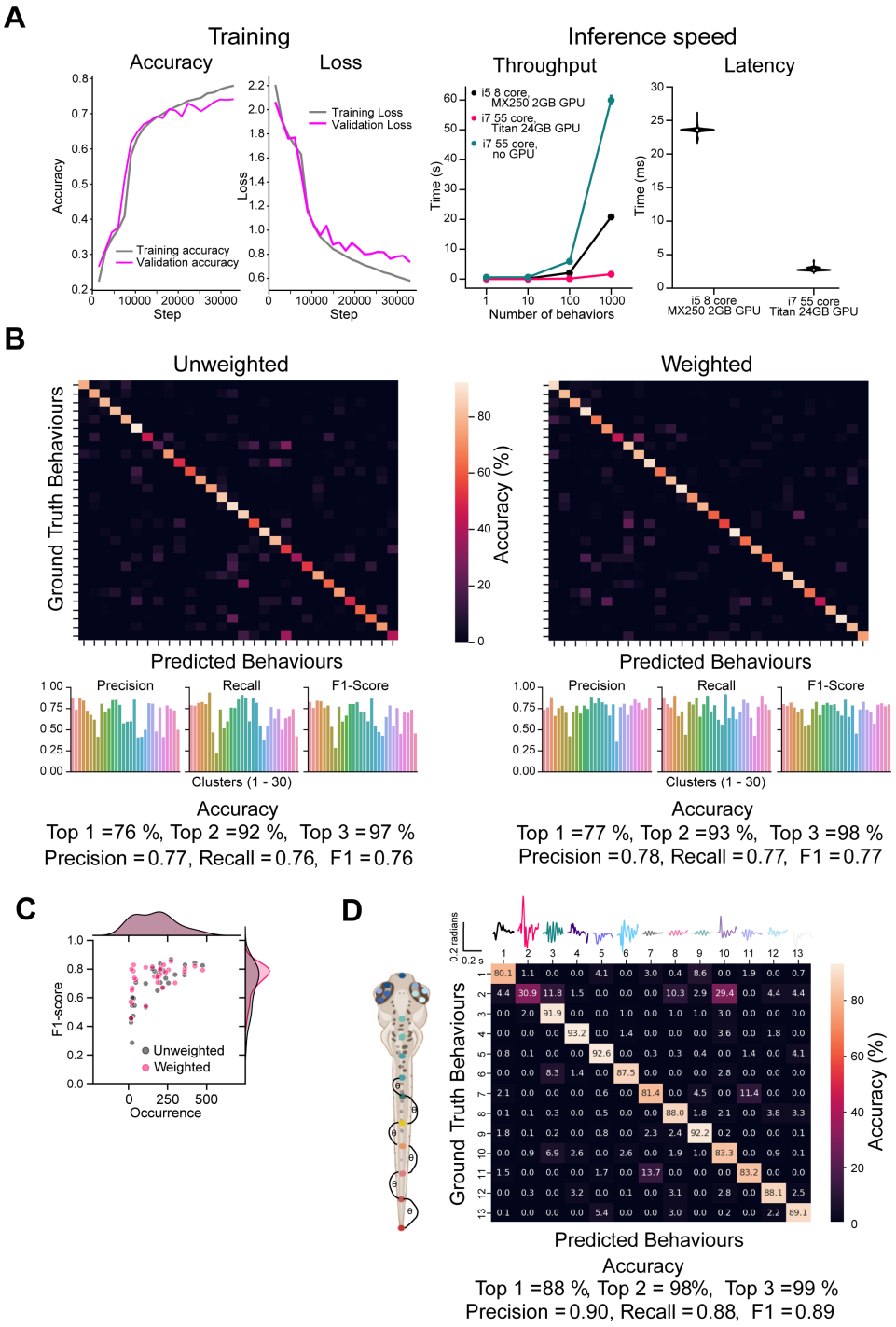

Figure 4. Accurate classification of zebrafish behaviour using a trained ST-GCN model. (A) Training and validation accuracy and loss reported during each training step. Latency for one sample and inference speed for batch sizes 1, 10, 100, 1000 on different systems: a CPU-only system, a system with 2GB MX250 GPU, and one with a 24 GB Titan RTX GPU. (B) Top: a confusion matrix highlighting the per cent of correct classifications compared with ground truth behavioural labels for unweighted and weighted behavioural training data. Bottom: precision, recall and F1-score for unweighted and weighted trained models. (C) A scatterplot of F1-score versus the occurrence of a behavioural label to show weighting labels improves the overall F1-score. (D) An ST-GCN model and subsequent confusion matrix trained on the ZebRep dataset [26,55] where only angle information is included in the feature matrix of the pose graph. The zebrafish graphic was created in BioRender (DJ27IPKD7A). Representative tail movements of the last tail segment are shown above for each swim type in the dataset (note angle in this dataset is defined as the angle of each segment relative to the more rostral segment in radians).

'jumping' labels for direct comparison with other tools that had removed these too. Behaviour bouts were extracted from the dataset according to the labelling metadata and split into train, validation and test dataset splits using PoseR. Neural networks trained on body-only pose information excelled at identifying groom behaviours (92% accuracy) from rearing behaviours (100% accuracy), however due to the lack of environmental context within the pose estimation unsupported rearing was more often confused with supported rearing (figure 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 behavioural model with accuracies of 89%, 75%, 86% for support and unsupported rearing and grooming, respectively. Distinguishing between unsupported and supported rearing could simply be established *post hoc* by the occurrence of the rear at the proximity of the wall, however here we use this example to demonstrate the versatility of ST-GCN models

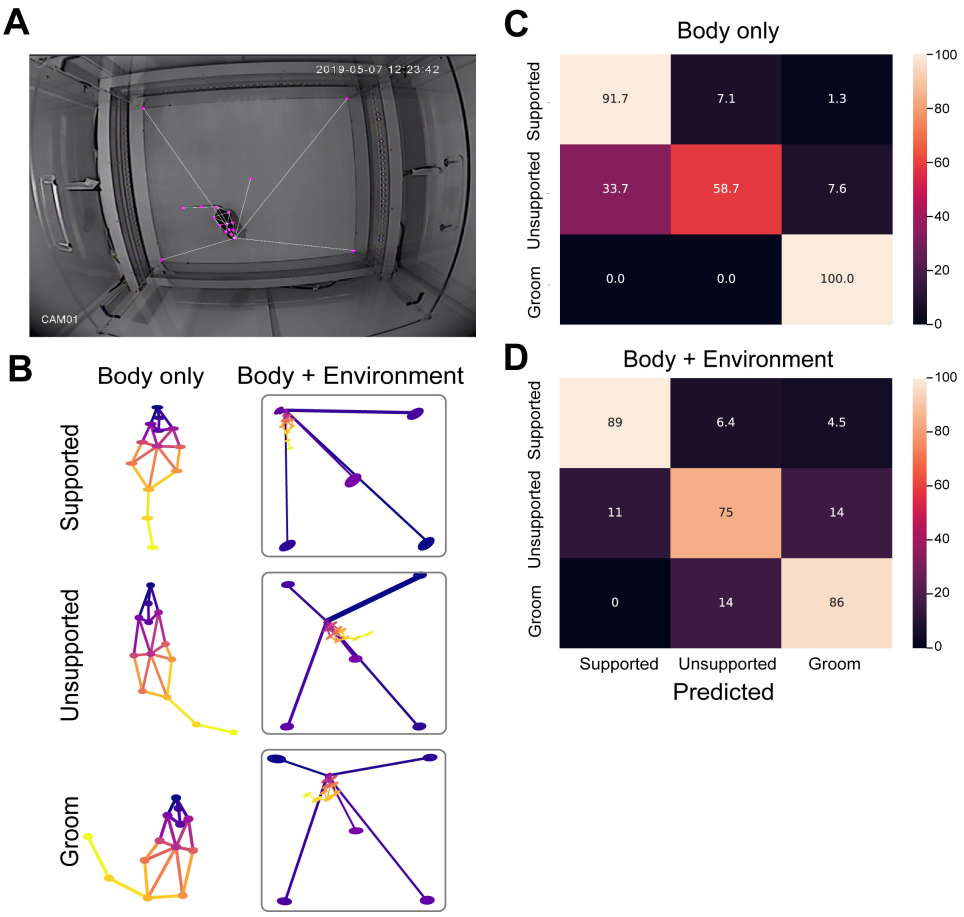

Figure 5. Accurate classification of mouse behaviour in the open field test. (A) The open field arena with tracked key points of the mouse body as well as the arena overlayed in magenta reproduced from Sturman *et al.* under CC By 4.0. (B) Dataset contains three behaviours: supported and unsupported 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.

to include information about the environment to intrinsically make accurate classifications about behaviours distinguishable by the proximity of the occurrence to specific objects or features in the environment (figure 5D).

3.5. PoseR models can be extended to classify social behaviours

Engagement in social behaviours is an important metric for screening the effect of different pharmacological compounds and an efficient method of classifying these behaviours is crucial. In addition to including environmental context within the pose graph, multiple individuals can be represented and combined as individuals and/or fully-connected community pose graphs to learn social behaviours across different animal species. We used three published datasets, FlyvFly [37], CALMS21 [39], PAIRR24M [56] to demonstrate the versatility of PoseR and the ST-GCN approach in classifying social behaviours. Briefly, FlyvFly contains pose estimations of three points, the tip of each wing and the body centre, of two flies interacting in a variety of contexts with behaviours labelled as lunge, wing-threat, tussle, wing-extension, circle or copulation (figure 6A). PAIRR24M consists of 25 three-dimensional (3D) body coordinates of two rats interacting in an open field arena, which we used to demonstrate the ability of PoseR to train models of 3D pose sequences representing rat social behaviour (figure 6B). CALMS21 is a dataset focused on four behaviours of two interacting mice, divided into two tasks defined by the number of annotators and size of training data available (figure 6C). We extracted pose sequences centred on each frame, with a window size of 14 frames. Pose estimations were assigned into train, validation and test splits and PoseR was used to train ST-GCN models to classify social behaviours. Models were able to accurately classify behaviours with respective accuracies of 98%, 85% and 62% in the test splits of FlyVFly, CALMS21 and PAIRR24M, respectively, with example ethograms showing a favourable comparison between frame-by-frame ground truth and model prediction. The ethograms highlight a good precision of behaviour across species and contexts, that is, the model accurately predicts when a behaviour does occur. However, there were several instances of false positives, where the model predicts a brief occurrence of behaviour when it does not occur, thus, some *post hoc* and annotator led refinements of these short false positives are necessary for best results; these tools are included in the PoseR plugin (figure 6D).

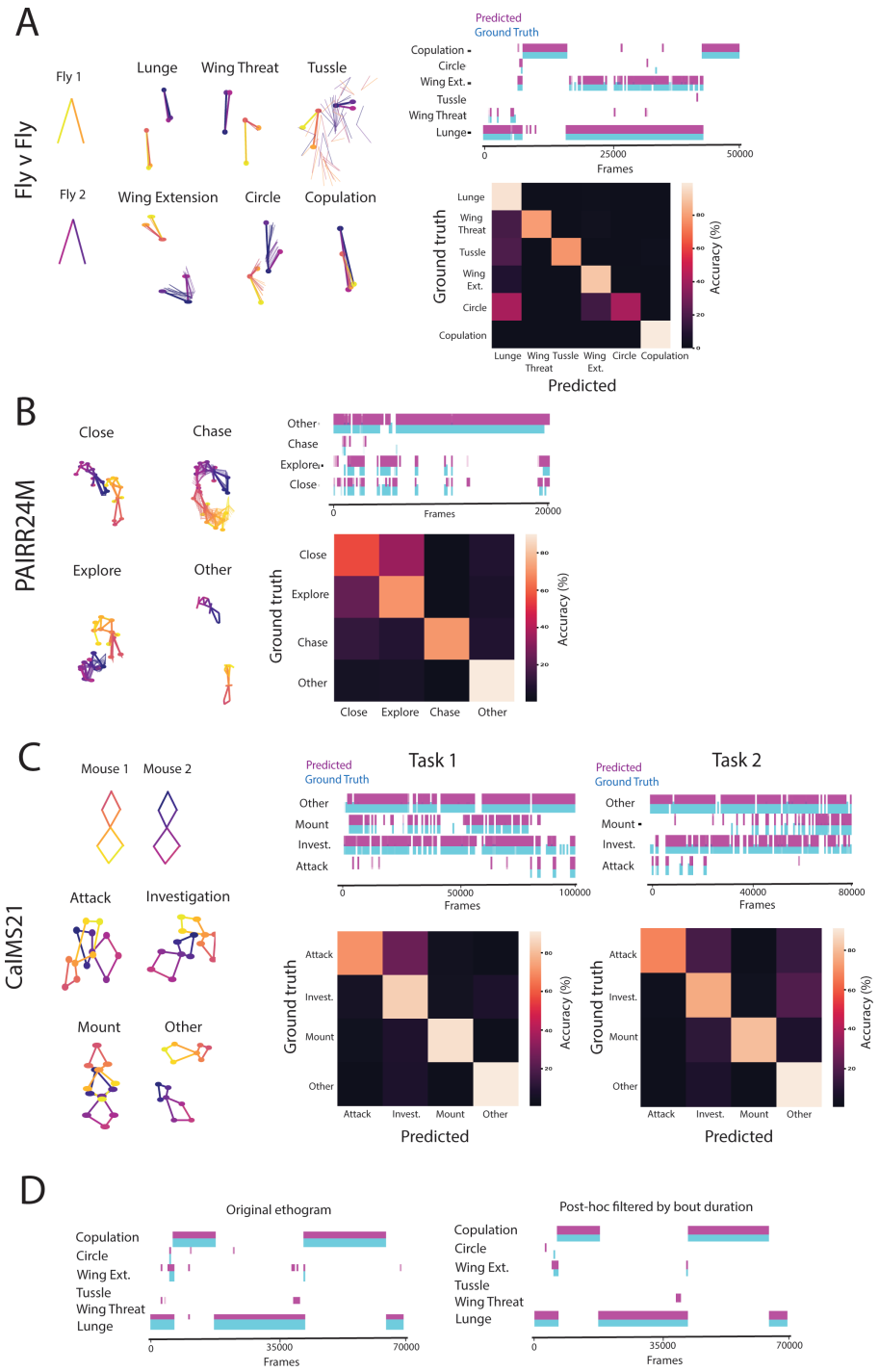

Figure 6. Classifying fly, mouse and rat social behaviours with PoseR. (A) Fly v fly dataset. Left: each fly is represented by three nodes that label the tip of each wing and the centre of fly. Middle: six behaviours were labelled in this dataset; lunge, wing-threat, tussle, wing-extension, circle and copulation and appear distinct based on their temporal and spatial pose graph. Top right: an ethogram showing a frame-by-frame comparison of the ground truth behavioural label and model classifications. Bottom right: a confusion matrix denoting the accuracy of the model for each behaviour. (B) PAIRR24M dataset. Left: each rat is represented by 25 nodes in 3D space. Social behaviours consisted of close, chase, explore and other. Top right: an ethogram comparing the frame-by-frame accuracy of the model classifications versus ground truth labels. Bottom right: a confusion matrix highlighting the accuracy of the model for each behaviour. (C) CALMS21 dataset. Left: each mouse is represented by seven nodes that label the body of the mouse. Four behaviours, attack, investigation, mount and other are contained in this dataset. Middle: performance on task 1 visualized by ethogram comparing model classifications versus ground truth and a confusion matrix showing behaviour level accuracy. Left: task 2 containing reduced amount of training data, showing an ethogram example and confusion matrix. (D) Left: original ethogram of FlyvFly video subset, right: the same ethogram after *post hoc* filtering defining the minimum bout duration for behaviour to remove erroneous noisy classifications.

4. Availability and future directions

Here, we presented a deep learning toolbox, PoseR, with the aim to accelerate the understanding of animal behaviour. Using the versatile zebrafish larva as an animal model, we demonstrated the flexibility of PoseR in extracting and annotating behaviours from pose estimations and training deep learning models to recognize a range of these behaviours. We also highlight the versatility of applying tensor decomposition to behavioural data as a powerful method for dimensionality reduction preceding unsupervised behaviour discovery. We designed PoseR to be fast and accessible: we leveraged Python libraries that offer

Table 1. Accuracy metrics on datasets used to train models in PoseR. 'Dataset' denotes the name used of the datasets used in this work, where 'ZebLR' is the simple proof of principle left/right/forward swim dataset, 'ZebTensor' is the tensor-decomposed clustered dataset of swim types to test the PoseR models on a large amount of behaviours, 'ZebRep' refers to the published Marques *et al.* dataset, 'open field test (body only)' and 'open field test (body + environment)' refer to the Sturman *et al.* dataset with and without arena context. FlyvFly, CALMS21 and PAIRR24 are multi-individual datasets for social behaviour. Accuracy, F1, recall and precision are averages across all behaviours within the specified dataset. In datasets where 'other' is present as a label, this is excluded from the final metric score.

dataset name	dataset reference	accuracy	F1	recall	precision
ZebLR	this work	0.90	0.90	0.90	0.90
ZebTensor	this work	0.76	0.76	0.76	0.77
ZebTensor (weighted)	this work	0.77	0.77	0.77	0.78
ZebRep	[26]	0.88	0.89	0.88	0.90
open field test (body only)	[14]	0.8	0.8	0.8	0.81
open field test (body + environment)	[14]	0.84	0.86	0.84	0.89
FlyvFly	[37]	0.98	0.70	0.79	0.65
CALMS21 Task 1	[39]	0.85	0.80	0.83	0.77
CALMS21 Task 2	[39]	0.78	0.78	0.78	0.78
PAIRR24M	[38,56]	0.62	0.60	0.63	0.59

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 across species and behavioural contexts (table 1). 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 behaviour to new behavioural 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 and as a social group can include key point coordinates representing important features in the environment as well as connections between multiple individuals within our framework to understand how animals behave in a context-specific and social manner.

While 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 and we additionally provide pre-trained YOLO pose estimation models to facilitate plug-in-and-play pose estimation and behaviour classification for the example species and contexts [22] (electronic supplementary material, figure S3A). The classification models presented here are modified ST-GCN. These networks have clear advantages: 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 behaviour [57]. 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. We further extended this to environments with multiple individuals; however, other key locations and objects could be included in a scene graph to provide more utility for researchers performing complex behavioural 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 behavioural dataset to train and test our supervised ST-GCN models. Previous attempts to extract zebrafish larvae behaviour have used a range of techniques from t-SNE embedding, density-based clustering to FuzzyArt algorithms to effectively discover and describe zebrafish behaviours in a range of experimental conditions [25,26,30]. However, unsupervised clustering, while a powerful and useful technique, does not represent an absolute truth of the number or separation of behaviours and can lead to an unreproducible and variable numbers of clusters depending on the context and parameter selection. We report a division of our swim bouts derived from visual stimuli and predation assays into 30 clusters. In similar experimental set-ups, others have reported a range of distinct swim types from 13 [26] to 36 (18 pairs of mirrored movements) [25]. We created an ST-GCN model to accurately classify the 13 behaviours from [26] using only the angle information of the tail as no (x, y) pose estimation data were present in this dataset. The ability of the ST-GCN model to learn these different behaviours 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 behaviour from pose estimations, and we were unable to train an adequate model to learn the 36 categories of behaviour (18 pairs of mirrored movements) from Johnson *et al*. We do not know whether this was due to the larger number of clusters, a requirement of model parameter optimization or as a consequence of the low temporal resolution (60 fps) of this dataset compared with 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 behaviour evoked by a variety of visual stimuli and prey.

The PoseR toolbox developed here sits within a healthy ecosystem of emerging and maturing behavioural analysis tools that aim to discover behaviours in an unsupervised way or classify behaviours manually. PoseR initially aimed to provide a platform

Table 2. Comparison of PoseR against other behavioural tools. Macro-weighted F1 scores for PoseR models in comparison with reported results from DeepEthogram and TREBA on the open field test, FlyvFly and CALMS21 datasets.

dataset	PoseR	DeepEthogram (medium)	TREBA
open field test	0.84	0.92	—
FlyVFly	0.70	—	0.72
CALMS21 Task 1	0.80	—	0.83
CALMS21 Task 2	0.78	—	0.77

Table 3. Example durations for analysing poses from one video for each dataset model. For each dataset, an example video was analysed to demonstrate typical durations of analysis with a batch size of 16.

model	video duration	no. frames	time taken
ZebLR	40 min @ 330 fps	780 065	33 min
ZebTensor	40 min @ 330 fps	780 065	32.6 min
open field test	10 min @ 25 fps	15 253	36.8 s
CALMS21	2 min @ 30 fps	3632	3.5 s
FlyVFly	28 min @ 30 fps	49 406	18.3 s
PAIRR24M	60 min @ 30 fps	108 000	1 min 52.3 s

Table 4. Example time taken for analysis of a 40 min video containing 877 swim bouts extracted with PoseR. The time taken to extract swim bouts and classify them using PoseR models for long high-frame rate videos.

model	video duration (min)	bouts extracted	time taken (s)
ZebLR	approx. 40	877	8.7
ZebTensor	approx. 40	877	8.29

to efficiently extract zebrafish swim bouts and to train accurate behavioural classifiers to learn the repertoire of zebrafish behaviour. However, we also found that using skeleton-based action recognition in PoseR approached the performance of classic convolutional models such as DeepEthogram in the context of the Sturman mouse open-field test (table 2). Other tools such as BSOID [58] that specialize in open-field behaviour classification achieve accuracies of around 90% on unsupported and supported rearing in bottom-up camera recordings in comparison with 89% and 75% presented here on the top-down Sturman dataset. Recall of groom behaviour was significantly better when using pose estimations of the mouse body alone and performance was substandard when environmental features of the arena were included in the pose graph. It is therefore clear that balancing the inclusion of environmental context without hindering accuracy of non-environmental behavioural is crucial and further refinement in PoseR model architecture and training strategy is required to reach the high standard in this context [15,58]. Notably, we observed comparable or slightly improved performance compared with TREBA (pose + TREBA (w/Task Programming)) [17] when PoseR models were trained on social fly and mouse datasets suggesting that PoseR may be more suited to these complex behaviours and contexts where the connectivity of the pose graph is important in the classification of the behaviour (table 3). Skeleton-based action recognition compared with video-based action recognition requires the additional preprocessing step of pose estimation from models like DeepLabCut and YOLO. This additional step can increase the total processing latency depending on the pose estimation model size and should be noted by users when comparing performance and choosing the appropriate tool for the behavioural context. Whilst pose estimation models and image-based action recognition models both suffer from background noise, skeleton-based action recognition models, as used here, can mitigate the effect of pose estimation errors by including confidence intervals of the pose estimation in the node features to teach the model when coordinates are confident or not. In our specific case, we acquired high frame rate (>300 fps) videos required for capturing zebrafish larvae behaviour 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 could interfere with video flow-based methods of behavioural 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 think the skeleton-based approach is likely to be more optimal than flow-based methods. The field of skeleton-based action recognition is one strand of the action recognition field as a whole and is making rapid progress with the development of novel and more powerful architectures for classifying behaviour [59–61]. It will therefore be interesting to see the comparative performance of these approaches on the contexts presented here. We observed that ST-GCN models for classifying behaviours required relatively short training durations, and propose this is likely due to the previously observed phenomenon of quicker convergence in graph networks [62]. Across the datasets presented here, training duration ranged from 13 min to 3 h; other tools like TREBA and DeepEthogram have reported 24 h of training. By ensuring models are shallow we can

Table 5. Comparison of PoseR model performance versus standard machine learning classifiers on zebrafish datasets. F1 scores for decision tree and naive Bayes skLearn models in comparison with PoseR St-GCN models on the ZebRep (Marques *et al*.) dataset and ZebTensor (this paper) dataset.

model	ZebRep	ZebTensor
naive Bayes	0.51	0.30
decision tree	0.57	0.18
PoseR	0.89	0.77

also achieve fast inference speeds across datasets of between 240 and 435 fps, leading to analysis durations of under 2 min for hour-long recordings at 30 fps (table 3). For zebrafish-specific data, analysing extracted bouts only instead of individual frames reduces analysis durations, for example, for a 40 min video recorded at 330 fps from 33 min to 9 s (table 4). An important aim of neuroscience is to understand how neural activity underlies behaviour and these inference speeds would be sufficient to enable real-time classification of behaviour. Our tool, combined with fast pose estimation [63] and neural activity recording, therefore offers an exciting opportunity to directly correlate *in vivo* neural activity with behaviour, and trigger experimental manipulation with behavioural cues, advancing the effort to understand how the brain produces behaviour.

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 3D convolutional neural networks, and including an RGB video layer [64]. 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 behaviours. 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 behaviour in an unsupervised way using only information contained in a pose graph or knowledge graph of a visual scene in video (table 5).

Ethics. All procedures were carried out according to the UK Animals (Scientific Procedures) Act 1986 and approved by the UK Home Office (licence number PP8440114).

Data accessibility. PoseR is installable via pypi https://pypi.org/project/PoseR-napari/ and hosted on GitHub at https://github.com/pnm4sfix/PoseR. All data are available from the Zenodo repository [65].

Supplementary material is available online at [66].

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Authors' contributions. P.N.M.: conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing—original draft, writing—review and editing; B.B.: formal analysis, investigation; H.C.A.: investigation, resources; A.G.: formal analysis, investigation; M.F.Z.: conceptualization, project administration, resources, supervision, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. This work was supported by the Biotechnology and Biological Sciences Research Council Grant BB/T006560/1, and an RS Macdonald Charitable Trust Grant. Manuscript revisions were supported by an Leverhulme Early Career Fellowship ECF_2022_105.

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 would also like to thank Joe Chapman and technical staff from the Scottish Ocean Institute for zebrafish husbandry support, and the University of St Andrews for computational support via its central HPC facility.

References

1. Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, Bethge M. 2018 DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281–1289. (doi:10.1038/s41593-018-0209-y)
2. Pereira TD *et al*. 2022 SLEAP: a deep learning system for multi-animal pose tracking. *Nat. Methods* **19**, 486–495. (doi:10.1038/s41592-022-01426-1)
3. Berman GJ, Choi DM, Bialek W, Shaevitz JW. 2014 Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**. (doi:10.1098/rsif.2014.0672)
4. Wiltschko AB *et al*. 2020 Revealing the structure of pharmacobehavioral space through motion sequencing. *Nat. Neurosci.* **23**, 1433–1443. (doi:10.1038/s41593-020-00706-3)
5. Wiltschko AB, Johnson MJ, Iurilli G, Peterson RE, Katon JM, Pashkovski SL, Abraira VE, Adams RP, Datta SR. 2015 Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135. (doi:10.1016/j.neuron.2015.11.031)
6. Hsu AI, Yttri EA. 2021 B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nat. Commun.* **12**, 5188. (doi:10.1038/s41467-021-25420-x)
7. Weinreb C *et al*. 2023 Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *bioRxiv* 2023.03.16.532307. (doi:10.1101/2023.03.16.532307)
8. Luxem K, Mocellin P, Fuhrmann F, Kürsch J, Miller SR, Palop JJ, Remy S, Bauer P. 2020 Identifying behavioral structure from deep variational embeddings of animal motion. *Commun. Biol.* **5**, 1–19. (doi:10.1038/s42003-022-04080-7)
9. Weinreb C *et al*. 2024 Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *Nat. Methods* **21**, 1329–1339. (doi:10.1101/2023.03.16.532307)
10. Segalin C, Williams J, Karigo T, Hui M, Zelikowsky M, Sun JJ, Perona P, Anderson DJ, Kennedy A. 2021 The mouse action recognition system (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife* **10**. (doi:10.7554/eLife.63720)

11. Kabra M, Robie AA, Rivera-Alba M, Branson S, Branson K. 2013 JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* **10**, 64–67. (doi:10.1038/nmeth.2281)
12. Nilsson SRO *et al*. 2020 Simple Behavioral Analysis (SimBA)—an open source toolkit for computer classification of complex social behaviors in experimental animals. *bioRxiv*. (doi:10.1101/2020.04.19.049452)
13. Dankert H, Wang L, Hoopfer ED, Anderson DJ, Perona P. 2009 Automated monitoring and analysis of social behavior in *Drosophila*. *Nat. Methods* **6**, 297–303. (doi:10.1038/nmeth.1310)
14. Sturman O *et al*. 2020 Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology* **45**, 1942–1952. (doi:10.1038/s41386-020-0776-y)
15. Bohnslav JP *et al*. 2021 DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife* **10**, e63377. (doi:10.7554/eLife.63377)
16. Tillmann JF, Hsu AI, Schwarz MK, Yttri EA. 2024 A-SOiD, an active-learning platform for expert-guided, data-efficient discovery of behavior. *Nat. Methods* **21**, 703–711. (doi:10.1038/s41592-024-02200-1)
17. Sun JJ, Kennedy A, Zhan E, Anderson DJ, Yue Y, Perona P. 2020 Task programming: learning data efficient behavior representations. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit* **2001**, 2875–2884. (doi:10.1109/CVPR46437.2021.00290)
18. Yan S, Xiong Y, Lin D. 2018 Spatial temporal graph convolutional networks for skeleton-based action recognition. In *32nd AAAI Conf. on Artificial Intelligence*. vol. **32**. Washington, DC: AAAI. (doi:10.1609/aaai.v32i1.12328)
19. Jiménez-Luna J, Grisoni F, Schneider G. 2020 Drug discovery with explainable artificial intelligence. *Nat. Mach. Intell.* **2**, 573–584. (doi:10.1038/s42256-020-00236-4)
20. Jumper J *et al*. 2021 Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589. (doi:10.1038/s41586-021-03819-2)
21. Hamilton WL, Ying R, Leskovec J. 2017 Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* 1025–1035.
22. Jocher G, Qiu J, Chaurasia A. 2023 Ultralytics YOLO (version 8.0.0). *GitHub*. See https://github.com/ultralytics/ultralytics.
23. Sofroniew N *et al*. 2022 . napari: A multi-dimensional image viewer for Python. *Zenodo* https://zenodo.org/record/7276432
24. Kalueff AV *et al*. 2013 Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond. *Zebrafish* **10**, 70–86. (doi:10.1089/zeb.2012.0861)
25. Johnson RE, Linderman S, Panier T, Wee CL, Song E, Herrera KJ, Miller A, Engert F. 2020 Probabilistic models of larval zebrafish behavior reveal structure on many scales. *Curr. Biol.* **30**, 70–82.(doi:10.1016/j.cub.2019.11.026)
26. Marques JC, Lackner S, Félix R, Orger MB. 2018 Structure of the zebrafish locomotor repertoire revealed with unsupervised behavioral clustering. *Curr. Biol.* **28**, 181–195.(doi:10.1016/j.cub.2017.12.002)
27. Semmelhack JL, Donovan JC, Thiele TR, Kuehn E, Laurell E, Baier H. 2014 A dedicated visual pathway for prey detection in larval zebrafish. *eLife* **3**, e04878. (doi:10.7554/elife.04878)
28. Breier B, OnkenA. 2019 Analysis of video feature learning in two-stream CNNs on the example of zebrafish swim bout classification. In *8th Int. Conf. on Learning Representations, ICLR 2020*. https://arxiv.org/abs/1912.09857v1.
29. Palmér T, Ek F, Enqvist O, Olsson R, Åström K, Petersson P. 2017 Action sequencing in the spontaneous swimming behavior of zebrafish larvae—implications for drug development. *Sci. Rep.* **7**, 3191. (doi:10.1038/s41598-017-03144-7)
30. Yang P, Takahashi H, Murase M, Itoh M. 2021 Zebrafish behavior feature recognition using three-dimensional tracking and machine learning. *Sci. Rep.* **11**, 1–9. (doi:10.1038/s41598-021-92854-0)
31. Guilbeault NC, Guerguiev J, Martin M, Tate I, Thiele TR. 2021 BonZeb: open-source, modular software tools for high-resolution zebrafish tracking and analysis. *Sci. Rep.* **11**, 1–21. (doi:10.1038/s41598-021-85896-x)
32. Geng Y, Yates C, Peterson RT. 2023 Social behavioral profiling by unsupervised deep learning reveals a stimulative effect of dopamine D3 agonists on zebrafish sociality. *Cell Rep. Methods* **3**, 100381. (doi:10.1016/j.crmeth.2022.100381)
33. Braun D, Rosenberg AM, Rabaniam E, Haruvi R, Malamud D, Barbara R, Aiznkot T, Levavi-Sivan B, Kawashima T. 2024 High-resolution tracking of unconfined zebrafish behavior reveals stimulatory and anxiolytic effects of psilocybin. *Mol. Psychiatry* **29**, 1046–1062. (doi:10.1038/s41380-023-02391-7)
34. Rubinstein Y, Moshkovitz M, Ottenheimer I, Shapira S, Tiomkin S, Avitan L. 2025 A detailed quantification of larval zebrafish behavioral repertoire uncovers principles of hunting behavior. *iScience* **28**, 112213. (doi:10.1016/j.isci.2025.112213)
35. Reddy G, Desban L, Tanaka H, Roussel J, Mirat O, Wyart C. 2022 A lexical approach for identifying behavioural action sequences. *PLoS Comput. Biol.* **18**, e1009672. (doi:10.1371/journal.pcbi.1009672)
36. Mearns DS, Donovan JC, Fernandes AM, Semmelhack JL, Baier H. 2020 Deconstructing hunting behavior reveals a tightly coupled stimulus–response loop. *Curr. Biol.* **30**, 54–69.(doi:10.1016/j.cub.2019.11.022)
37. Eyjolfsdottir E, Branson S, Burgos-Artizzu XP, Hoopfer ED, Schor J, Anderson DJ, Perona P. 2024 Fly v. Fly Dataset. *CaltechDATA*. See https://data.caltech.edu/records/zrznw-w7386.
38. Marshall J, Dunn T, Aldarondo D, Ölveczky BP, Gellis AJ, Klibaite U. 2024 PAIRS_dataset. *Figshare*. See https://figshare.com/articles/dataset/PAIRS_dataset/14754374.
39. Sun JJ. 2021 The multi-agent behavior dataset: mouse dyadic social interactions. *Arxiv* 2104.02710. (doi:10.48550/arXiv.2104.02710)
40. Lopes G *et al*. 2015 Bonsai: an event-based framework for processing and controlling data streams. *Front. Neuroinformatics* **9**, 7. (doi:10.3389/fninf.2015.00007)
41. Paszke A *et al*. 2019 Chanan GPyTorch: an imperative style, high-performance deep learning library. *arXiv* 1912.01703. (doi:10.48550/arXiv.1912.01703)
42. Lightning AI. 2025 PyTorch Lightning: The deep learning framework to pretrain and finetune AI models. See https://github.com/Lightning-AI/pytorch-lightning.
43. Virtanen P *et al*. 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272. (doi:10.1038/s41592-019-0686-2)
44. Kolda TG, Bader BW. 2009 Tensor decompositions and applications. *SIAM Rev.* **51**, 455–500. (doi:10.1137/07070111x)
45. Williams AH, Kim TH, Wang F, Vyas S, Ryu SI, Shenoy KV, Schnitzer M, Kolda TG, Ganguli S. 2018 Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. *Neuron* **98**, 1099–1115. (doi:10.1016/j.neuron.2018.05.015)
46. neurostatslab/tensortools. 2023 A very simple and barebones tensor decomposition library for CP decomposition a.k.a. PARAFAC a.k.a. TCA. *GitHub*. See https://github.com/neurostatslab/tensortools.
47. Davies DL, Bouldin DW. 1979 A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1**, 224–227. (doi:10.1109/tpami.1979.4766909)
48. Rousseeuw PJ. 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65. (doi:10.1016/0377-0427(87)90125-7)
49. OpenMMLAB. 2023 MMSkeleton: a OpenMMLAB toolbox for human pose estimation, skeleton-based action recognition, and action synthesis. See https://github.com/open-mmlab/mmskeleton.
50. Fero K, Yokogawa T, Burgess HA. 2011 The behavioral repertoire of larval zebrafish. In *Neuromethods* (eds AV Kalueff, JM Cachat), pp. 249–291. Springer Nature. (doi:10.1007/978-1-60761-922-2_12)
51. Healy J, McInnes L. 2024 Uniform manifold approximation and projection. *Nat. Rev. Methods Prim* **4**, 82. (doi:10.1038/s43586-024-00363-x)

52. Budick SA, O'Malley DM. 2000 Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture. *J. Exp. Biol.* **203**, 2565–2579. (doi:10.1242/jeb.203.17.2565)
53. Borla MA, Palecek B, Budick S, O'Malley DM. 2002 Prey capture by larval zebrafish: evidence for fine axial motor control. *Brain Behav. Evol.* **60**, 207–229. (doi:10.1159/000066699)
54. Patterson BW, Abraham AO, MacIver MA, McLean DL. 2013 Visually guided gradation of prey capture movements in larval zebrafish. *J. Exp. Biol.* **216**, 3071–3083. (doi:10.1242/jeb.087742)
55. Orger M, Mendeley Data. 2018 Structure of the zebrafish locomotor repertoire revealed with unsupervised behavioural clustering Marques. *Mendeley Data*. See https://data.mendeley.com/datasets/r9vn7x287r/1.
56. Marshall JD, Klibaite U, Gellis A, Aldarondo DE, Ölveczky BP, Dunn TW. 2021 The PAIR-R24M dataset for multi-animal 3D pose estimation. *bioRxiv*. (doi:10.1101/2021.11.23.469743)
57. 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. *Appl. Anim. Behav. Sci.* **249**, 105594. (doi:10.1016/j.applanim.2022.105594)
58. Hsu AI, Yttri EA. 2021 B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nat. Commun.* **12**, 1–13. (doi:10.1038/s41467-021-25420-x)
59. Cheng K, Zhang Y, He X, Chen W, Cheng J, Lu H. 2020 Skeleton-based action recognition with shift graph convolutional network. In *2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, pp. 180–189. (doi:10.1109/CVPR42600.2020.00026)
60. Liu J, Wang X, Wang C, Gao Y, Liu M. 2024 Temporal decoupling graph convolutional network for skeleton-based gesture recognition. *IEEE Trans. Multimed.* **26**, 811–823. (doi:10.1109/tmm.2023.3271811)
61. Liu J, Yin B, Lin J, Wen J, Li Y, Liu M. 2024 HDBN: a novel hybrid dual-branch network for robust skeleton-based action recognition. In *2024 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, Niagara Falls, ON, Canada. (doi:10.1109/ICMEW63481.2024.10645450)
62. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. *arXiv* 1609.02907. doi:10.48550/arXiv.1609.02907
63. Kane GA, Lopes G, Saunders JL, Mathis A, Mathis MW. 2020 Real-time, low-latency closed-loop feedback using markerless posture tracking. *eLife* **9**, 1–29. (doi:10.7554/elife.61909)
64. Duan H, Zhao Y, Chen K, Lin D, Dai B. 2022 Revisiting skeleton-based action recognition. In *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, pp. 2959–2968. (doi:10.1109/CVPR52688.2022.00298)
65. Mullen P, Bowlby B, Armstrong H, Zwart M. 2023 Datasets of larval zebrafish behaviour, Mullen *et al*., 2023 (0.0.1) [Data set]. *Zenodo*. (doi:10.5281/zenodo.7807968)
66. Mullen PN, Bowlby B, Armstrong HC, Gray A, Zwart MF. 2025 Supplementary material from: PoseR: a deep learning toolbox for classifying animal behavior. Figshare. (doi:10.6084/m9.figshare.c.8168694)