

Automated neuron tracking inside moving and deforming *C. elegans* using deep learning and targeted augmentation

In the format provided by the
authors and unedited

Supplementary Results

Tracking different *C. elegans* strains without hyper-parameter tuning

To test Targettrack on different constellations of fluorescent labels with the same set of hyper-parameters (Methods), we annotated 10 images to track 13 neurons as key-points in the nerve ring of a panneuronal mNeptune/GCaMP strain (ADS1041, Supplementary Figure 7). We also tested Targettrack in key-point mode further with strains with substantially different labeling geometries and patterns (Supplementary Figure 7): ADS27 and SJR17 contain nuclear localized fluorophores as well as cytosolic fluorophores in thin neurites. ADS1018 has fluorophores in the intestine lacking the ovoid appearance of neurons. Targettrack generalized well to track complex shapes beyond ovoid cell bodies.

Tracking animals using training sets based on other individuals

We tested whether the 3DCN could accurately track neurons in one individual but using training data from a set of different worms with the same fluorescent neuronal labels (Supplementary Figure 8). We demonstrate this capability by annotating 23 neurons in 5 frames for 5 recordings of hermaphrodite heads, then proofreading the tracking results of taCNN(3DCN) for an average of 8 frames per dataset. We then trained a generalist network on a total of 61 frames (5 annotations per dataset + proofread frames). We achieved a tracking accuracy > 94% with neurons in 21 other animals of the same genotype (Supplementary Figure 8 A, B). We confirmed that the tracks indicate that our worms are not overly restricted (Supplementary Figure 8 C) and that the extracted neural signals from these animals captured meaningful neural activity (Supplementary Figure 8 D).

The AVA neuron plays a critical role in regulating forward and backward movement³⁵, where increases in AVA calcium dynamics are correlated with the onset of backward movement^{36,37}. Our analysis of AVA confirms these previous calcium imaging findings (Supplementary Figure 9 A).

We reproduced previous findings of the experience-dependent thermosensory response of the AFD neuron (Supplementary Figure 9 B)^{38–40}. We confirm a previous analysis predicting worm velocity from brain-wide neural activity (Supplementary Figure 9 C)¹⁰. When we attempted to predict the activity of individual neurons based on the activities of all other neurons, we found that the activities of motor neurons and command interneurons can be better predicted than sensory neurons (Supplementary Figure 9 D).

Comparisons with other methods

We compared our 3DCN architecture with 3D U-Net²⁴ for key-point and volumetric tracking (Fig. 2 C, G). We found substantially higher accuracies with 3DCN using small training sets (smaller than 10). This advantage disappears for larger training sets (25 and above), as expected.

Deep neural networks have been used to perform point-set registration when tracking neurons by training the networks to find the most likely alignment of 3D images. Deep neural networks have also been used to learn the most likely trajectory in collective motion tracking^{18,19}.

fDNC¹⁹ and EMC2¹⁷ require key-point representations of neurons, which is difficult to reconcile with neuronal shapes that are not ovoid. Similarly, 3DeeCellTracker¹⁸ requires each neuron to be separable via a watershed transform, effectively restricting the method to ovoid neurons.

We compared segmentation and tracking by taCNN with Coherent Point Drift (CPD) point-set registration^{28,29}. CPD²⁸ tracks neurons by finding the optimal correspondence between neuron locations in two images. Point-set registration requires fully segmented neurons in all image volumes, whereas taCNN performs both segmentation and tracking. To specifically compare the ability of these algorithms to track neurons, we provided the CPD²⁸ algorithm with GT key-points (i.e. already segmented key-points) from our brain-wide imaging datasets of male mating behavior. We also used our posture space embedding (which we obtained from our neural network experiments) to augment the performance of the CPD²⁸ method by only asking it to track neurons across maximally similar image volumes (**Supplementary Note 4**). As when we compared the iCNN and taCNN performances, we selected a reference image set with known labels for the CPD²⁸ to generate matches. The larger the reference image set, the better CPD²⁸ should work – similar to the way in which increasing the size of a training set improves a convolutional neural network – and we confirmed this trend (Supplementary Figure 5 A, B). However, we found that CPD²⁸

accurately tracked less than 50% of neurons with reference image set sizes where the taCNN accurately tracked over 80% (Table 1, Supplementary Figure 5 A, B).

Pretraining with other data

We tested whether pre-training with annotated worm images from other laboratories improves the performance of our method. Pre-training the 3DCN first on annotated images from ref.¹⁵, and then training it with the initial training set or with targeted augmentation applied, we found no discernible benefits of pre-training (Supplementary Figure 5 C).

Using the GUI

In order to make viewing the recordings convenient, image color, gamma value, as well as the minimum and maximum of the color ranges can be adjusted interactively. The user moves through the z-dimension at a single timepoint by scrolling with the scroll wheel of the computer mouse, and moves through time using the keyboard. The GUI is optimized to annotate 3D images with a time dimension. A key-point is added by clicking on the appropriate location on a 2D slice with the computer mouse. 3D volumes can be drawn with cubic pencils or a 'bucket' tool, inspired by similar methods found in 2D image editing software. One can also enable a functionality to automatically register points to the local image maximum, relevant to tracking blobs. Key-points can be highlighted and their location traces in time can be visualized. For calcium imaging purposes, we also integrate a functionality within the GUI to visualize the computed calcium signal – thus, the user can efficiently find frames where the calcium signal may be corrupted.

Supplementary Note 1: Selective cross entropy loss

We use a loss function which allows neurons to remain un-annotated in some frames if they are outside the field of view or too dim to be annotated. The cross entropy loss L_{CE} and the selective cross entropy loss L_{sCE} are defined as

$$l_{CE}(x, y)_i = -\log \left(\frac{\exp(x_{i,y_i})}{\sum_{c=1}^C \exp(x_{i,c})} \right) \quad L_{CE}(x, y) = \frac{\sum_{i=1}^{N_{vox}} l_{CE}(x, y)_i}{N_{vox}} \quad (S1)$$

$$l_{sCE}(x, y)_i = -\log \left(\frac{\exp(x_{i,y_i})}{\sum_{c=1; c \in \text{unique}(y)}^C \exp(x_{i,c})} \right) \quad L_{sCE}(x, y) = \frac{\sum_{i=1}^{N_{vox}} l_{sCE}(x, y)_i}{N_{vox}} \quad (S2)$$

where N_{vox} is the number of voxels, $x_{i,c}$ is the logit value given to the class c at voxel i , y_i is the ground truth class of the voxel i , and C is the number of classes. In our case, $N_{vox} = \text{Width} \times \text{Height} \times \text{Depth}$ and $C = (\# \text{ of objects}) + 1$. $\text{unique}(y)$ returns the set of classes present in the current instance y .

Supplementary Note 2: Iteratively selecting target images

We use the following process to select target images:

1. For a dataset with frames indices $0 \leq i < T$
2. Begin with GT , the set of ground truth frame indices, $D \in \mathbb{R}^{T \times T}$ the symmetric matrix of pairwise posture distances and $targets = []$.
3. While $\text{numel}(targets) < N_{additional}$
 - (a) Find the minimum posture distance over all GT and already added target images for all non-GT images.
 $MD_j = \min_{i \in GT \cup targets} D(i, j)$
 - (b) Find the non-GT or target image for which this minimum distance is maximized.
 $j_{add} = \text{argmax}_{j \notin GT \cup targets} (MD_j)$
 - (c) $targets \leftarrow j_{add}$

Supplementary Note 3: Calculating the entropy of each frame

We can use the raw output of the neural network as logits to compute an uncertainty value for all frames. These are frames more probable to be ones where the neural network has failed to produce an accurate tracking. The uncertainty values are calculated as:

1. Let NN be the trained neural network seen as a function transforming the frames into prediction logits

$$NN : \mathbb{R}^{(C \times W \times H \times D)} \rightarrow \mathbb{R}^{N_{points} \times W \times H \times D}$$
2. For each frame $F \in \mathbb{R}^{(C \times W \times H \times D)}$:
 - (a) Compute the logits for every voxel

$$L = NN(F)$$
 - (b) Compute the entropy value for every voxel

$$S_{ijk} = \sum_n \exp(L_{nijk})$$
 - (c) Average over all voxels to get the frame uncertainty

$$U = \frac{1}{W \times H \times D} \sum_{ijk} S_{ijk}$$

Supplementary Note 4: Details of the CPD method

We use CPD (Coherent Point Drift)²⁸ together with a nearest neighbor matching scheme as a simple tracking method for comparison. We assume a perfectly solved segmentation problem using the ground truth pointset from manual annotation. The algorithm is as follows:

1. We begin with N_{GT} annotated pointsets, i.e., points are provided with labels. We also use the posture distance matrix $D \in \mathbb{R}^{T \times T}$
2. For each non-annotated pointset, $P_i \in \mathbb{R}^{N_{points}, 3}$:
 - (a) The annotated pointset nearest to P_i in the posture space is found and called P_i^{ref}

$$j = \operatorname{argmin}_j D(i, j) \quad P_i^{ref} = P_j$$
 - (b) P_i is matched to P_i^{ref} using different versions of CPD²⁸. The resulting, deformed P_i , is called P_i^d

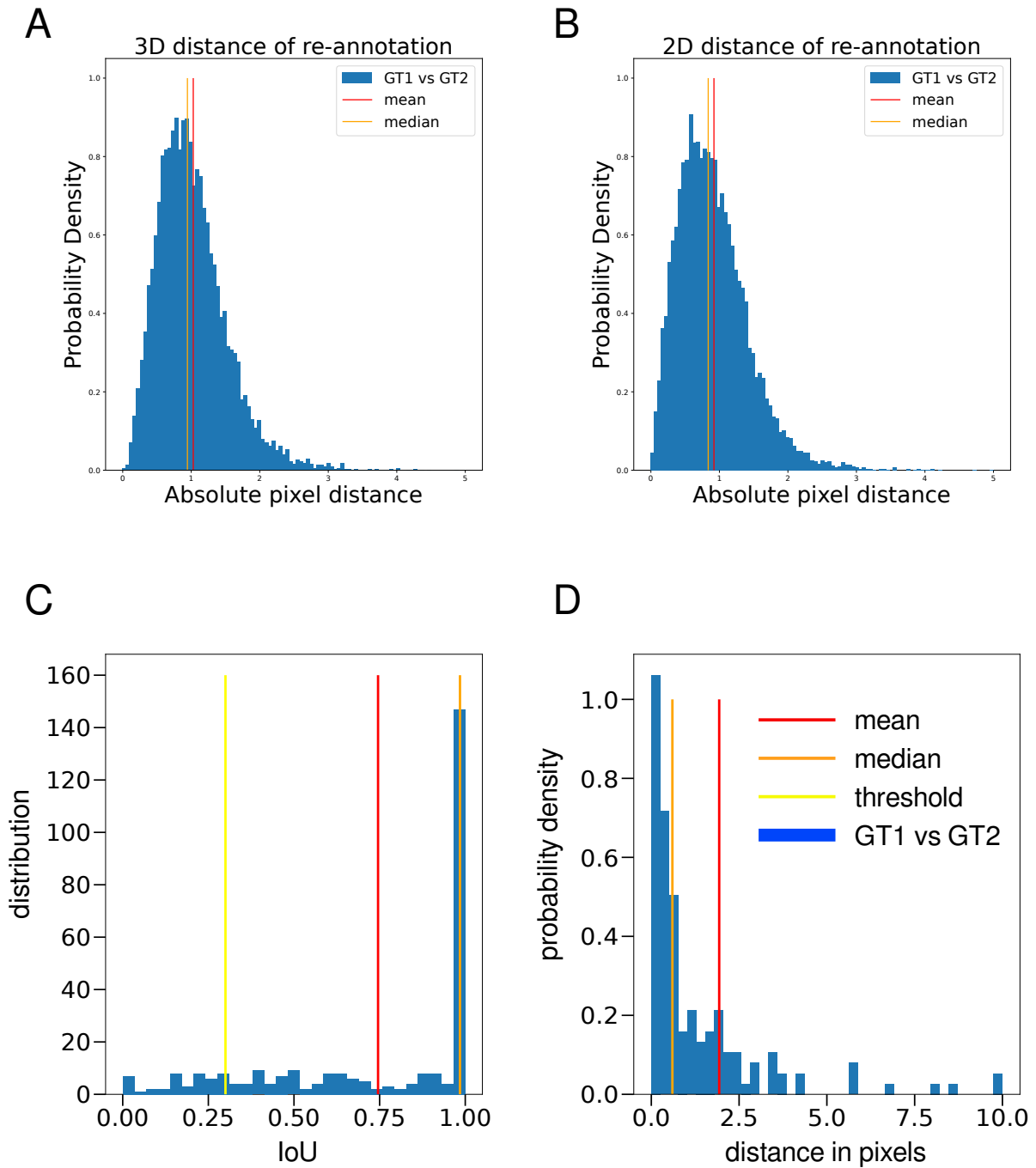
$$P_i^d = CPD(P_i)$$
 - (c) Each point in P_i^d gets assigned the label of the nearest point in P_i^{ref}
 - (d) The labels are propagated to the corresponding point in P_i .

Supplementary Tables

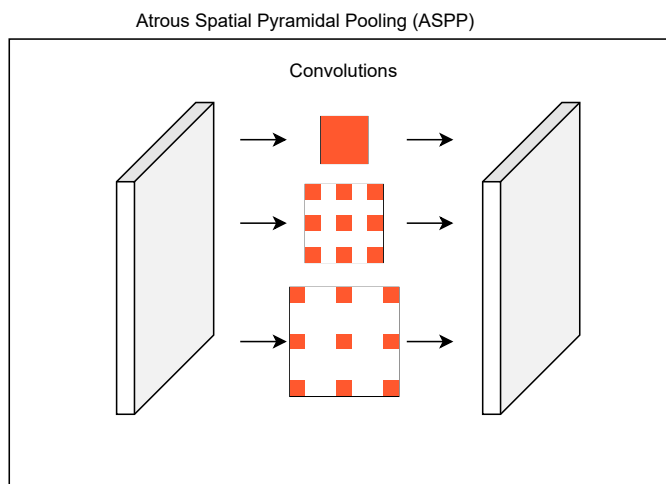
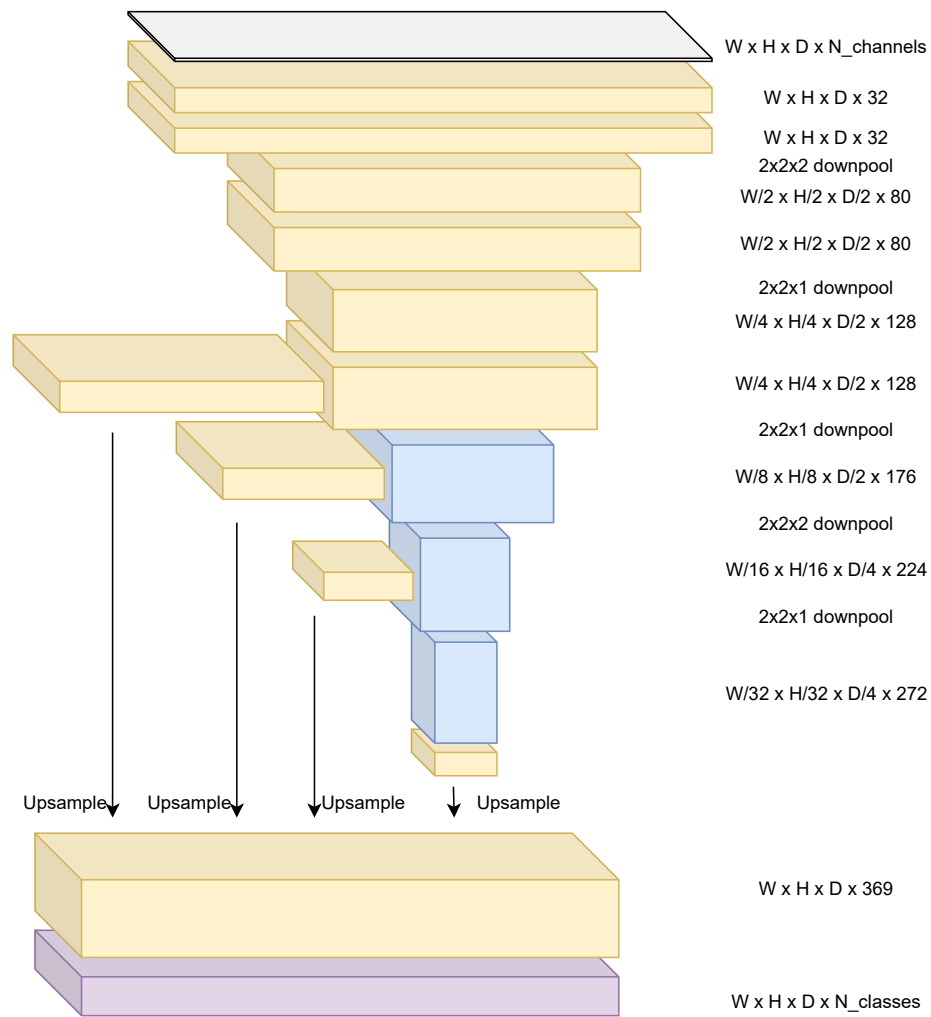
	N_{GT}					
	5	10	15	25	35	50
p values for taCNN vs. iCNN in Fig. 2 A (n=20 for 4 recordings and 5 random GT annotation samples)	1.5e-05	3.9e-07	3.0e-08	2.1e-06	1.2e-06	2.4e-03
p values for taCNN vs. iCNN in Fig. 2 B (n=20 for 4 recordings and 5 random GT annotation samples)	3.4e-11	4.6e-10	1.4e-11	5.0e-15	5.7e-12	3.1e-09
p values for 3DCN vs. 3D U-Net in Fig. 2 C with targeted augmentation (n=4 for 4 recordings)	1.0e-03	1.6e-03	9.6e-03	1.5e-02	1.9e-02	2.5e-02
p values for taCNN vs. iCNN in Fig. 2 E (n=18 for 3 recordings and 6 random GT annotation samples)	1.7e-12	8.1e-10		4.1e-05	6.6e-03	1.2e-01
p values for taCNN vs. iCNN in Fig. 2 F (n=18 for 3 recordings and 6 random GT annotation samples)	3.7e-06	3.1e-09		1.0e-02	2.7e-01	7.1e-01
p values for 3DCN vs. 3D U-Net in Fig. 2 G with targeted augmentation (n=3 for 3 recordings)	9.6e-02	1.3e-01		1.1e-01		6.8e-01

Supplementary Table 1: p values for plots in Fig. 2 based on the one-tailed t-test.

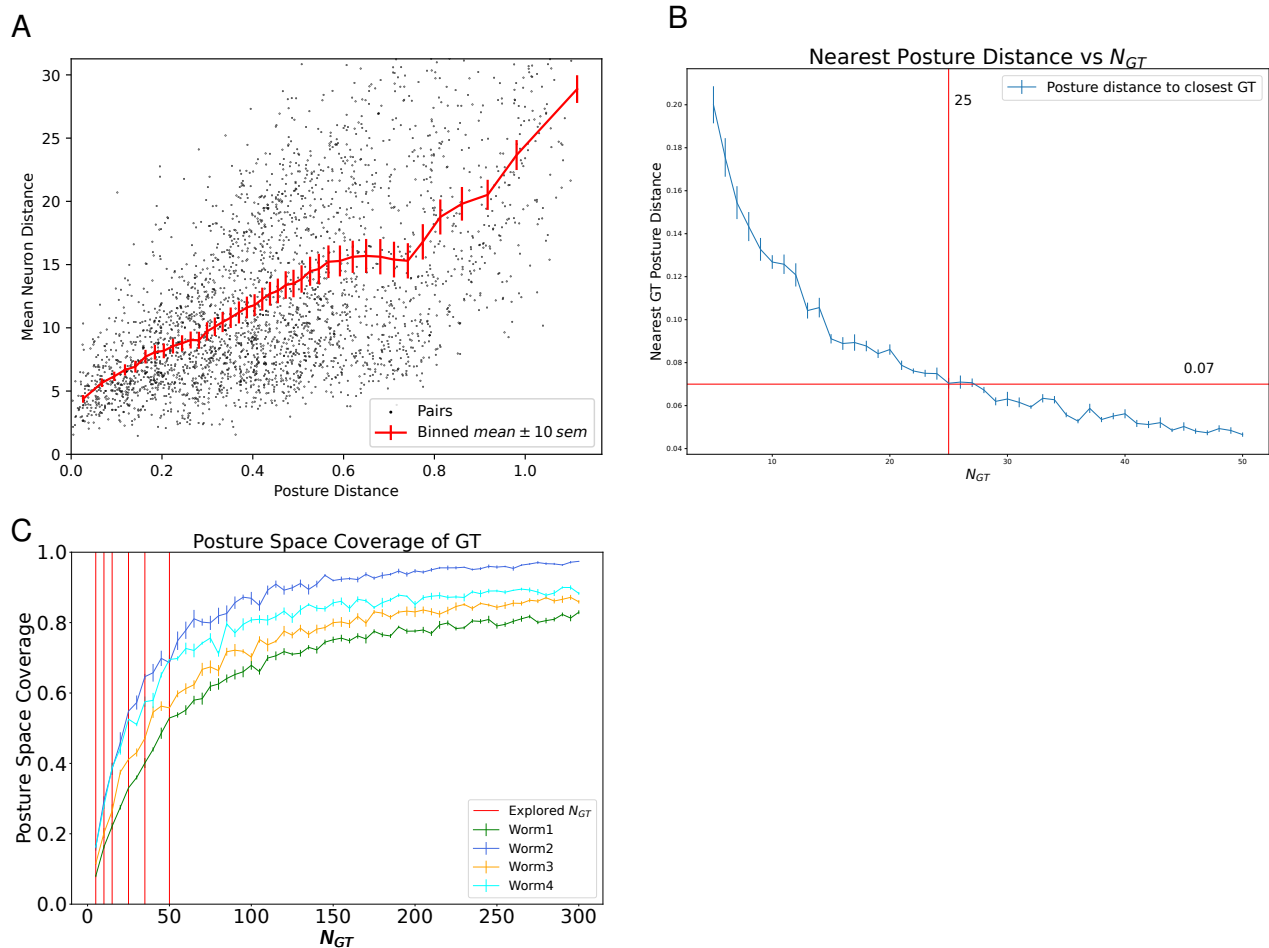
Supplementary Figures



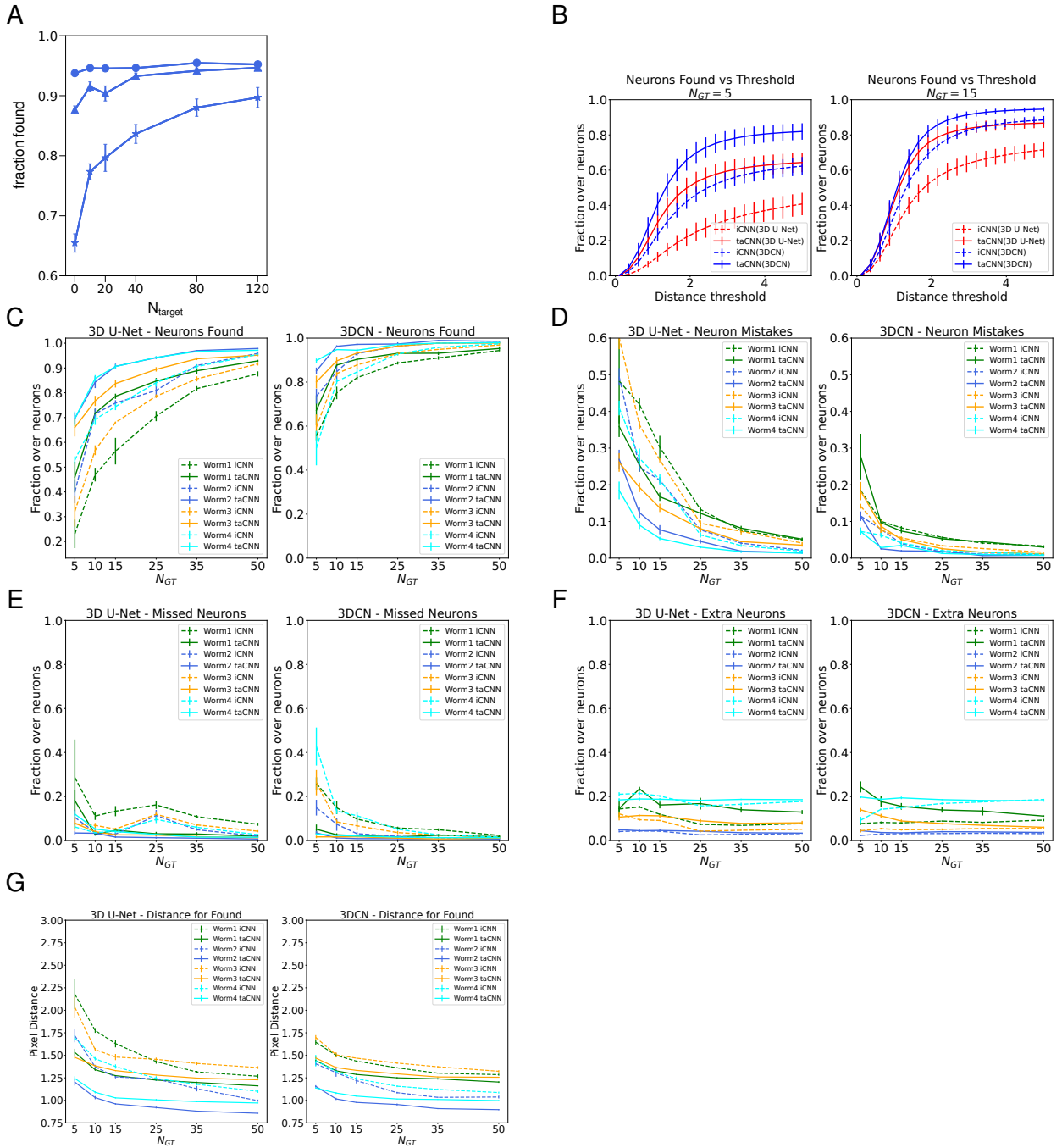
Supplementary Figure 1: **Reliability of manual annotations.** **A, B.** Precision of re-annotation of 10256 points from 23 neurons and 500 frames. The first annotation was not visible to the human annotator for the re-labeling. **C, D.** Reliability of manual 3D volume annotations. 25 images encompassing 293 neurons as 3D volumes were annotated twice from scratch manually. In panel C, the IoU distribution of the annotations for the same objects is shown. The yellow line is the IoU threshold used for defining found cells in the manuscript. In panel D, the distribution of 3D distances between the centers of the corresponding 3D volumes is shown.



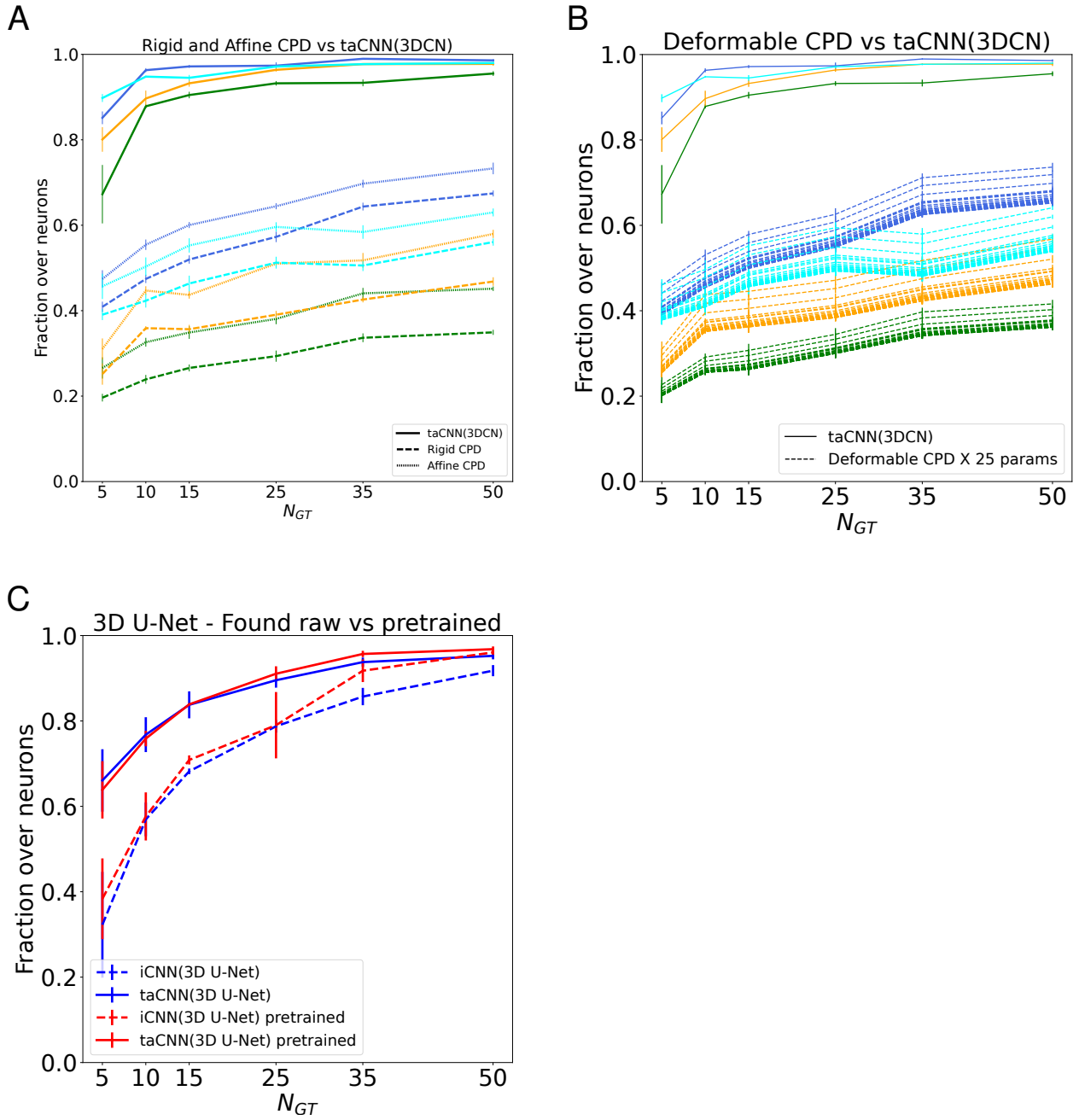
Supplementary Figure 2: Illustration of the 3DCN neural network architecture and of the atrous convolutions.



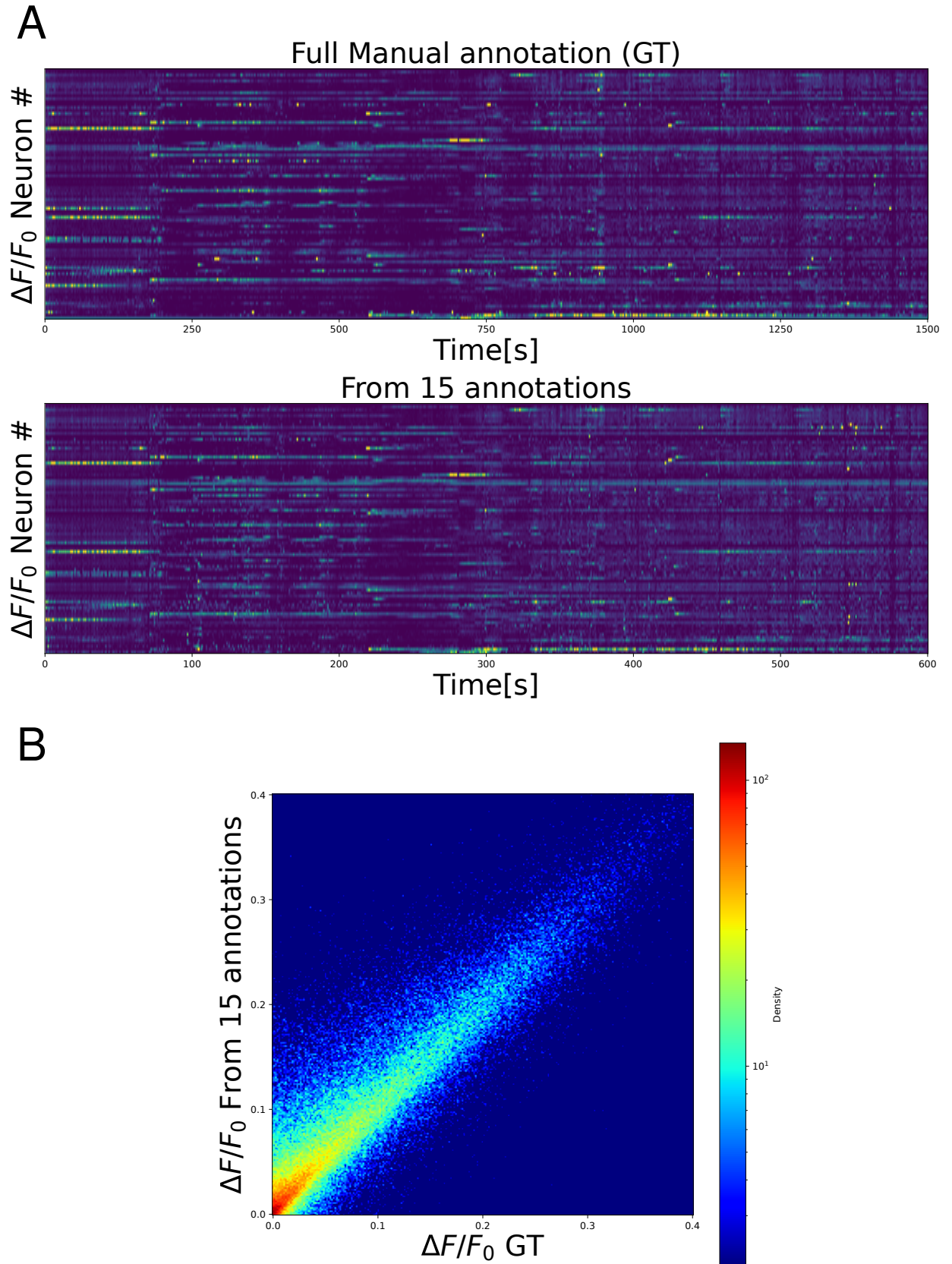
Supplementary Figure 3: **Analysis of posture space.** **A.** Correlation between real-space neuronal distances (y-axis) and posture space distances (x-axis). Using the full ground truth annotations of the recordings at our disposal, we calculated the real-space Euclidean distances between the same neurons in two 3D images from the same recording (minus the mean displacement between all neurons in the two images). These distances, averaged over all neurons in each pair of images, are represented by the vertical coordinate. The horizontal coordinate represents the Euclidean distance in posture space. Each point in the scatter plot represents a pair of two 3D images from different timepoints. The red line and error bars are binned averages and 10 times the standard error of the mean, respectively. Note that the ground truth annotations are not needed for mapping images to the posture space in the Targettrack pipeline but were used here for computing the actual distances between pairs of neurons. One recording consisting of 3002 image volumes of one worm was analyzed here. The images were filtered so that the same 10 neurons were in the field of view in at least 100 images. The scatter plot shows a random selection of 3000 out of 143341 pairs of filtered images. **B.** The capability of 3DCN to generalize quantified in terms of posture space distances. We picked random subsets of GT images, standing in for random selections of images annotated by the user. Then, we calculated the posture space distance of each image in the recording to this subset as a function of the number of images in the subset. The data points show the average of these minimum distances versus the sizes of the subsets. The error bars are standard errors of the mean over random sampling the GT images ($n=10$). The vertical red line at $N_{GT} = 25$ indicates the number of GT images and annotations where the iCNN-3DCN achieved a high accuracy (Fig. 2). The corresponding posture space distance (0.07) is used in panel C as the distance over which the neural network generalizes well. The underlying data is the same as in panel A. **C.** Number of GT images needed to cover posture space versus needed to saturate iCNN accuracy. For the plots Worm1, Worm2, Worm3, and Worm4, we analyzed the coverage of the posture space as a function of the number of ground truth images used. We generated the posture space, then randomly sampled ground truth images, and evaluated the area of the posture space covered by the GT frames. We used a radius of 0.07 and computed the area of posture space covered by GT images by adding the unique area of circles of radius 0.07 around each GT point. The total area of posture space for all images was similarly computed. For easier comparison with Fig. 2, six red vertical lines are shown, which correspond to the set of N_{GT} used to evaluate iCNN and taCNN in Fig. 2 A-C, $N_{GT} = 5, 10, 15, 25, 35, 50$. The plots represent the same data as colored in Fig. 2 A, B. The error bars represent the standard error of the mean when sampling different random ground truths ($n=5$).



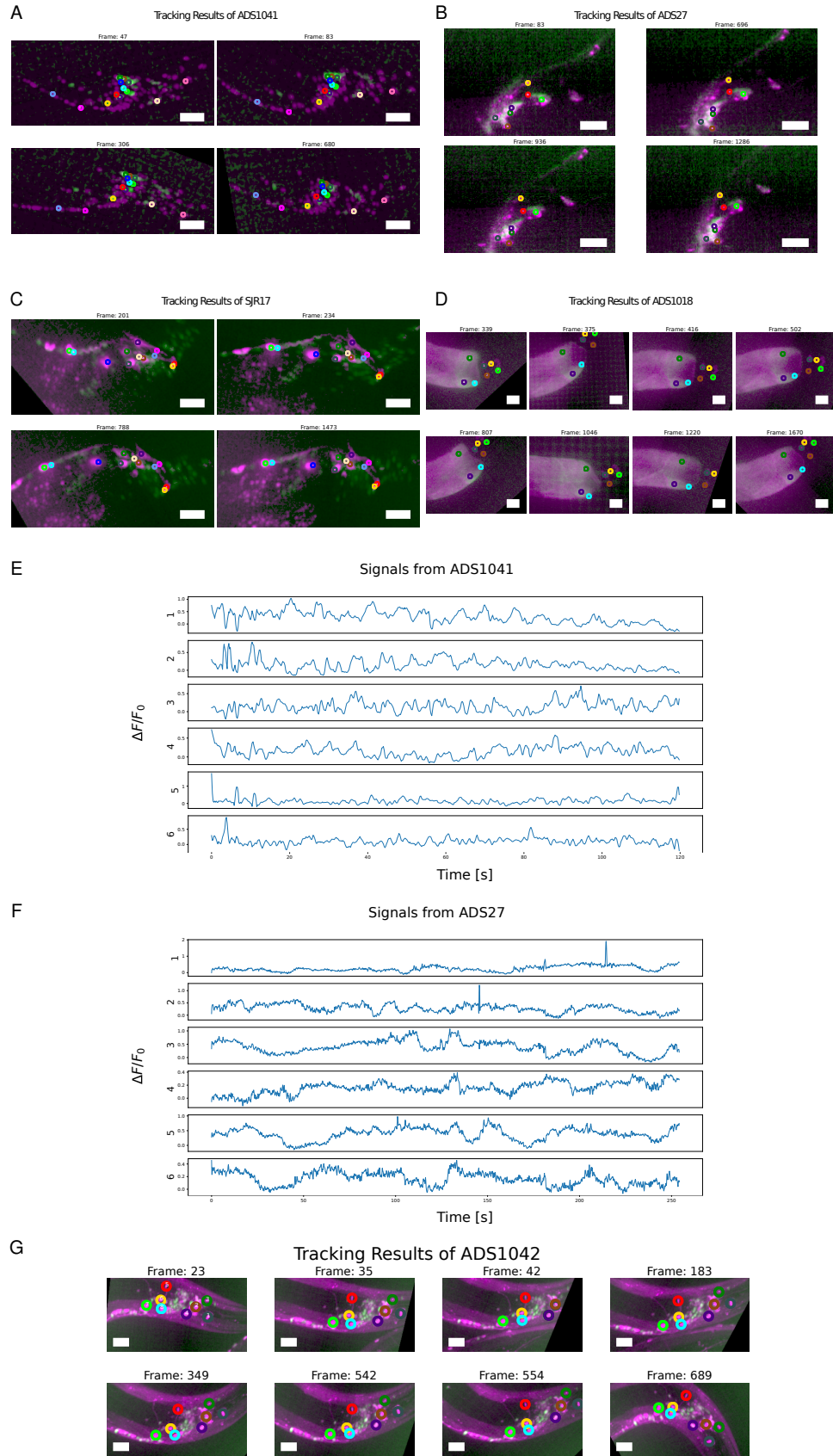
Supplementary Figure 4: Performance of the taCNN as a function of target images, dependence on the distance threshold between GT and predicted key-points, and comparison with 3D U-Net. **A.** Fraction of objects found for one recording (blue plot in Fig. 2 E, F) with $N_{GT} = 5$ (stars), $N_{GT} = 15$ (triangles), or $N_{GT} = 25$ (circles) ground-truth annotations as a function of the target set size N_{target} . For each data point, ground truth annotations were sampled $n=6$ times. Error bars represent the standard error of the mean (SEM). **B.** Fraction of key-points found depending on the pixel distance threshold from the GT annotation used. The y-axis values represent the fractions found, averaged over 5 random GT samplings for the 4 recordings. The error bars indicate the standard error of the mean over different recordings ($n=4$) for $N_{GT} = 5$ (left) and for $N_{GT} = 15$ (right). **C.** Fraction of neurons successfully found by (left) 3D U-Net²⁴ and (right) 3DCN when applying TA or not, depending on the number of ground truth frames used. The error bars are the standard error of the mean estimated by resampling the ground truth images ($n=5$). The different datasets are not averaged together due to the high variance between datasets compared to when selecting different ground truth frames. **D.** Fraction of neurons detected at a wrong location by (left) 3D U-Net²⁴ and (right) 3DCN when applying TA or not, depending on the number of ground truth frames used. The error bars are the standard error of the mean estimated by resampling the ground truth images ($n=5$). **E.** Fraction of neurons missed (not reported at all) by (left) 3D U-Net²⁴ and (right) 3DCN when applying TA or not, depending on the number of ground truth frames used. The error bars are the standard error of the mean estimated by resampling the ground truth images ($n=5$). **F.** Fraction of extra neurons found by (left) 3D U-Net²⁴ and (right) 3DCN when applying TA or not, depending on the number of ground truth frames used. Since the ground truth human annotation has neurons which have not been annotated due to low brightness or simply by mistake, these detections may be actual neurons. The error bars are standard error of the mean estimated by resampling the ground truth images ($n=5$). **G.** Distance to ground truth annotation for the found neurons for (left) 3D U-Net²⁴ and (right) 3DCN when applying TA or not, depending on the number of ground truth frames used. The error bars are standard error of the mean estimated by resampling the ground truth images ($n=5$).



Supplementary Figure 5: **Comparison of Targettrack to CPD and 3D U-Net.** **A.** Performance of CPD²⁸ on our 4 male *C. elegans* tail recordings. We used the CPD²⁸ implemented by pyCPD²⁹. The affine CPD allows more degrees of freedom than rigid CPD. The error bars represent the standard error of the mean estimated by resampling the ground truth images ($n=5$) for each recording. **B.** Performance of deformable CPD²⁸ on our 4 male *C. elegans* tail recordings. We used the CPD²⁸ implemented by pyCPD²⁹. We applied deformable CPD with parameters from all combinations of $\alpha \in [0.5, 1.875, 3.25, 4.625, 6.]$ and $\beta \in [0.5, 1.875, 3.25, 4.625, 6.]$. The error bars represent the standard error of the mean estimated by resampling the ground truth images ($n=5$) for each recording. **C.** Performance of 3D U-Net²⁴ as a function of the size of the training set and of pretraining. 3D U-Net was pre-trained to predict neuron regions versus background without tracking each neuron individually. The 3D U-Net was pre-trained on all frames from 4 male tails and 2 worms from the NERVE dataset, including the test frames. The trained 3D U-Net models were tested on one worm recording, and the error bars represent the standard error of the mean estimated by resampling the ground truth images ($n=5$).

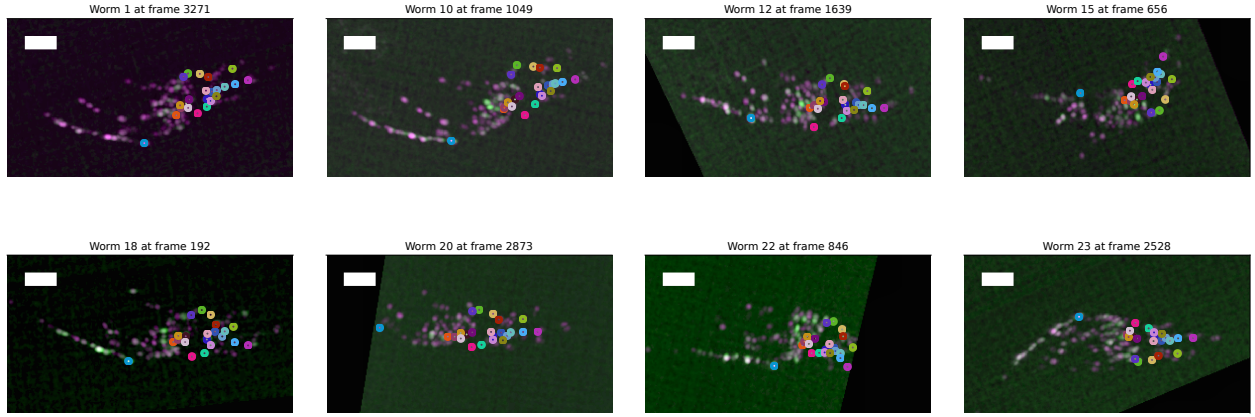


Supplementary Figure 6: **Evaluation of the result of running Targettrack with 15 annotated images.** **A.** Neural activity of a male tail reported as $\Delta F/F_0$. (Top) signal from manual annotation of 3002 frames, (bottom) the signal using neurons tracked with taCNN but not proofread using 15 annotated images. The signal has not been filtered or smoothed. **B.** Neural activity extracted using a neural network trained with 15 annotated images versus ground-truth. Density heatmap pooling all neuron activities at all time points. Note the log color scale.

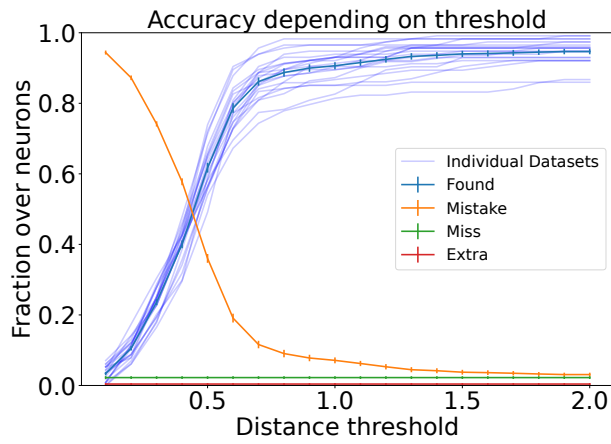


Supplementary Figure 7: **Tracking different *C. elegans* strains with no hyper-parameter changes.** All shown frames are test images that were not included in the training set. Scalebar: 20 μm . **A.** Tracking head neurons of a pan-neuronal mNeptune/GCaMP strain. (835 frames, $N_{GT} = 10$) **B.** Tracking key-points in a strain with cytosolic RFP/GCaMP. (1590 frames, $N_{GT} = 9$) **C.** Tracking cell nuclei and neurites simultaneously as key-points. (1711 frames, $N_{GT} = 10$) **D.** Tracking different key-points in the worm's intestine. (1870 frames, $N_{GT} = 9$) **E.** The GCaMP F/F_0 signal extracted from the ADS1041 strain in panel A. We plot signals from raw predictions without any proofreading. **F.** The GCaMP F/F_0 signal extracted from the ADS27 strain in panel B. We plot signals from raw predictions without manual proofreading and correction. **G.** Tracking the head neurons of a *dop-3::RFP* and panneuronal mNeptune/GCaMP strain with different imaging settings. 897 frames, $N_{GT} = 10$.

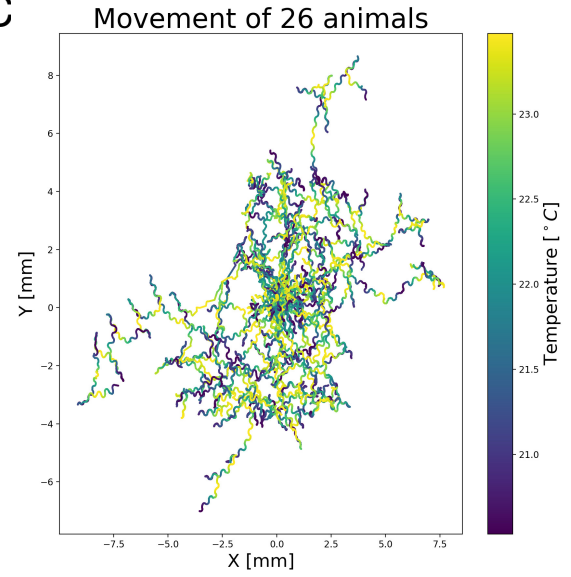
A



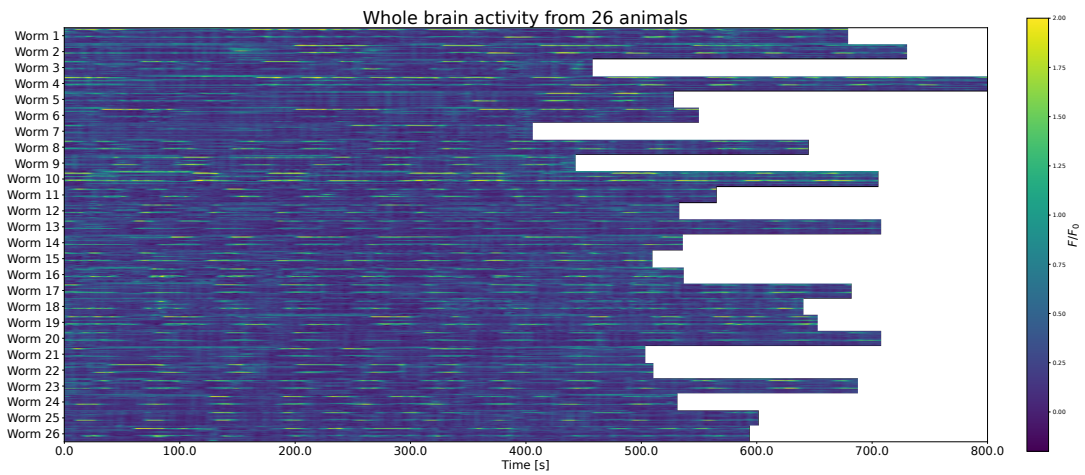
B



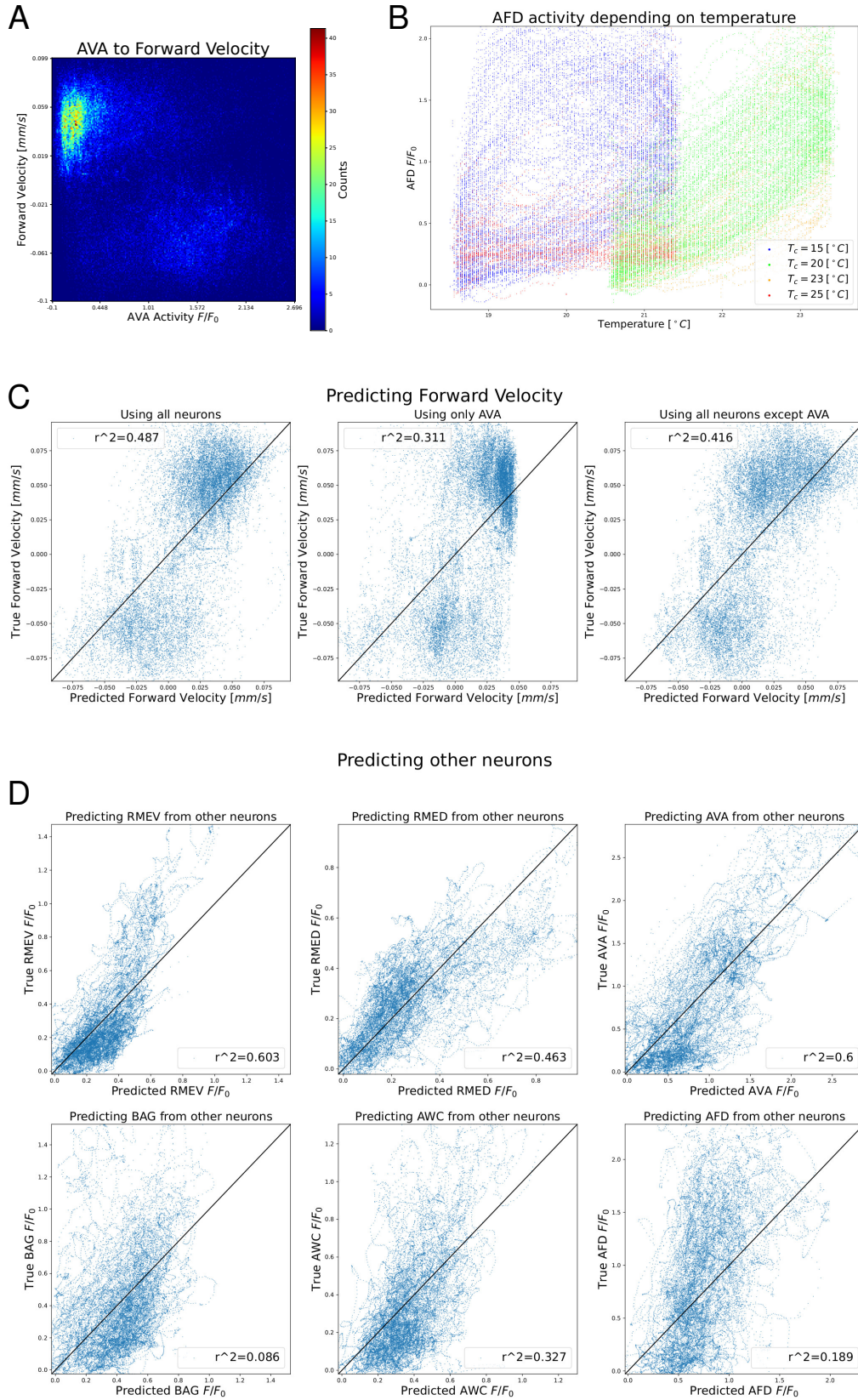
C



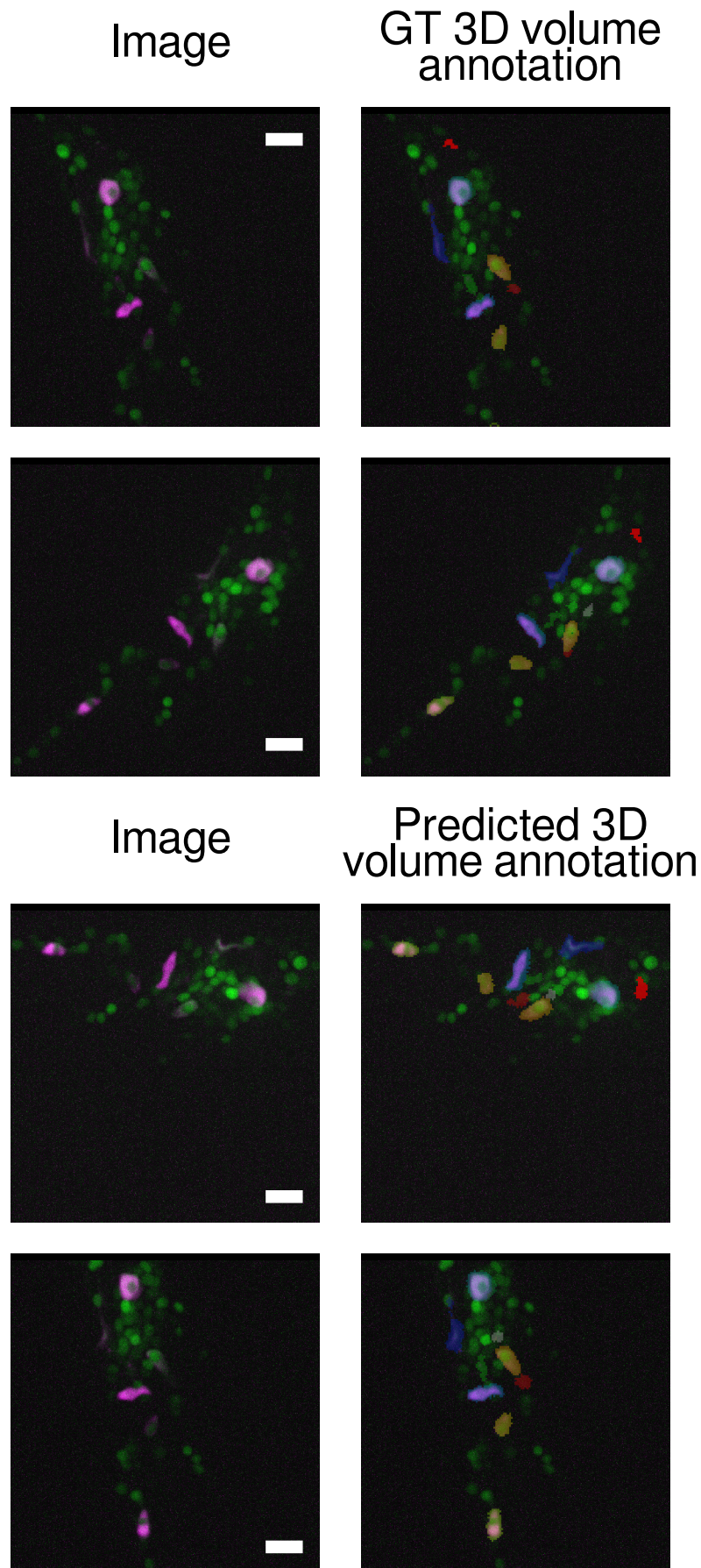
D



Supplementary Figure 8: **Tracking individual worms based on annotations of other worm recordings.** We analyzed Targettrack on 26 pan-neuronal GCaMP/mNeptune strain (ZM9624) hermaphrodite worms using a training set of 61 ground truth annotations from 5 of the individuals and no manual correction of Targettrack annotations. **A.** We show one frame per animal for 8 animals out of 21 animals which have been tracked fully automatically. Scalebar: 20 μm . **B.** Tracking accuracy for 23 neurons in the 21 head recordings which were not in the training set. We evaluated the accuracy by manually annotating 5 random frames in each of the 21 datasets. We plot the fraction of found, mistaken, missed, and extra neurons. The error bars denote the standard error of the mean over the unannotated datasets ($n=21$). **C.** Head coordinates of the 26 animals. We align all the starting head points at $(x, y) = (0, 0)$. The tracks are colored by the temperature experience of the worm at each location. The worms are experiencing a sinusoidal temperature change of period 60 seconds and amplitude 1°C after an initial phase of constant baseline temperature. **D.** Neural activity of 23 identical matching neurons from the 26 animals.



Supplementary Figure 9: **Reproducing known functional properties of *C. elegans* neural activity using activity from Targettrack.** Tracking and segmentations were obtained as in Supplementary Figure 8. **A.** Density heat map of the Forward Velocity and AVA activity.⁴¹ **B.** Plot of AFD neuron activity depending on temperature.^{38–40} **C.** To predict the worm's forward velocity using neural activity, we randomly selected 20 worms from the set described in Supplementary Figure 8 and trained a linear regression model.¹⁰ Then, we evaluated its performance on 6 held-out worms. **D.** Predicting a neuron's activity from other neurons. Similarly to panel C, we fit a linear regression model on 20 randomly selected worms described in Supplementary Figure 8, then evaluated on 6 held-out worms. Each plot shows the result of predicting a neuron's activity from 33 other neurons in the *C. elegans* head.



Supplementary Figure 10: **Sample images and annotations from a recording with a worm (strain SJR15) with cytosolic fluorescent labels.** Left column shows the original recordings as maximum intensity projections along the z-axis (optical axis). The right column shows either the corresponding GT annotations (first two rows) or typical predicted annotations of our pipeline (last two rows). Here, we trained the taCNN(3DCN) with 15 GT images and with targeted augmentation. Note that in the bottom row, not all neurons of interest were visible in the field of view. Scalebar: 10 μm .

Supplementary References

35. Meng, J., Ahamed, T., Yu, B., Hung, W., Mouridi, S. E., Wang, Z., Zhang, Y., Wen, Q., Boulin, T., Gao, S. & Zhen, M. A tonically active neuron continuously drives mutually exclusive motor states at different timescales. *bioRxiv* (2022).
36. Kawano, T., Po, M. D., Gao, S., Leung, G., Ryu, W. S. & Zhen, M. An imbalancing act: Gap junctions reduce the backward motor circuit activity to bias *C. elegans* for forward locomotion. *Neuron* **72**, 572–586 (2011).
37. Chronis, N., Zimmer, M. & Bargmann, C. I. Microfluidics for in vivo imaging of neuronal and behavioral activity in *Caenorhabditis elegans*. *Nature Methods* **4**, 727–731 (2007).
38. Mori, I. & Ohshima, Y. Neural regulation of thermotaxis in *Caenorhabditis elegans*. *Nature* **376**, 344–348 (1995).
39. Clark, D. A., Biron, D., Sengupta, P. & Samuel, A. D. T. The AFD sensory neurons encode multiple functions underlying thermotactic behavior in *Caenorhabditis elegans*. *Journal of Neuroscience* **26**, 7444–7451 (2006).
40. Goodman, M. B. & Sengupta, P. The extraordinary AFD thermosensor of *C. elegans*. *Pflügers Archiv-European Journal of Physiology* **470**, 839–849 (2018).
41. Gray, J. M., Hill, J. J. & Bargmann, C. I. A circuit for navigation in *Caenorhabditis elegans*. *Proc. Natl. Acad. Sci. U. S. A.* **102**, 3184–3191 (2005).