Article

Patterns

Segmentation tracking and clustering system enables accurate multi-animal tracking of social behaviors

Highlights

- segTracker.ai: an unsupervised SOTA multi-animal pose-tracking algorithm
- segCluster: a weakly supervised clustering algorithm for animal social behaviors
- STCS reveals links between social stress and motor impairments in ASD mice

Authors

Cheng Tang, Yang Zhou, Shuaizhu Zhao, ..., Youming Lu, Guangzhi Ma, Hao Li

Correspondence

maguangzhi@hust.edu.cn (G.M.), lihaochn@hust.edu.cn (H.L.)

In brief

Leveraging the latest in computer vision, this study introduces a comprehensive framework for automated, quantitative, and objective analysis of animal social behaviors through advanced pose estimation and multi-animal tracking. Addressing the challenges of reliable long-term tracking and the interpretation of social interactions influenced by genetic variations, this work offers a novel approach for computational ethology, facilitating accessible and collaborative research across teams via the Internet.

Tang et al., 2024, Patterns *5*, 101057
November 8, 2024
https://doi.org/10.1016/j.patter.2024.101057

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Segmentation tracking and clustering system enables accurate multi-animal tracking of social behaviors

Cheng Tang,[1,2,6] Yang Zhou,[1,4,6] Shuaizhu Zhao,[1,4] Mingshu Xie,[1,4] Ruizhe Zhang,[5] Xiaoyan Long,[5] Lingqiang Zhu,[1,4] Youming Lu,[1,4] Guangzhi Ma,[3,*] and Hao Li[1,4,7,*]

[1]Innovation Center of Brain Medical Sciences, the Ministry of Education, China, Huazhong University of Science and Technology, Wuhan 430022, China
[2]Department of Nuclear Medicine, Wuhan Union Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430022, China
[3]School of Computer Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[4]Department of Pathophysiology, School of Basic Medicine and Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430030, China
[5]Wuhan Union Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430022, China
[6]These authors contributed equally
[7]Lead contact
*Correspondence: maguangzhi@hust.edu.cn (G.M.), lihaochn@hust.edu.cn (H.L.)
https://doi.org/10.1016/j.patter.2024.101057

THE BIGGER PICTURE In modern animal social experimentation, digital cameras combined with advanced computer vision technologies can be used for animal pose estimation, to track animals as they move about their enclosures, and to recognize and quantify different types of behaviors. However, persistent challenges remain: reliable 2D unsupervised tracking of animals under complex social conditions is unresolved, and interpreting social behaviors from pose and tracking data is notably difficult. The computational framework described in this work offers an integrated solution to these challenges under weakly supervised conditions, bridging computational science with biomedical research.

SUMMARY

Accurate analysis of social behaviors in animals is hindered by methodological challenges. Here, we develop a segmentation tracking and clustering system (STCS) to address two major challenges in computational neuroethology: reliable multi-animal tracking and pose estimation under complex interaction conditions and providing interpretable insights into social differences guided by genotype information. We established a comprehensive, long-term, multi-animal-tracking dataset across various experimental settings. Benchmarking STCS against state-of-the-art tracking algorithms, we demonstrated its superior efficacy in analyzing behavioral experiments and establishing a robust tracking baseline. By analyzing the behavior of mice with autism spectrum disorder (ASD) using a novel weakly supervised clustering method under both solitary and social conditions, STCS reveals potential links between social stress and motor impairments. Benefiting from its modular and web-based design, STCS allows researchers to easily integrate the latest computer vision methods, enabling comprehensive behavior analysis services over the Internet, even from a single laptop.

INTRODUCTION

Understanding the intricate nature of social behavior[1] is crucial in the field of neuroscience, as it offers profound insights into how organisms interact, communicate, and form relationships within their social milieu. In this context, studying social behavior in animals serves as a pivotal bridge between molecular[2] and systems-level neuroscience.[3] Investigations using rodent models, in particular, offer a valuable means to dissect the neural circuits and molecular underpinnings governing social interactions. Autism spectrum disorders (ASDs) are a heterogeneous group of complex neurodevelopmental disorders characterized by impairments in social interaction, communication, and restricted and repetitive behaviors.[4,5] Traditional assessments, such as

the three-chamber test,[6] often occur in controlled environments, potentially oversimplifying the complexity of natural interactions.[7] This limitation is especially pertinent in conditions marked by social deficits, such as ASD.[8,9]

A variety of deep-learning-based methods for multi-animal behavior analysis are available,[9,10] including DeepLabCut (DLC)[11] and SLEAP,[12] which facilitate simultaneous animal pose estimation and tracking. However, these tools have a number of limitations, including unreliable identity (ID) switches[13] and the requirement for thousands of labeled images to visually distinguish animals like SIPEC.[14] Common workarounds involve tracking a distinctive part of the animal for identification or pre-recording individual activity videos for each animal to establish individual information. However, the former fails to fully utilize the animal's complete visual representation,[15] and the latter introduces additional workload and extraneous variables into the animal experiments.[16] Although idTracker.ai[17] offers a solution by achieving extremely high tracking accuracy with little intervention, automatically identifying visually similar animals—a task beyond the capacity of average human performance—it is computationally intensive and struggles with highly deformable and socially active animals in complex environments.[11,18] Beyond tracking, analyzing complex social-behavior patterns is also challenging in behavioral neuroscience. Existing approaches, often manual and rule based, are laborious and require tailored sets of rules for different animals or specific conditions. While methods such as those proposed in DLC and other systems[19–21] offer post-processing of results, they frequently fall short of capturing the full spectrum of social behaviors. These gaps underscore the necessity for developing more advanced analytical tools.

Recognizing these challenges and limitations in behavioral studies, we developed a segmented tracking and clustering system (STCS) for multi-animal behavioral analysis. STCS comprises two novel components: segTracker.ai and segCluster. SegTracker.ai is an animal-, model-, and device-agnostic unsupervised pose-tracking protocol that enables tracking of social behaviors in complex settings with different combinations of cutting-edge deep-learning methods. SegCluster, on the other hand, is a weakly supervised social-behavior clustering protocol that utilizes a novel autoencoder neural network based on the spatiotemporal graph convolutional network (STGCN)[22,23] with an innovative social convolution (SC) block. This combination not only quantifies behavior traits with unprecedented precision but also delves deeper into analyzing social-behavior patterns. Application of STCS in *Shank3* knockout mice, a model for ASD,[24,25] enabled identification of nine distinct interactive behavioral patterns. By comparing the locomotive characteristics of solitary mice versus those in groups, our data revealed potential mechanisms underlying the emergence of these varied interaction patterns. Thus, STCS is able to capture subtle behavioral differences across different phenotypes.

RESULTS

Development of an STCS

We developed an STCS by introducing a transformative approach to monitor animal's social behaviors with the incorporation of two innovative components: segTracker.ai for tracking and segCluster for analysis. Inspired by DLC[11] and idTracker.ai,[17] we divided the multi-animal tracking and pose estimations into several sub-tasks: animal detection, segmentation, and pose estimation per frame. We then generated local tracklets and associated these tracklets to form complete tracks for animals (Figure 1A). To conquer these sub-tasks, we introduced four deep-learning modules: the detection and segmentation model (Figure 1B), the optical-flow model (Figure 1C), the pose-estimation model (Figure 1D), and the classification/re-ID model (Figure 1E). The detection and segmentation model generated bounding boxes for top-down pose estimation and instance-segmentation masks for generating tracklets. The optical-flow model estimated the motion between adjacent frames to perform mask warping for better matching. Tracklets were merged using various strategies and then used to train the classifier, which subsequently identified the remaining tracklets iteratively until tracking was complete. Together, this modular approach enables our comprehensive and accurate tracking of multiple animals, handling complex behaviors and interactions.

An integral component of STCS was dedicated to the nuanced differentiation and clustering of lab animals' social behaviors. To leverage rich genotype or group data for comprehensive, unbiased social-behavior analysis, we employed a multi-task autoencoder based on the spatiotemporal graph convolutional network (STGCN). This autoencoder features a reconstruction head for capturing latent behavioral patterns, complemented by a classification head that integrates genotype information. Post training and evaluation, we extracted behavioral embeddings from the autoencoder's bottleneck, facilitating dimensionality reduction and clustering as depicted in Figure 1F. However, STGCN in its original form was not tailored for social-behavior recognition as it amalgamates individual data only at the last fully connected layer. To adapt STGCN for social-behavior studies while preserving its skeletal graph structure, we innovated by introducing an SC module into the original graph convolutional network (GCN) block, thereby creating the social GCN block. This module aggregates data from previous stages, gathering inputs from neighboring key points defined in a skeletal structure, comprising three crucial components (Figure 1G):

(1) Grouped Conv1d: processes data from multiple subjects simultaneously, capturing their spatial interactions.
(2) BatchNorm: normalizes data across multiple subjects, enhancing spatial interaction capture.
(3) ReLU activation: adds non-linearity, enabling the model to learn complex behaviors.

By grouping the same key points across different individuals, the SC module enhanced the data flow of interaction data throughout the network stages. Our experiments demonstrated its efficacy in improving classification tasks across diverse scenarios, including varying animal numbers, input features,[26] and experimental settings (Figure S1).

To democratize the use of STCS, especially for researchers with limited computational expertise, we developed a user-friendly browser/server architecture. This architecture includes a streamlined graphical user interface on the front end, supported by multiple work queues on the back end (Figure 1H, I). Notably, segTracker.ai, a core component of STCS, can be

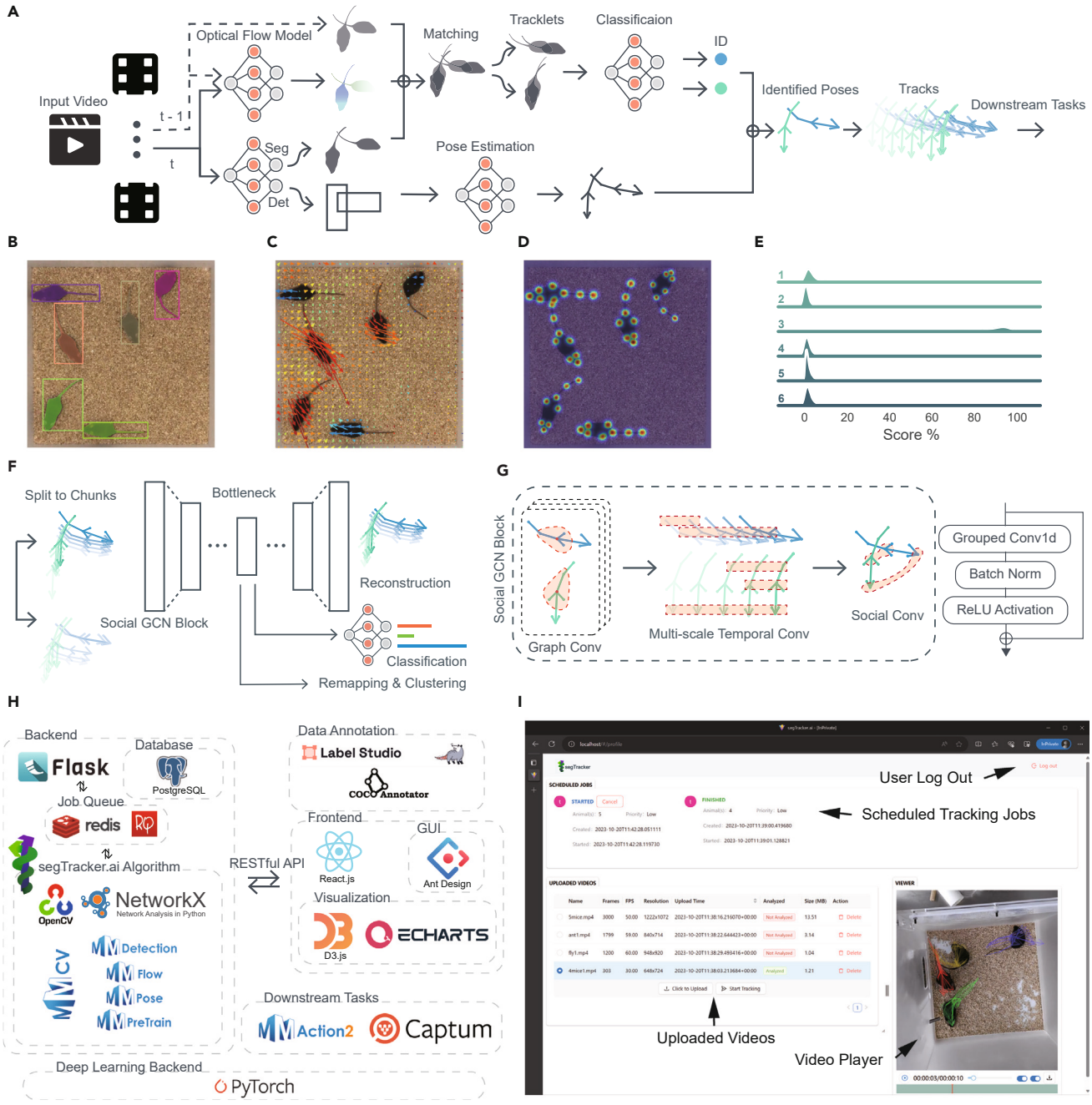

Figure 1. Comprehensive workflow of STCS

(A) Schematic representation of the STCS framework, integrating segTracker.ai for expert deep-learning-based animal-detection, segmentation, optical-flow-estimation, pose-estimation, and identification tasks.

(B) Visual output from the animal segmentation and detection model, showcasing the precision in identifying individual subjects within a complex environment.

(C) Results from the optical-flow-estimation model, illustrating the dynamic flow and movement trajectories of the subjects across sequential frames.

(D) Outputs from the pose-estimation model, delineating the postural positioning of each subject, critical for detailed behavioral analysis.

(E) Classification model results, providing accurate identification of individual subjects, an essential step for longitudinal behavioral studies.

(F) Processed pose sequences from segTracker.ai, segmented into data chunks for input into segCluster's social STGCN—a multi-task autoencoder designed for reconstructing pose data and classifying social interactions, further refined through UMAP and agglomerative clustering for nuanced behavior pattern analysis.

(G) The Social GCN Block, enhancing the original GCN Block for improved integration and interpretation of social-behavior data.

(H) Architectural layout of STCS, illustrating its browser/server (B/S) design with RESTful API support, optimized for seamless deployment via Docker, facilitating collaborative research via the Internet.

(I) User-friendly interface of STCS, accessible through the Edge browser, demonstrating the system's ease of use for end users.

conveniently deployed on a standard personal laptop or PC with a modest GPU, yet robustly delivers comprehensive lab animal tracking services to entire research groups via the Internet.

Diverse datasets compilation for detection, instance segmentation, pose estimation, and multi-animal tracking

In developing robust systems for detection, instance segmentation, pose estimation, and long-term animal tracking, we compiled diverse datasets crucial for training and validating our models. We manually annotated video frames of mouse behavior obtained from various experimental setups and recording devices to train the instance-segmentation and pose-estimation models (Figure 2A). Additionally, a smaller number of frames from idTracker.ai featuring ants and fruit flies (about 20–30 frames) were labeled, ensuring generalizability across different video types without specific optimizations.

Unlike existing methods such as DLC or AlphaTracker, which may require fine-tuning for specific video types to achieve high-quality pose-estimation outputs, our strategy employs a single, versatile model designed to operate across a diverse array of video settings. This approach reduces the need for multiple specialized models in varied real-world applications.

On the other hand, long-term tracking of lab animals is crucial for evaluating the efficacy of tracking and identification algorithms.[27] In previous studies, experimental animal tracking algorithms could only be validated on short-term datasets ranging from tens of seconds to a few minutes, or a few hundred to a few thousand frames.[11,12,28–31] To our knowledge, there are no open-source datasets for long-term tracking of multiple rodents publicly available. The complexity of animal interactions in ethological experiment videos, combined with varying video lengths, experimental setups, and the number of animals involved, poses significant challenges for tracking methods. Notably, an increase in the number of animals does *not* necessarily correlate with increased tracking difficulty in the videos.[31] For instance, Jiang et al.[28] found that tracking a few deformable, occluding mice poses a greater challenge than tracking a larger number of locusts. Therefore, constructing a publicly accessible dataset is essential for impartially testing different tracking algorithms.

We compiled an extensive multi-animal tracking dataset in the multiple object tracking (MOT) format,[32] derived from real-world social-behavior experiments, and employed occlusion metrics[31] to quantitatively describe the complexity of our tracking dataset. This dataset encompassed a diverse array of challenges, such as a 1-h video featuring four wild-type (WT) mice (4×WT) with a single labeled instance, exhibiting complex behaviors such as occlusion, chasing, and fighting over 108,003 frames. The demanding nature of this video's length and complexity underscores the necessity for precise and consistent identification and tracking. Further diversifying the dataset, we included videos such as six-mouse WT (6×WT), two-mouse WT (2×WT) in complex settings, low-resolution four-mouse WT (Noldus), and four-mouse Parkinson's disease model (4×PD) videos to test the system under varied frame rates and video qualities. Additionally, the dataset is enriched with 14-ant and 10-fly videos from idTracker.ai, enhancing its comprehensiveness for smaller species (Figures 2A; Table 1). To evaluate the robustness of our tracking algorithm in minimal movement scenarios, we also provide a video where mice are stationary or asleep (Note S1).

Development of deep-learning models and benchmarking against state-of-the-art (SOTA) tracking methods

We trained multiple distinct deep-learning models using annotated data of mice, ants, and fruit flies for performing instance-segmentation and pose-estimation tasks. Preliminary evaluations on the validation and test datasets (Figure 2B) demonstrated the robustness and accuracy of our models without requiring specific fine-tuning.

Accurate identification of individual animals during extended tracking sessions is crucial for behavioral tracing and analysis systems. We assessed various convolutional neural network (CNN) models, including idTracker.ai's idCNN,[17] DenseNet121[33] (used by SIPEC), MobileNetV2,[34] and OSNet,[35] for their ability to recognize similar lab animals within a subset of the 4×WT dataset. OSNet, pretrained on ImageNet, outperformed others, showing nuanced identification capabilities (Figure 2C). LayerCAM[36] analysis revealed that OSNet's class activation maps (CAMs) were sharply focused, particularly effective at identifying unique markings on mice, in contrast to the more diffuse maps from idTracker.ai (Figure 2D). This was further supported by a confusion matrix analysis, illustrating OSNet's superior accuracy in classifying visually distinct and similar mice (Figure 2E).

On the other hand, segTracker.ai, a core algorithm in our system, leverages deep models but remains independent of specific models or algorithms. It supports integration with various deep-learning architectures, allowing researchers to customize their tracking and analysis strategies. We showcase segTracker.ai's model-agnostic feature with two distinct configurations (Figures 2A; Table 2):

(1) We use Mask Scoring RCNN[37] for detection and instance segmentation due to its capability to automatically evaluate mask quality. YOLOv8 Pose[38] is employed as a single-stage pose-estimation method for its balance of accuracy and efficiency, particularly advantageous in crowded scenes. For motion estimation, we select the Farneback optical-flow-estimation algorithm,[39] a classical dense optical-flow method not based on deep learning. Finally, OSNet[35] is chosen as the re-identification model for its superior identification performance in our experiments.

(2) This is a combination purely based on deep-learning models. We employ RTMDet-tiny-Ins[40] as the detection and instance-segmentation model, currently a state-of-the-art method for real-time instance segmentation. MobileNetV2,[34] combined with a top-down pose-estimation approach,[41] is used for pose estimation, offering higher pose-estimation accuracy. LiteFlowNetV2,[42] a lightweight network for optical-flow estimation, achieves a balance of speed and precision. We again choose OSNet as the re-identification model.

Subsequently, we conducted a thorough comparison of segTracker.ai using the aforementioned trained models against nine other SOTA tracking systems. These systems span a wide range

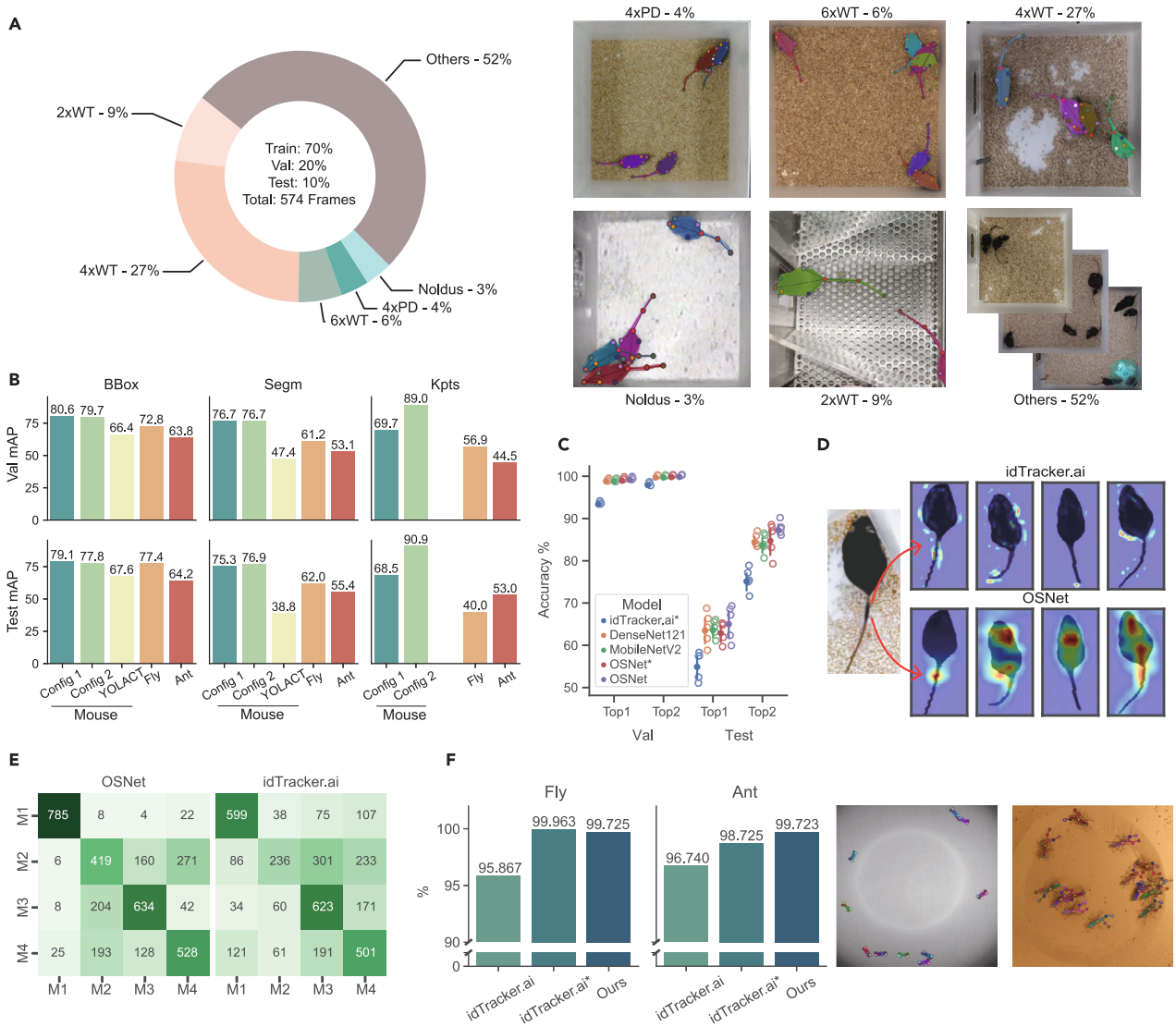

Figure 2. Dataset construction and performance metrics of segTracker.ai

(A) The mouse annotation dataset for instance segmentation and pose estimation comprises videos from various neurobiology experiments. Over half of the images are sourced from videos outside the tracking dataset. On the right are representative snapshots of the five tracking videos as well as some other video snapshots that constitute the instance-segmentation and pose-estimation dataset.

(B) Benchmarking results showing the efficacy of expert models for bounding box detection (BBox), segmentation (Segm), and key point (Kpts) accuracy across different animal species. Config 1, Mask Scoring RCNN as the detector, YOLOv8-s Pose for single-stage pose estimation. Config 2, RTMDet-Ins-tiny as the detector, MobileNetV2 for top-down pose estimation. The ant and fruit fly datasets were trained using Config 2.

(C) Comparative identification accuracies for four different animals within the 4×WT dataset, evaluated using networks from idTracker.ai, DenseNet121 (employed in SIPEC), MobileNetV2, and OSNet. The accuracies are reported as top1 and top2 for both validation and test datasets, with error bars indicating a 95% confidence interval (CI).

(D) LayerCAM visualizations comparing the focus of the final CNN layer between the idTracker.ai network and OSNet, with the manually annotated M1 mouse near the tail base as a reference (left image).

(E) Confusion matrix illustrating classification outcomes for OSNet and the idTracker.ai network, showcasing the difference in model discernment capabilities on the test dataset.

(F) Tracking accuracy assessment for the 10flies and 14ants datasets compared to idTracker.ai, with representative snapshots of the datasets. The asterisk denotes accuracy as reported in the original publication.

of domains: four general simple online and real-time tracking (SORT) algorithms,[43–46] one tracker named DEVA with the capacity to "track anything,"[47] and four systems specifically designed for tracking experimental animals[11,12,17,29] (Tables 2, 3, and S6).

Due to the lack of essential trajectory correction mechanisms, the SORT algorithms may struggle to effectively perform stable tracking of lab animals in the long term. Even with the incorporation of appearance information, general

Table 1. Basic information of the annotated videos

Name	Size (MB)	Duration	Resolution	fps	Recording device	OC	OL	TBO	IBO	Ψ
4×WT[a]	208	01:00:00	648 × 724	30	Smartisan OD103	3,099	2.61	2.04	0.16	636
6×WT[b]	344	00:10:02	1,080 × 1,080	60	iPhone 13 Plus	1,440	1.35	1.16	0.16	266
2×WT[c]	157	00:30:10	608 × 864	30	iPhone 13	304	5.20	6.70	0.17	39
4×PD[d]	212	00:10:24	820 × 800	15	S-YUE Webcam	773	1.80	1.43	0.20	193
Noldus[e]	10.5	00:10:00	236 × 184	25	Noldus camera	392	3.72	2.39	0.15	93
14ants[f]	109	00:12:57	840 × 714	59.94	–	–	–	–	–	–
10flies[f]	32.5	00:10:12	948 × 920	60	–	–	–	–	–	–

OC, occluded counts; OL, occluded length; TBO, time between occlusions; IBO, intersection between occlusion; Ψ, a single video-complexity metric.
[a]With complex interactions, 1/4 is marked, cropped, and recoded to H.264.
[b]Cropped with ffmpeg.
[c]Complex environment, 1/2 is marked, animals move out of view from time to time, cropped and recoded to H.264.
[d]Low image quality and low FPS, cropped and recoded to H.264.
[e]Extremely low image quality, choppy video recording, cropped and recoded to H.264.
[f]Cropped and recoded to H.264, provided with idTracker.ai.

re-identification techniques fail to yield any benefits, as they lack fine-tuning on mouse images. DEVA, utilizing instance-segmentation information, faces challenges with mask deformation during tracking. Among the four tracking systems designed for experimental animals, only idTracker.ai can track multiple animals with relatively high precision, yet it struggles with complex experimental setups and videos containing more intricate interactions. Furthermore, idTracker.ai relies solely on the visual recognition of the mouse body, neglecting the visual features of the mouse's tail, potentially hindering tracking in some scenarios.

Considering the additional computation introduced by dense optical-flow estimation, we conducted ablation experiments by removing the flow-estimation process to investigate the role of the optical flow in segTracker.ai (Table S1). Results highlight its necessity in the low-frame-rate video (15 frames per second [fps]) and overall enhancement of tracking performance.

We also evaluated the performance of segTracker.ai in tracking mice during extended periods of inactivity. Prolonged inactivity can pose challenges for vision-based tracking algorithms,[17] yet segTracker.ai with Config. 2 achieved high tracking accuracy (92.30 IDF1 score). Its robust detector maintained clear differentiation among closely interacting mice, including during sleep, demonstrating reliable tracking capabilities in stationary scenarios.

In rigorous analysis across various mouse videos, segTracker.ai consistently outperformed the other methods, which not only underscores our method's superiority but also brings to light the inherent challenges in multi-animal tracking under complex social conditions.

Our evaluation also extended to smaller species, using the ant and fruit fly videos from idTracker.ai (Table 1). Here, segTracker.ai (Config. 2) achieved over 99.7% tracking accuracy with less time consumption compared to idTracker.ai, a testament to its computational efficiency and robustness (Figure 2F; Table S2). These results demonstrated segTracker.ai's advanced capabilities in handling a wide range of animal models and video qualities, contributing to ongoing efforts to enhance tracking accuracy and efficiency in the study of animal behavior.

Better-trained models yield better tracking results

The quality of instance masks can significantly impact segTracker.ai's tracking performance. Experimentation with YOLACT[49] as the instance-segmentation model (Figure 2A), known for its real-time efficiency, revealed issues such as mask leakage,[50] which notably impaired tracking results. Specifically, segTracker.ai based on YOLACT successfully tracked only one out of five videos, showing varied declines across metrics compared to other model configurations (Table 2).

To assess the impact of training samples on tracking performance, we trained instance-segmentation and pose-estimation models using 20%, 40%, 60%, and 80% of the available samples. Results indicate that insufficient training samples compromise instance mask quality and subsequent tracking performance. Notably, using as little as 40% of the training data (about 160 labeled frames) enabled successful tracking across all videos in our dataset. Performance generally improved with additional training samples, demonstrating a steady enhancement in tracking capabilities (Table 4).

SegTracker.ai demonstrates reliable tracking even with limited annotation data. However, adding more training data or switching to a model with better instance-segmentation performance proves to be effective in enhancing the tracking capabilities of segTracker.ai, even reversing the difficulty of fully tracking some complex videos. This flexibility is crucial for researchers seeking to optimize tracking outcomes without extensive computational expertise. By supporting integration with various instance-segmentation models, segTracker.ai leverages advancements in computer vision to continually enhance experimental animal tracking.

Decoding social-behavior variations in Shank3 knockout mice

We established a *Shank3* knockout mouse model associated with ASD (Figure S2A). Gene and protein deletions were confirmed through PCR and western blot analyses (Figures S2B and S2C). Classified into WT, heterozygous (HE), and homozygous (HO) categories, fine home-cage behavioral analyses and classic three-chamber social tests found that HO mice exhibited increased grooming and decreased movement,

Table 2. Benchmarking against SOTA tracking systems and algorithms

	4×PD			Noldus			6×WT			2×WT			4×WT		
Tracker	IDF	IDP	IDR	IDF	IDP	IDR	IDF	IDP	IDR	IDF	IDP	IDR	IDF	IDP	IDR
OCSORT[a,43]	18.8	18.5	19.2	37.2	37.9	36.5	58.6	58.8	58.3	11.9	12.3	11.5	15.8	16.0	15.5
ByteTrack[a,44]	4.7	4.7	4.6	22.0	22.3	21.7	27.6	27.7	27.5	7.8	8.0	7.6	7.7	7.8	7.6
DeepOCSORT[b,45]	25.7	26.1	25.2	27.2	26.7	27.7	48.9	49.1	48.7	16.4	17.0	15.9	17.5	17.7	17.2
BoTSORT[b,46]	43.8	44.1	43.5	34.2	34.6	33.7	37.8	37.9	37.7	19.7	20.2	19.1	17.4	17.5	17.2
DEVA[47]	40.3	40.3	40.3	23.1	25.2	21.3	45.8	46.1	45.6	11.8	8.0	22.2	22.8	21.1	24.8
AlphaTracker2[c,d,29,48]	28.5	28.7	28.3	40.4	40.8	40.0	27.1	27.2	26.9	49.5	51.6	47.6	26.7	26.8	26.6
DLC[d,11]	34.1	34.1	34.1	26.5	27.3	25.8	36.6	36.7	36.6	42.2	43.2	41.2	28.4	28.6	28.3
SLEAP[d,e,12]	29.0	29.5	28.4	35.4	35.8	35.1	42.9	43.2	42.6	42.8	44.5	41.2	27.7	28.3	27.1
idTracker.ai[17]	91.4	91.4	91.4	83.2	83.2	83.2	N/A			N/A			N/A		
Ours (Config. 1)[f]	95.1#	94.0#	96.3*	91.6#	89.9#	93.4*	96.6#	95.3#	97.9#	84.2#	82.7#	85.8#	95.1#	93.9#	96.3*
Ours (Config. 2)[g]	96.2*	96.8*	95.6#	92.1*	93.5*	90.7#	99.1*	99.4*	98.9*	92.3*	94.7*	90.0*	97.1*	98.2*	96.1#
Ours (YOLACT)[h]	N/A			81.0	87.1	75.6	N/A			51.3	56.5	46.9	44.8	51.0	39.9

The best-performing tracker is marked with an asterisk (*), while the second-best tracker is marked with a hash symbol (#). We also tried to evaluated TRex. However, due to the issue raised in https://github.com/mooch443/trex/issues/209, we were unable to evaluate its tracking performance on our datasets.

[a]SORT algorithms without utilizing appearance information.

[b]SORT algorithms utilizing appearance information. We used OSNet-AIN pretrained on ImageNet for the re-id network.

[c]We also attempted to evaluate the original AlphaTracker. However, we could not install the tracking system via Conda (UnavailableInvalidChannel error). We choose to evaluate AlphaTracker2 as an alternative as it is the successor of the original software and is developed by the same research team.

[d]AlphaTracker2 does not support datasets with different number of animals and DLC does not accommodate different input resolutions. The detection and pose-estimation models of SLEAP perform optimally when trained individually on each dataset rather than on a mixed dataset. We trained the pose estimation and detection models with a different dataset (see section "experimental procedures").

[e]Evaluation results were generated using SLEAP-simple tracker. We also attempted to assess the SLEAP-flow tracker. However, consistent errors occurred due to insufficient GPU RAM, despite using RTX 4090 (24G), which is the most advanced computing device available to us.

[f]Evaluation results were obtained using Mask Scoring RCNN[37] as the detector, YOLOv8-s Pose[38] for single-stage pose estimation, Farneback algorithm[39] for optical-flow estimation, and OSNet as the re-id model.

[g]Evaluation results were obtained using RTMDet-tiny-Ins[40] as the detector, MobileNetV2 for top-down pose estimation,[34,41] LiteFlowNetV2[42] for optical-flow estimation, and OSNet as the re-id model.

[h]Evaluation results were obtained using YOLOv8-s Seg[49] for instance segmentation, YOLOv8-s Pose for single-stage pose estimation, LiteFlowNetV2 for optical-flow estimation, and OSNet as the re-id model.

along with evident social deficits—specifically, reduced exploration time in the chamber containing a stranger mouse. In contrast, HE mice displayed no abnormalities in these experimental tests (Figures S2D and S2E). Traditional behavioral paradigms challenge our ability to fully and accurately understand social behavior, particularly in distinguishing subtle behavioral variations in HE individuals.

To overcome these limitations, we designed further experiments with two setups: the newcomer-introduction (NI) group and the homogeneous-genotype (HG) group. In the NI group, a new mouse of either WT, HE, or HO genotype was introduced to a group of WT mice. Conversely, the HG group involved both ASD-affected and control mice interacting in an open field, either individually or in genotype-matched groups. These experiments, supported by segCluster and kinematic analysis, aimed to elucidate the behavioral nuances across the three genotypes.

We next trained an autoencoder on identified pose sequences extracted from NI group videos, followed by uniform manifold approximation and projection (UMAP) dimension reduction and agglomerative clustering. The classification head of our model clustered behavior sequences by genotype, while the reconstruction head preserved behavior information within the embeddings. During training, the classification head ensures the behavior sequences of the same genotype were clustered closer together, and the reconstruction head ensures that the embeddings still contained behavior information. The classification accuracy and reconstruction quality are shown in Figure 3A, with key-point reconstruction errors reported in root-mean-square error (RMSE), measured in centimeters. We used the silhouette score (SS) to evaluate the goodness of clustering quality.[51] Typically, with SS over 0.50, it can be considered that a reasonable clustering structure has been detected. With the weakly supervised method, the model captures the structure of three genotypes (SS > 0.5), and the structure is more obvious in the remapped space than the original space (Figures 3B and 3C). Given the complexity and variability in social behaviors, we adjusted the SS threshold from the typical 0.5 to 0.4 to capture more nuanced patterns within each genotype's behavioral space. To be more concise, we picked the largest number of clusters above this threshold and took a closer look at these clustered phenotypes. As shown in Figures 3D and 3E, we were able to identify different phenotypes in each of the genotypes. The complete procedure yielded a comprehensive overview of behavioral patterns, as shown in Figure 3F.

Table 3. Evaluation result for tracking systems based on MOTA and MOTP

	4×PD		Noldus		6×WT		2×WT		4×WT	
Trackers	MOTA	MOTP	MOTA	MOTP	MOTA	MOTP	MOTA	MOTP	MOTA	MOTP
OCSORT	90.9	84.4	87.0	86.3*	98.1	89.4#	85.6	86.9#	94.3	90.1#
ByteTrack	62.5	73.9	84.0	82.1	94.6	82.0	83.7	83.7	91.5	84.6
DeepOCSORT	91.4	84.3	87.0	86.3*	98.1	89.4#	85.6	86.9#	94.3	90.1#
BoTSORT	94.2#	83.4	87.6*	86.2#	98.3#	89.3	85.9#	86.7	94.9#	89.8
DEVA	61.3	80.4	<0	73.4	65.9	86.8	<0	73.1	<0	81.5
AlphaTracker2	77.2	72.5	76.1	75.2	86.3	74.8	64.6	77.5	83.1	77.1
DLC	70.5	75.8	42.0	72.2	85.0	78.5	63.4	77.6	69.1	77.5
SLEAP	70.5	75.5	63.7	72.5	89.6	79.1	60.6	76.7	77.0	77.5
idTracker.ai	97.0*	90.1*	86.9#	73.5	N/A		N/A		N/A	
Config. 1	93.2	88.3#	81.6	70.1	98.7*	93.1*	90.2*	96.2*	97.1*	98.2*
Config. 2	90.9	83.6	84.9	85.5	95.8	88.8	84.6	86.3	92.4	89.2

The best-performing tracker is marked with an asterisk (*), while the second-best tracker is marked with a hash symbol (#). However, readers should be aware that MOTA/MOTP may unfairly favor methods using RTMDet (Config. 2) for result generation, such as idTracker.ai, segTracker.ai, and the SORT family, due to our ground truth derived from RTMDet and manual corrections. Methods such as DLC, SLEAP, and AlphaTracker2, which do not utilize RTMDet for detection boxes or instance masks, might not reflect their true performance on MOTA/MOTP. See also Note S1.

To evaluate the STGCN's capability in extracting meaningful features from skeletal sequences, we analyzed a training regimen on four NI videos, followed by a validation on two additional videos with entirely new WT, HE, and HO newcomers. This analysis achieved a validation classification accuracy of 0.47 (F1 score, 0.40; Figure 3A), which was markedly higher than the 0.33 expected by random chance. This elevated accuracy underscores the network's potential to predict an individual's genotype from fragments of skeleton sequences, thereby demonstrating its ability in identifying behavioral traits of the ASD mice and the effectiveness of using the genotype information as a supervision. Interestingly, the network tends to misclassify HO mice as WT, contrasting with traditional three-chamber tests, which struggle to differentiate WT from HE instead. This discrepancy highlights the nuanced behavioral differences observed between ASD mice under open-field and free-social conditions compared to the more controlled three-chamber test scenarios (Figure S2). The STGCN's effectiveness is further corroborated by an RMSE of 0.76 cm, indicative of its precise behavioral sequence reconstruction capabilities (Figure 3A).

Furthermore, we employed model interpretation techniques for deeper insights into the classifier and autoencoder, ensuring accurate phenotype categorization. LayerCAM, initially introduced in Figure 2D, was adapted for visualizing the spatiotemporal social-behavior CAMs of the clustered phenotypes. This method enabled the network to self-explain its processing, particularly highlighting its ability to identify the HO-genotype mice engaging closely with a WT mouse, attributing such interactions more significantly to the autistic behavior spectrum. We further explored the utility of spatiotemporal LayerCAM in a much more complex analysis protocol, such as a one vs. four interaction scenario, confirming its efficacy in distinguishing HO individuals based on their interaction patterns (Figure 3G).

Identification of social interaction patterns in ASD mice using spatiotemporal atlas

Inspired by previous research,[16] we constructed a one vs. one spatiotemporal social-behavior atlas, focusing on interactions between ASD and WT mice within a healthy WT community (Figure 3F). This atlas, rich in representative social-interaction patterns, allowed us to assign descriptive names to various behavioral clusters. While certain clusters demonstrated similar behaviors, such as either accepting or avoiding close interactions, one particular cluster encompassed a diverse range of interaction styles, challenging our ability to label it precisely (Figures S3 and S4).

The atlas is structured along two principal axes derived from behavior embeddings: UMAP1 and UMAP2. UMAP1 illustrated a gradient of interaction tendencies, ranging from welcoming, through avoidance, to outright escape or pre-emptive disengagement. UMAP2, on the other hand, delineated the intensity spectrum of close interactions, spanning from cautious, distant approaches to more engaging pursuits and intimate exchanges. Intriguingly, our analysis revealed distinct behavioral trends: HE mice exhibited a pronounced inclination to actively avoid or escape from close interactions with WT mice. Conversely, HO mice occasionally showed openness to engagement initiated by their WT counterparts. WT mice, in contrast, were more frequently engaged in active and reciprocal interactions. This atlas thus offers a nuanced view of social dynamics in ASD models, highlighting the variability and complexity inherent in these interactions.

Behavioral dynamics in *Shank3* knockout mice

To validate phenotypic distinctions identified by segCluster, we conducted a detailed kinematic analysis within the NI group of mice (Figure 4A). This was complemented by a carefully crafted set of behavioral discrimination criteria aimed at discerning various potential social behaviors (Figure 4B). In this schema, social behaviors were classified as active if initiated by the newly introduced mouse and passive if initiated by the existing healthy mice in the group. This methodological framework, while avoiding the need for elaborate behavior recognition neural networks, still demanded meticulous tracking of each mouse in the experiment, particularly those that were newly incorporated into the group. This approach allowed for a nuanced

Table 4. Impact of training set size on tracking results (IDF1)

Dataset	20%	40%	60%	80%	100%
4×PD	N/A	96.4#	95.6	96.8*	96.2
Noldus	33.6	71.8	90.0#	79.3	92.1*
6×WT	20.1	96.9	98.5#	93.7	99.1*
2×WT	81.3	86.3	91.1	91.8#	92.3*
4×WT	N/A	94.9	92.4	96.6#	97.1*

The experiment yielding the highest performance is marked with an asterisk (*), while the second-best tracker is marked with a hash symbol (#). The data indicate that training detectors and pose-estimation models with approximately 250 images can achieve high tracking accuracy without the necessity of retraining the models for different videos. Furthermore, continuously adding more annotated data also enhances tracking performance.

understanding of the social dynamics at play within different genotypic interactions.

Differential kinematic analysis in varied social settings

Our investigation commenced with the novel NI group setting. Notably, the HO mice displayed a marked decrease in movement speed and distance traveled, aligning with their known motor impairments (Figures 4C and 4D; Table S4). Despite their reduced mobility, these mice engaged more extensively in social interactions, as indicated by our tailored behavioral discrimination criteria. This unexpected propensity for social engagement was statistically significant, particularly in the passive tail and active head interactions, contrasting with conventional ASD models (Figure 4E).

Behavioral variations under social pressures

Expanding beyond the NI group, we examined mice within HE and HO groups and in isolation (Figure 4F). The data revealed significant behavioral alterations under different social conditions. In group settings, HE mice exhibited increased activity with longer net distance traveled, whereas HO mice became notably less mobile, traveling significantly less distance, suggesting an inverse relationship between their social environment and movement behaviors (Figures 4G–4J; Table S4 and S5). This observation was further substantiated through social-criterion analysis (Figure 4K), revealing that social contexts significantly affect the mobility and interaction patterns of these genetically modified mice.

Synthesis of social and motor impairment insights

Our findings depict a complex picture of social behavior in *Shank3* knockout mice. Contrary to traditional assumptions about ASD, HO mice, despite their motor limitations, engaged more actively in social settings. This suggests a nuanced interplay between social inclination and physical capability, challenging existing perceptions of social behavior in ASD models. Conversely, the reduced social interaction observed in HE mice, despite lacking motor deficits, hints at underlying behavioral or anxiety-related factors influencing their social engagement.

This comprehensive study, employing segTracker.ai, not only elucidates the nuanced social and motor phenotypes in *Shank3* knockout mice but also highlights the importance of considering both behavioral and physical aspects when studying neurodevelopmental disorders. Our approach offers a more holistic understanding of ASD, contributing to the development of more effective therapeutic strategies.

DISCUSSION

In this study, we have dedicated our efforts on two pivotal challenges in computational behavioral science: the upstream challenge of multi-animal tracking under complex interactive social conditions and the crucial downstream challenge of extracting social-behavior characteristics from skeletal sequences of animals with different genotypes. We have developed STCS, a system aimed at enhancing the study of social behavior in neuroscience.

STCS combines segTracker.ai, an advanced lab animal pose-tracking module inspired by idTracker.ai, and segCluster, a sophisticated tool for the analysis of social behavior. SegTracker.ai leverages state-of-the-art computer vision techniques for detection, instance segmentation, and pose estimation. This module is particularly adept at handling complex tracking with pose estimation across various species and video qualities with different combinations of deep-learning models, making it an invaluable asset for interdisciplinary research in computer science and behavioral neuroscience.

SegCluster, the second pillar of STCS, is an innovative module used to discern subtle differences in social behaviors of lab animals based on genotypes. This component employs a weakly supervised learning approach, coupled with dimensionality reduction and clustering analysis, to identify and categorize behavioral patterns in an unbiased, data-driven manner. The integration of these two modules of STCS enable us not only to efficiently gather data but also to comprehensively analyze complex social interactions.

Applying STCS to ASD research, particularly focusing on *Shank3* knockout models, has provided groundbreaking insights. Typically, HO *Shank3* knockouts are used as ASD models in research because HE knockouts often do not exhibit detectable anomalies using traditional behavioral methods.[8,24,25,52] This approach might overlook the complexities in the etiology of ASD, as HE models could offer vital clues to understanding the disorder's full spectrum. By conducting cage-free social experiments, STCS has adeptly identified subtle behavioral anomalies across a range of genotypes, capturing nuances in both HO and HE *Shank3* knockout mice, as well as in WT mice. This discovery is in line with recent research on *Shank3* knockout dogs, suggesting a broader applicability of our findings in understanding the complexity of ASD.[53,54] This finding underscores the capability of our system to capture nuanced differences across genotypes, suggesting a potential link between social disorders and locomotor impairments, potentially exacerbated by social pressures.

Remarkably, our study has shown that HE and HO mice with the same genetic defect exhibit contrasting behaviors, with HE mice being more active in social settings, while HO mice tend to be more stationary. This behavioral variance underscores the complexity of ASD, challenging traditional genotype-behavior correlations. Our integration of handcrafted rules for

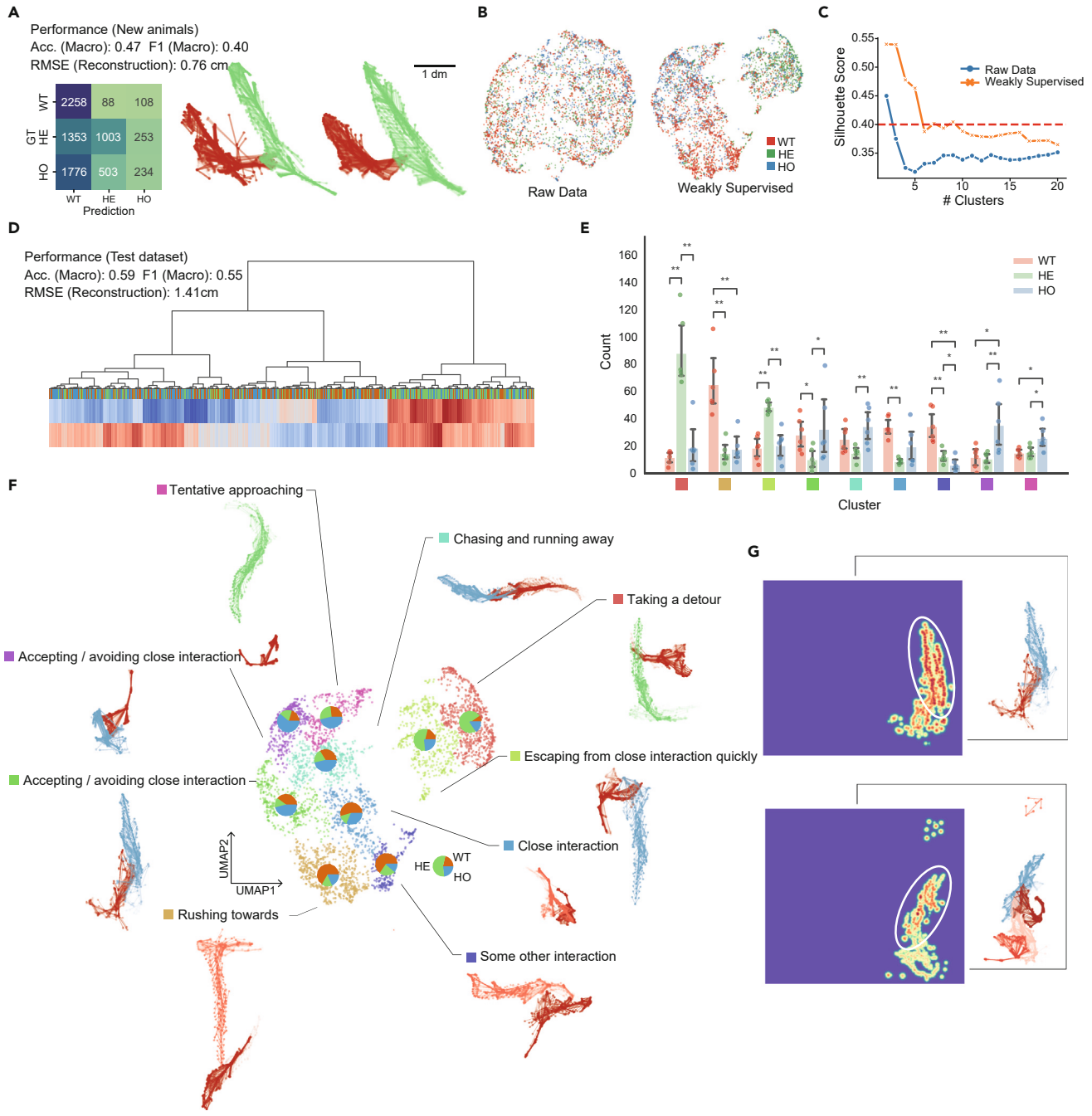

Figure 3. Deciphering social interaction patterns with segCluster

(A) Confusion matrix and autoencoder reconstruction outcomes for the test dataset comprising entirely new individuals, demonstrating predictive accuracy and reconstruction fidelity. Numbers in the confusion matrix represent social-behavioral chunks. Acc., macro average of the classification accuracy; F1, macro average of the classification F1 score.

(B) UMAP dimensionality reduction applied to raw and remapped data, highlighting complex clustering of social behaviors.

(C) Assessment of cluster integrity using SSs on raw and remapped data, with an SS threshold set at 0.40 to delineate satisfactory clustering outcomes.

(D) Agglomerative clustering dendrogram for remapped features, illustrating the hierarchical relationships and potential groupings within the data.

(E) Distribution of clustered social interaction behaviors among wild-type (WT), heterozygous (HE), and homozygous (HO) mice, with statistical significance determined by Dunn's test (*$p < 0.05$, **$p < 0.01$).

(F) Spatial-temporal social-behavior atlas illustrating the interactive dynamics of a new WT or ASD mouse integrating into a healthy community. Pie charts reflect composition of clustered phenotypes, with dark red skeletons representing existing WT community members and light red, green, and blue skeletons corresponding to the WT, HE, and HO newcomers, respectively (see also Figures S3 and S4).

(G) Spatial-temporal LayerCAM visualizations during one-on-one and one-versus-four social interactions highlighting skeletal movement trajectories. White ellipse and blue skeletons signify HO mouse, while red skeletons denote WT mouse.

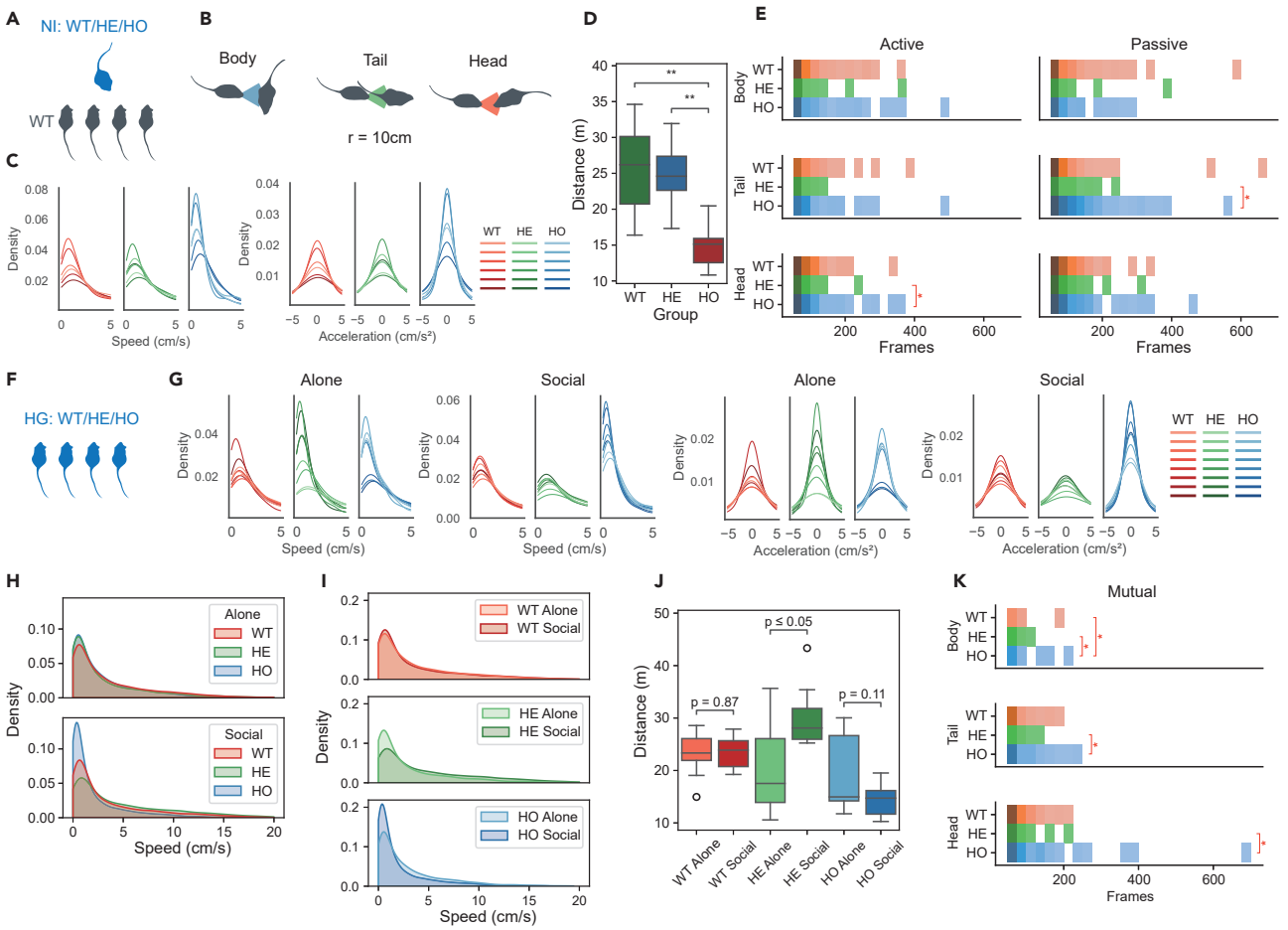

Figure 4. Kinetic metrics and rule-based social interaction distribution of ASD mice under social conditions

(A) Illustration of newcomer introduction (NI) group interaction focusing on social dynamics among WT, HE, and HO mice.

(B) Rule-based categorization of social interactions: body, tail, and head engagements, visualized with a 60° vision sector angle for interaction determination.

(C) Distribution plots showing variance in speed and acceleration among WT, HE, and HO mice within the NI group, highlighting kinetic behavior differences.

(D) Boxplots representing the total distance traversed over 10 min by WT, HE, and HO mice, emphasizing movement disparities in the NI group (Kruskal-Wallis test followed by Dunn's test, $^{**}p < 0.01$).

(E) Duration distribution of interaction types across ASD mouse genotypes, distinguishing active initiations by newly introduced mice versus passive responses from existing group members (Kruskal-Wallis test followed by Dunn's test, $^{*}p < 0.05$).

(F) Representation of the homogeneous genotype (HG) group setup, featuring consistent genotype cohorts.

(G) Overlaid density curves reflecting the speed and acceleration profiles for WT, HE, and HO mice in isolation versus within a group of four (HG group), illustrating the impact of social context on movement dynamics. A distinct stratification of kinematic parameters among WT, HE, and HO mice could be observed in social settings.

(H) Comparative speed distribution between solitary and grouped WT, HE, and HO mice, revealing differences in activity patterns influenced by social surroundings.

(I) Comparative speed distribution between WT, HE, and HO mice in solitary and grouped, revealing shifts in activity patterns influenced by social surroundings.

(J) Box-and-whisker plots detailing the total travel distance over a span of 10 min for different genotypes under varying experimental conditions, indicating genotype-specific mobility and social interaction trends (one-way ANOVA followed by Tukey's HSD test).

(K) Graphical representation of interaction duration across various social behaviors, with significant differences marked by genotype-specific responses (Kruskal-Wallis test followed by Dunn's test, $^{*}p < 0.05$), illustrating the intricacies of social engagement among ASD models. See also Tables S4 and S5.

social-behavior analysis has validated the phenotypes identified by segCluster, also highlighting the limitations of such rule-based methods in capturing the full spectrum of behavioral patterns.

In conclusion, STCS has not only contributed to our understanding of ASD but also represented a significant advancement in the study of social behavior. By providing a more comprehensive analysis of social interactions, STCS holds the promise of enhancing our understanding of a wide range of social behaviors and neurodevelopmental disorders. Its potential applications extend beyond ASD to other areas of behavioral neuroscience, such as aggression,[55,56] mating behaviors,[57,58] social hierarchy,[59–61] and social avoidance in depression,[62,63] making it a versatile tool for future research. STCS, therefore, stands as a testament to our

commitment to removing the limitations in sociability testing for rodents and, more broadly, in the field of behavior science.

EXPERIMENTAL PROCEDURES

Hardware and software configurations

Computational experiments were conducted on a Lenovo Thinkbook16p 2021 laptop equipped with an RTX3060 Max-Q (6GB) GPU, 16 GB RAM, and an AMD Ryzen 5800H CPU unless otherwise noted. These hardware configurations are readily available in the commercial market. Experiments employing our methods were executed on Ubuntu 20.04.6 LTS within the Windows Subsystem for Linux 2 (WSL2). In contrast, experiments involving DLC and idTracker.ai were performed on Windows 11, using Python environments provided by the respective packages. Notably, although the Thinkbook16p boasts 16 GB of RAM, a limit of 12 GB was allocated to WSL2 for computations using STCS.

The recording conditions of the tracking video dataset are detailed in Table 1, while the recording conditions for social-behavior videos of ASD mice are illustrated in Figure S5.

Dataset construction and annotation

We developed datasets for three animal species—mice, ants, and fruit flies—targeting detection, pose-estimation, and instance-segmentation tasks. These datasets were divided into training (70%), validation (20%), and testing (10%) sets. For ants and fruit flies, image data were sourced from videos included in the supplementary materials of the idTracker.ai paper. In contrast, mouse image data were sourced from a diverse array of videos captured using various devices and experimental settings, as detailed in Table 1, along with additional recordings conducted for ASD behavior experiments. We defined specific key points for each species: eight for mice (nose, left ear, right ear, left hind leg, right hind leg, tail base, mid-tail, tail tip), seven for ants (head, thorax, abdomen, 2 keypoints on each antenna), and six for fruit flies (left eye, right eye, thorax, abdomen, left wing, right wing).

Our approach for extracting images from video frames was multifaceted, combining equidistant interval, random sampling, and K-means clustering (as reported by DLC). Duplicate images were removed by annotators, who then conducted selective annotations. Further, inspired by SLEAP's human-in-the-loop annotation mode, we introduced a method for exporting low-confidence images for manual review, integrating the revised data back into the dataset. The final mouse dataset comprises 574 images (split as 401:115:58), the ant dataset 36 images (25:7:4), and the fruit fly dataset 33 images (23:6:4). All images were annotated using coco-annotator and exported in MSCOCO format.

For ground-truth generation in tracking tasks, we initially ran our algorithm on the videos listed in Table 1. The outputs were then transferred to label-studio for manual corrections. In the 1-h-long 4×WT dataset, which involved frequent intense interactions, three human annotators were employed. Annotator A was tasked with checking for ID switches frame-by-frame, annotator B focused on discerning minor visual differences between lab animals, and annotator C intervened for final decisions in cases of inconsistency between A and B. All tracking datasets, except for the 14ants and 10flies datasets, presented in this article comply with the MOT challenge format.[32]

We utilized the occlusion metrics proposed by Pedersen et al.[31] to measure the complexity of animal interactions within each video defined as follows:

(1) Occlusion count (OC): the average number of occlusion events per second.
(2) Occlusion length (OL): the average time in seconds of all occlusion events.
(3) Time between occlusions (TBO): the average time in seconds between occlusion events.
(4) Intersection between occlusions (IBO): a measure of how large a part of the animal is part of an occlusion event. The intersection in frame f for animal i is given by

$$IBO_{i,f} = \frac{1}{|bb_i|}\sum_{j=1}^{n} bb_i \cap bb_j, j \neq i$$

where n us the number of animals in an occlusion event, and bb_j is the set of pixel coordinates in the bounding box of animal j.

(5) Ψ: a single complexity measure combining all the four metrics given by

$$\Psi = \frac{OC \times OL \times IBO}{TBO},$$

and the measure falls in the interval $(0, +\infty)$ where a larger value indicates a higher video complexity.

Detection, instance segmentation, and pose estimation

Our system's pipeline is designed to be model agnostic, harnessing the versatility of the OpenMMLab platform (https://github.com/open-mmlab/mmcv). This flexibility allows users to experiment with various algorithm combinations. For our specific implementation, considering hardware constraints, we opted for Mask Scoring RCNN,[37] RTMDet-tiny-Ins,[40] and YOLOv8-s Segm (YOLACT[49]) for instance-segmentation, YOLOv8-s Pose,[38] and MobileNetV2-based pose-estimation[34,41] approach.

For high-quality instance segmentation, we adhered to most default data augmentation protocols for training RTMDet models, including RandomResize, RandomCrop, YOLOXHSVRandomAug, RandomFlip, and MixUp. Han et al.[16] suggest the effectiveness of CopyPaste augmentation in multi-animal tracking; therefore, we incorporated this technique as well. However, mosaic augmentation was excluded due to compatibility issues.

We trained the Mask Scoring RCNN network and RTMDet-tiny-Ins model with MMDetection toolkit (https://github.com/open-mmlab/mmdetection) and YOLOv8-s Segm model with the ultralytics toolkit (https://github.com/ultralytics/ultralytics).

YOLO Pose is a single-stage pose-estimation method that employs deep-learning models to directly regress the coordinates of key points. The other pose-estimation module employs a top-down methodology, utilizing MobileNetV2 as the backbone—a popular lightweight model used in various applications, including animal pose estimation in DLC and SLEAP. This module predicts heatmaps representing the probability distributions of key points for each individual. In post-processing, peaks on these heatmaps are connected to form skeletons. We implemented data-augmentation strategies such as RandomFlip and RandomBBoxTransform to enhance the performance of the pose-estimation model. YOLO Pose was trained with the ultralytics toolkit and MobileNetV2 top-down pose-estimation model was trained with MMPose toolkit (https://github.com/open-mmlab/mmpose).

We observed discrepancies between the evaluation tools provided by the YOLOv8 system and the results from cocoapi. Therefore, we uniformly used cocoapi for evaluating our instance-segmentation and pose-estimation results. The sigma value for animal key points was consistently set to 0.1.

Subsequent to instance segmentation and pose estimation, we isolated instances by masking out the background. Frames were then cropped around the minimal enclosing rectangle for each instance and aligned using pose data, resulting in what we term "instance frames." These frames are preserved for further classification.

Optimizing detection, instance segmentation, and pose estimation specifically for laboratory animals was not within the scope of this study. The models chosen for our experiments yielded satisfactory results. We encourage readers seeking more technical details or alternative methods with potentially superior performance on laboratory animals to consult the original papers on SIPEC, SLEAP, DLC, etc.

Generating tracklets

Local tracklet generation primarily hinges on the instance-segmentation masks and optical-flow estimates, which predict subsequent frame masks. Initially, a soft non-maximum suppression (NMS)[64] is applied to reduce redundant detections, followed by retaining the top k detections based on confidence scores, where k represents the maximum number of animals in the video. We then calculate the distances between masks in adjacent frames for data association.

The distance metric used is intersection over maximum (IoM), akin to intersection over union (IoU). For Boolean matrices M_1 and M_2, the IoM is defined as

$$\mathrm{IoM}(M_1, M_2) = \frac{\sum(M_1 \wedge M_2)}{\max(\sum M_1, \sum M_2)}.$$

For an instance mask M_t at time t, we predict its next frame location ($\tau + 1$) using optical flow between the current frame and the subsequent frame $I \in R^{W \times H \times 2}$ (forward warping):

$$\widehat{M_{\tau+1}}\left(\left[x + I_{xy0}\right], \left[y + I_{xy1}\right]\right) = M_\tau(x, y),$$

where $[\cdot]$ denotes rounding to the nearest integer function.

Given the potential for warping holes and misalignment in forward warping, we initially compute the mean masking intensities, apply thresholding, and then perform a closing operation to fill predicted mask holes.

A cost matrix based on IoM distances is constructed for associating forwarded and detected masks:

$$\mathrm{Cost}[i, j] = -\mathrm{IoM}(\widehat{M_{\tau,i}}, M_{\tau,j}).$$

Associations are established using the Hungarian algorithm, accepted only if the cost matrix entry exceeds a threshold $\mathrm{Cost}[i, j] \leq \mathrm{Thr}_{\mathrm{mask}}$. In our experiment, we set $\mathrm{Thr}_{\mathrm{mask}} = -0.7$.

Unmatched forwarded masks are kept alive for a predetermined period, with the tracklet marked as stale if no new matches occur. Conversely, detection of an unmatched new mask results in a new tracklet added to the active pool.

After generating the tracklets, we applied the three-sigma rule to exclude those with unusually fast or slow-motion speeds, as well as those with average areas significantly larger or smaller than typical. This approach effectively filters out anomalies in movement speed and size. Tracklets filtered out by this criterion will undergo classification at the final stage of the association procedure.

Merging tracklets with soft border

In scenarios where local tracklets present temporary contradictions—preventing their amalgamation into a singular track—we leverage prior knowledge to address this challenge. To this end, we construct an undirected graph $G(V, E)$ representing the relationships among local tracklets, where V is the set of tracklets and E is the set of pairs with contradictory attributes.

The criteria for determining contradictions between tracklets in our system, segTracker.ai, diverge from those used in idTracker.ai. The latter marks tracklets as contradictory if they coexist in the same frame upon iteration through all frames. In contrast, segTracker.ai, grappling with imperfect segmentation masks, implements soft borders for tracklets. We define the edge set E as

$$E = \{\{t_1, t_2\} \mid \mathrm{Overlap}(t_1, t_2) \geq \mathrm{Thr}_{\mathrm{border}} \wedge t_1, t_2 \in V\}$$

where $\mathrm{Overlap}(\cdot, \cdot)$ quantifies the number of frames in which two tracklets coexist. We set $\mathrm{Thr}_{\mathrm{border}} = 30$ frames in our experiments.

Utilizing the contradictory graph G, mutually contradictory tracklet groups are discerned by identifying maximal cliques.[65] For a clique C, it holds that $|C| \leq k$, where $|\cdot|$ denotes the cardinality of a clique. Considering the largest cliques possible, where $|C^*| = k$, we define a mergeable tracklet set S_i as

$$S_i = N(c_1) \cap \cdots \cap N(c_{i-1}) \cap N(c_{i+1}) \cap \cdots \cap N(c_k),$$

with c_j being the vertex in C^* and $N(\cdot)$ denoting the neighbors of a given vertex.

Visual identification and distance-based association

In segTracker.ai, tracklets are merged to form tracks using two primary methods. The first method merges tracklets based on the confidence scores provided by the classification model. We set a confidence threshold of 0.95 and establish a minimum number of instance frames required for merging. The system first identifies all unclassified tracklets with a confidence above 0.95 and merges them with the corresponding individual track. If contradictory tracklets are classified as the same individual during this process, the merging is halted, and a new model training round is initiated. However, relying solely on high-confidence tracklets may lead to identification failures; hence, we introduce a secondary strategy that considers the confidence distance between unclassified and previously classified tracklets. We treat the association as a progressive clustering process, analogous to the concept of triplet loss. This loss guides the classification model to minimize distances within the same cluster and maximize distances between different clusters. Meanwhile, the fully connected (FC) layers act as classifiers, taking feature maps as inputs and yielding classified individuals as outputs. In our design, the FC layers are retained, with the clustering process being externalized from the loss function. This allows the model to leverage the FC layers' strong fitting capabilities for more accurate predictions.

Given the classification model f and all the images in the classified track T_i for individual i, we take the average of neural network output $P_{T_i} = \overline{f(T_i)}$ as the clustering center. We propose two methods to measure the distance between unclassified tracklets and classified tracks. First, using the Jensen-Shannon divergence (JSD), we compare the average output distribution $P_t = \overline{f(t)}$ for an unclassified tracklet t against P_T:

$$\mathrm{JS}(\mathrm{P_t} \| \mathrm{P_T}) = \frac{1}{2}\mathrm{KL}\left(\mathrm{P_t} \| \frac{\mathrm{P_t}+\mathrm{P_T}}{2}\right) + \frac{1}{2}\mathrm{KL}\left(\mathrm{P_T} \| \frac{\mathrm{P_t}+\mathrm{P_T}}{2}\right),$$

where $KL(\cdot \| \cdot)$ denotes the Kullback-Leible divergence, which is defined by:

$$KL(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}, i = 1, 2, \cdots, k.$$

Alternatively, we consider the output probability distributions as appearance features, calculating the cosine distance:

$$\cos(P_t, P_T) = \frac{P_t \cdot P_T}{|P_t| \cdot |P_T|}.$$

The distance matrix between classified tracks and unclassified tracklet groups (as formed in the "merging tracklets with soft border" section) is then computed. The Hungarian algorithm is employed to determine optimal matches, accepted if the average matching distance falls below a set threshold, $\mathrm{Thr}_{\mathrm{dist}}(JS) = 0.1$ and $\mathrm{Thr}_{\mathrm{dist}}(cos) = 0.05$ in our experiments.

Tracking performance evaluation

We evaluated our tracking results on the 10flies and 14ants datasets against the original outputs from idTracker.ai, as provided in their supplementary data. The idTracker.ai authors reported near-perfect tracking accuracies (100% and 99.943%, respectively). Our analysis, based on the trajectories_wo_gaps.npy data, revealed minor discrepancies in accuracy, potentially stemming from the trajectories_interpolated.npy file. We converted the centroids of individual masks into the idTracker.ai NumPy array format, treating each entry with a not-a-number (NaN) value as non-identified.

For each individual i at time τ, centroid coordinates from our method and idTracker.ai are denoted as $p(i, \tau)$ and $p^*(i, \tau)$, respectively. We marked pairs as inconsistent when the Euclidean distance $\|p^*(i, \tau) - p(i, \tau)\|$ exceeded the animal's length. Instances with inconsistencies lasting less than 0.5 s were compared against idTracker.ai outputs as ground truth. Inconsistencies over 0.5 s prompted manual video clip checks for ground-truth determination.

Additionally, we tested idTracker.ai (version 4.0.12) on our own video datasets. In instances where idTracker.ai encountered an out-of-memory (OOM) error, we attempted the process up to three times; persistent OOM errors led to marking the results as not applicable (N/A). Given that idTracker.ai does not provide detection boxes or pose-estimation results, our evaluation using TrackEval (https://github.com/JonathonLuiten/TrackEval) involved matching our detection frames generated by RTMDet with the tracking outputs of idTracker.ai. This approach allowed for a comparative assessment of tracking accuracy under our detection framework.

We compared our system with the unsupervised tracking component of DLC on our own video datasets. Due to DLC and AlphaTracker2's (AT2) requirement for consistent image resolution, we annotated 120–150 images per video for each dataset, undergoing 30,000 training iterations and 400–600 epochs (400 for pose estimation and 600 for detection), respectively. Although SLEAP supports multiple input resolutions, we have found that

training a model individually for each video yields better pose-estimation quality, and thus SLEAP has also adopted this dataset-specific approach. The annotation process was dual pronged: equidistantly sampled images annotated by our model and manually corrected, and images obtained via DLC's internal K-means-based sampling algorithm, which were exclusively manually annotated. We trained DLC's re-ID transformer model using default parameters. To evaluate tracking performance, we converted DLC, SLEAP, and AT2's pose-estimation results into detection boxes, allowing for a level comparison of tracking capabilities between DLC, AT2, and our system.

Due to the plethora of configurable parameters offered by software like SLEAP and DLC, it is impractical to explore the optimal parameter combination using methods such as grid search. Therefore, we largely maintained the default parameter settings of the software. A list containing the configurable parameters used during our experiments is included in Note S2.

DEVA offers a video instance-tracking solution but requires the provision of instance masks for each frame. We input the instance-segmentation results from RTMDet-tiny-Ins into DEVA, as it is the best instance-segmentation model we have validated. We set DEVA's tracking mode to semionline and max_num_objects to 6 for tracking mice.

For OCSORT, ByteTrack, DeepOCSORT, and BoTSORT, we used the default parameters provided by yolo_tracking (https://github.com/mikel-brostrom/yolo_tracking) and input the detection boxes obtained from RTMDet-tiny-Ins into these algorithms to acquire tracking results. Based on our visual observations, we consider OCSORT to have stronger capabilities in motion modeling. However, due to the lack of an appropriate re-identification mechanism, all SORT algorithms struggle to achieve long-term stable tracking results.

We focus mainly on identity metrics including identification F1 score (IDF1), identification recall (IDR), and identification precision (IDP),[66] which primarily measures the data-association accuracy for MOT trackers. This emphasis is due to the majority of bounding boxes being automatically generated by RTMDet and subsequently manually corrected, rendering direct comparisons based on detector performance with other methods as potentially unfair (Note S3). However, we also provided comprehensive evaluation results based on multiple object tracking accuracy (MOTA)[67] metrics (Table 3) and higher-order tracking accuracy (HOTA; Table S6),[68] which consider both the stability of long-term tracking and the performance of the detector. For IDF1, IDR, IDP, MOTA, and HOTA, higher values indicate superior tracking performance. We noted that some animal-tracking tools performed poorly on our dataset, not only due to instability in long-term tracking but also because of a higher rate of missed detections. This could be attributed to our dataset being composed of various types of images, the need for more annotated data, or the requirement for more meticulous tuning by domain experts in these methods.

Weakly supervised behavior remapping

We utilized the spatiotemporal GCN (STGCN) implemented in MMAction2 as the backbone for our autoencoder, chiefly due to its excellent performance in spatiotemporal modeling. We represented the mouse skeleton as a directed graph with specific key points connections such as {(left ear, nose), (right ear, nose), (tail base, nose), (left hind leg, tail base), (right hind leg, tail base), (mid tail, tail base), (tail tip, mid tail)}.

Traditionally, STGCNs analyze individual behaviors and integrate this information in the final layer via an FC network for social action recognition. To augment its capacity for modeling social interactions without altering the STGCN's structure or input-output format, we introduced a novel, lightweight SC module. This module is essentially a linear layer that amalgamates features from different individuals at a specific key point and time. We termed it Conv due to its resemblance to group convolution operations in aggregating behavioral information across various individuals and for its consistency with the existing GCN Block structure. To ensure that this module does not compromise the extraction of individual behavioral features, we incorporated residual connections into the SC module. The output for a given input $x_{in} \in \mathbb{R}^{N\times T\times K\times C}$ is given by

$$\boldsymbol{x}_{out}[:,\tau,k,:] = x_{in}[:,\tau,k,:] + Act(W_k \boldsymbol{x}_{in}[:,\tau,k,:] + b),$$

where N is the number of individuals, τ the timestamp, k a keypoint, C the feature channel number, $Act(\cdot)$ the activation function, $W_k \in \mathbb{R}^{NC\times NC}$ the learnable weight matrix for keypoint k, and b is the bias term.

The STGCN autoencoder is structured with symmetrically arranged encoder and decoder sets (Figure S6). The encoder comprises seven GCN blocks, while the temporal modeling module utilizes a multi-scale temporal convolutional network. During encoding, feature channels are expanded to 32 and subsequently downsampled to four channels at a 1/2 sampling rate in specific layers, with temporal downsampling also implemented. The decoder, with six GCN blocks and a simpler temporal convolutional network, follows a similar expansion and downsampling pattern, with an additional FC layer mapping the features to Cartesian coordinates for action sequences. The autoencoder is trained using the mean squared error (MSE) loss function. The classification head of the autoencoder, tasked with predicting the genotype from extracted features, takes the encoder's output as input. Given the indirect correlation between social behaviors and genotype information, we treated genotype prediction as a weakly supervised task. To accommodate potential inaccuracies in genotype labels, we implemented label smoothing with a larger epsilon value (0.6) for the three genotypes (WT, HE, HO) and trained the model using cross-entropy loss.

For neural network input, we segmented action sequences from videos into 2.5-s intervals, retaining different animals as necessary. For instance, in the NI-group experiment, we included a newcomer and one WT mouse to form a simpler 1vs1 sequence. We adopted two strategies for generating training and test sets. One strategy assessed whether the neural network's extracted features could differentiate behavioral variations among different genotypes: we trained on selected sequences from four videos per genotype, chose the model with the highest reconstruction accuracy, and tested on the remaining videos to predict genotypes, comparing the accuracy against a random-guess baseline of 0.33. The other strategy, aimed at feature extraction and clustering, involved dividing each video into two equal segments, using the first half for training and the second for testing, selecting the model with the highest classification accuracy on the test set for feature extraction.

Clustering of ASD mice behavior

In the process of clustering ASD mice behavior, we refined the action sequences designated for feature extraction, as detailed in the section "weakly supervised behavior remapping." This refinement involved filtering based on the minimum interspecies distance, focusing exclusively on sequences where the distance between two mice remained below 10 cm. Utilizing the bottleneck layer of the STGCN autoencoder, we extracted embeddings from these sequences.

For dimensional reduction, we applied UMAP[69] to transform these high-dimensional feature sets into a two-dimensional space. This step facilitated a more tractable analysis of the behavioral data. Subsequent hierarchical clustering was performed, with the optimal number of clusters determined by exceeding a predefined threshold of 0.4, as gauged by the SS.

To understand the genotype distribution within each identified social-behavior category, we conducted statistical analyses using the Kruskal-Wallis test, followed by Dunn's test for *post hoc* analysis. This approach allowed for a comprehensive assessment of behavioral variations across different genotypes.

LayerCAM for interpreting spatiotemporal social behavior in STGCN autoencoders

We employed LayerCAM to perform an interpretable analysis of spatiotemporal social-behavior sequences. For a more detailed understanding through CAMs, we focused on the final GCN layer preceding temporal downsampling in our visualization analysis. This selection was critical to achieve finer granularity in the resulting CAMs.

Given that the length of behavior sequences might require padding to match the standard input length of the neural network, we resampled the generated CAMs to ensure congruence with the original sequence lengths. The resampling process involved summing the CAMs across feature channels and mapping these aggregated values to their respective time points and skeletal key points.

Furthermore, to mitigate the issue of false-positive activations commonly observed at relatively stationary key points, a consequence of the continuous accumulation in spatiotemporal CAMs, we adopted a specialized visualization technique. This technique prioritizes the maximum activation for each key point over the standard practice of accumulating activation values. Such an

approach ensures that the visualizations accurately reflect the most significant activations, thereby providing a more precise interpretation of the network's focus and decision-making process in relation to specific behavioral sequences.

Mouse

Shank3 KO mice were bred and raised under identical conditions in compliance with the guidelines set by the Institutional Animal Care and Use Committee at the animal core facility of the Huazhong University of Science and Technology in Wuhan, China. Mice were housed in groups of three to five per cage and maintained under a 12-h light-dark cycle, with lights switched on at 8 a.m. The environment was kept at a consistent ambient temperature (21°C ± 1°C) and humidity (50% ± 5%). Behavioral tests were conducted during the light phase of the cycle.

Shank3−/− mice were generated by Gem-Pharma-Tech, Nanjing, China (Primers: KO-F, 5′-AGGGCAGGGAAGCCAATAAGCATCCAAT-3′; KO-R, 5′-ACTCACCCACTGTCCACCCACCCGAAAT-3′; WT-F, 5′-TGGCCATGGC TCTATGCTGG-3′; WT-R, 5′-GGGCCACCTTATCTGTGCTGT-3′).

Mice genotyping

DNA was extracted from tail snips for pup identification using a PCR-based sequence analysis. Tail snips (4 mm) from mice were collected in sterile 1.5-mL microcentrifuge tubes containing 100 μL of tissue digestion solution buffer (composed of 0.5844 g of NaCl, 0.1211 g of Tris, 0.744 g of EDTA-2Na, and 5 mL of 10% SDS, pH 8.0) along with 100 μL of proteinase K. The samples were then incubated on a shaking table at 55°C for 6 h. Subsequently, the digested samples were heated at 98°C for 18 min, followed by centrifugation at 14,000 rpm for 18 min. Post centrifugation, 10 μL of supernatant was transferred to a new sterile 1.5-mL microcentrifuge tube and mixed with 40 μL of double-distilled water. PCR reactions were carried out in a 20-μL volume, comprising 1.5 μL of isolated tail DNA sample, 6.5 μL of H_2O, 10 μL of 2× GoTaq master mix, and 1 μL each of forward and reverse primers, as previously described. The PCR cycling conditions were as follows: 98°C for 3 min (98°C for 10 s, 66°C for 20 s) for 35 cycles, 68°C for 10 min, and 4°C for 5 min. Subsequently, the PCR products were resolved on 2% agarose gels in Tris-borate-EDTA (TBE) buffer.

Western blot

Brain tissues were isolated from $Shank3^{+/+}$, $Shank3^{+/-}$, and $Shank3^{-/-}$ mice, homogenized, and diluted in a buffer containing 200 mM Tris-Cl (pH 7.6), 8% SDS, and 40% glycerol. The protein concentration was determined using a BCA kit (Pierce). Final concentrations of 10% β-mercaptoethanol and 0.05% bromophenol blue were added, and the samples were boiled for 10 min in a water bath. Subsequently, the proteins in the extracts were separated by 10% SDS-PAGE, transferred to nitrocellulose membranes, and blocked with 2% BSA in TBST for 1 h. The membranes were then incubated overnight at 4°C with specific primary antibodies, including rabbit monoclonal anti-Shank3 (1:1,000, sc377088, Santa Cruz) and mouse monoclonal anti-beta-actin (1:5,000, MA5-15452, Invitrogen). Following primary antibody incubation, the membrane was washed with TBST and incubated with the appropriate secondary antibodies (1:1,000, Odyssey) for 1 h at room temperature (22°C ± 1°C). After further washing, the blots were scanned using an infrared imaging system (Odyssey, LI-COR). Band densities were quantitatively analyzed using Kodak Digital Science 1-D software (Eastman Kodak).

Home cage behaviors

Individual mice of each genotype were video-recorded alone in the PhenoTyper home cages (Noldus, Holland), which provides a home cage environment for a mouse, with the ability of bedding, shelter, and food and water supply. Mice were placed into the PhenoTyper home cages (40 × 40 × 40 cm, Noldus, Holland) and monitored between 1 p.m. and 4 p.m. Locomotion and spontaneous behavior were detected by the interruption of infrared beams by the body of the mouse over a consecutive 30 min. Automated video analysis was conducted by EthoVisionXT (Noldus, Holland) to index time spent performing individual behaviors.

Three-chamber sociability test

The three-chamber apparatus, constructed from durable white plastic, measured 102 cm in length, 47 cm in width, and 45 cm in height. The two transparent side-crossing walls were designed with a 10-cm width to allow mice to move freely between chambers. The cup-like containers, each with a diameter of 10 cm and a height of 12 cm, were constructed using metal wires with 1-cm gaps, enabling air exchange while preventing direct physical interactions. Atop each cup, a cone-shaped object, also made from the same material, was positioned to deter climbing by the test mouse. During the habituation session, mice were gently placed in the center of the apparatus, flanked by two empty containers on each side, and given a 10-min period for exploration. Following a 1-h delay, mice progressed to a 10-min test session. Here, an unfamiliar WT mouse, matched in age, sex, and strain, was placed in one container to serve as the social stimulus, while a novel object occupied the other. The positioning of containers was systematically alternated across all experiments to prevent potential biases. Exploratory time on each side during the habituation session was meticulously recorded. In the subsequent test session, interaction behavior was defined as a mouse approaching its nose within 1 cm of a container. The social preference discrimination index, a metric previously described, was computed as $(M - O)/(M + O)$, with M representing the duration spent interacting with the unfamiliar mouse, and O denoting the time devoted to exploring the novel object.

Statistics

Statistical analyses were conducted using Scipy version 1.10.1 and the statannotations package version 0.5.0 for statistical functions, with *post hoc* tests performed using scikit_posthocs version 0.7.0. The significance threshold was set at $p < 0.05$. Kinematic analyses of experimental animals were initially subjected to a Shapiro-Wilk test for normality and Levene's test for homogeneity of variances. Kinematic data and duration of rule-based behavioral tasks that met the assumptions of normality and homogeneity of variance were analyzed using ANOVA, with Tukey's honestly significant difference (HSD) test for *post hoc* comparisons. For data that did not meet the requirements for ANOVA, Kruskal-Wallis non-parametric tests were used, followed by Dunn's *post hoc* test for pairwise comparisons. The effectiveness of the SC module was assessed using the Mann-Whitney U test. Differences in the distribution of clustered behavioral categories across different genotypes of experimental animals were evaluated using the Kruskal-Wallis test.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to the lead contact, Hao Li (lihaochn@hust.edu.cn).

Materials availability

No new biological materials were generated by this study.

Data and code availability

The source code, benchmark datasets, video clips along with their tracking and pose-estimation results, and all trained models including their configuration files from our research are publicly accessible on figshare.[70–74] The STCS code is also publicly available at our GitHub repository: https://github.com/tctco/STCS.

ACKNOWLEDGMENTS

This work was supported by grants from the National Natural Science Foundation of China (grants: 31721002, 81920208014, and 31930051 to Y.L.; 32200795 to H.L.) and Natural Science Foundation of Hubei Province (2022CFB608) to H.L. We thank the idTracker.ai team for developing and updating the excellent software. We also thank Dr. Chuan Lai for kindly providing the 2×WT video.

AUTHOR CONTRIBUTIONS

Conceptualization, C.T., Y.Z., and H.L. Software engineering, GUI design, algorithm design and analysis, benchmarking test, and the new network module

design, C.T. The 4×WT video was recorded by Y.Z. The 6×WT, 4×PD, Noldus, NI, and HG videos were recorded by S.Z. and H.L. The 4×WT tracking dataset was annotated by C.T., Y.Z., and M.X. The 6×WT, 2×WT, 4×PD, and Noldus tracking datasets were annotated by C.T. The pose and detection datasets were annotated by C.T., Y.Z., X.L., and R.Z. Behavior experiments were conducted by S.Z., M.X., and H.L. Biochemistry experiments were conducted by M.X. The article was mainly written by C.T., Y.Z., H.L., and Y.L. with input from all authors. H.L., G.M., Y.L., and L.Z. co-supervised the project.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work, the authors used ChatGPT for language polishing and translation purposes. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2024.101057.

Received: February 27, 2024
Revised: April 18, 2024
Accepted: August 13, 2024
Published: September 10, 2024

REFERENCES

1. Levin, S.A. (2013). Encyclopedia of Biodiversity (Academic Press), pp. 571–579.
2. LeBoeuf, A.C., Benton, R., and Keller, L. (2013). The molecular basis of social behavior: models, methods and advances. Curr. Opin. Neurobiol. *23*, 3–10. https://doi.org/10.1016/j.conb.2012.08.008.
3. Chen, P., and Hong, W. (2018). Neural circuit mechanisms of social behavior. Neuron *98*, 16–30. https://doi.org/10.1016/j.neuron.2018.02.026.
4. Guo, B., Chen, J., Chen, Q., Ren, K., Feng, D., Mao, H., Yao, H., Yang, J., Liu, H., Liu, Y., et al. (2019). Anterior cingulate cortex dysfunction underlies social deficits in Shank3 mutant mice. Nat. Neurosci. *22*, 1223–1234. https://doi.org/10.1038/s41593-019-0445-9.
5. Bauer, H.F., Delling, J.P., Bockmann, J., Boeckers, T.M., and Schön, M. (2022). Development of sex- and genotype-specific behavioral phenotypes in a Shank3 mouse model for neurodevelopmental disorders. Front. Behav. Neurosci. *16*, 1051175. https://doi.org/10.3389/fnbeh.2022.1051175.
6. Nadler, J.J., Moy, S.S., Dold, G., Trang, D., Simmons, N., Perez, A., Young, N.B., Barbaro, R.P., Piven, J., Magnuson, T.R., and Crawley, J.N. (2004). Automated apparatus for quantitation of social approach behaviors in mice. Gene Brain Behav. *3*, 303–314. https://doi.org/10.1111/j.1601-183X.2004.00071.x.
7. Jabarin, R., Netser, S., and Wagner, S. (2022). Beyond the three-chamber test: toward a multimodal and objective assessment of social behavior in rodents. Mol. Autism. *13*, 41. https://doi.org/10.1186/s13229-022-00521-6.
8. Orefice, L.L., Mosko, J.R., Morency, D.T., Wells, M.F., Tasnim, A., Mozeika, S.M., Ye, M., Chirila, A.M., Emanuel, A.J., Rankin, G., et al. (2019). Targeting Peripheral Somatosensory Neurons to Improve Tactile-Related Phenotypes in ASD Models. Cell *178*, 867–886.e24. https://doi.org/10.1016/j.cell.2019.07.024.
9. Panadeiro, V., Rodriguez, A., Henry, J., Wlodkowic, D., and Andersson, M. (2021). A review of 28 free animal-tracking software applications: current features and limitations. Lab. Anim *50*, 246–254. https://doi.org/10.1038/s41684-021-00811-1.
10. Pereira, T.D., Shaevitz, J.W., and Murthy, M. (2020). Quantifying behavior to understand the brain. Nat. Neurosci. *23*, 1537–1549. https://doi.org/10.1038/s41593-020-00734-z.
11. Lauer, J., Zhou, M., Ye, S., Menegas, W., Schneider, S., Nath, T., Rahman, M.M., Di Santo, V., Soberanes, D., Feng, G., et al. (2022). Multi-animal pose estimation, identification and tracking with DeepLabCut. Nat. Methods *19*, 496–504. https://doi.org/10.1038/s41592-022-01443-0.
12. Pereira, T.D., Tabris, N., Matsliah, A., Turner, D.M., Li, J., Ravindranath, S., Papadoyannis, E.S., Normand, E., Deutsch, D.S., Wang, Z.Y., et al. (2022). SLEAP: A deep learning system for multi-animal pose tracking. Nat. Methods *19*, 486–495. https://doi.org/10.1038/s41592-022-01426-1.
13. Agezo, S., and Berman, G.J. (2022). Tracking together: estimating social poses. Nat. Methods *19*, 410–411. https://doi.org/10.1038/s41592-022-01452-z.
14. Marks, M., Jin, Q., Sturman, O., von Ziegler, L., Kollmorgen, S., von der Behrens, W., Mante, V., Bohacek, J., and Yanik, M.F. (2022). Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments. Nat. Mach. Intell. *4*, 331–340. https://doi.org/10.1038/s42256-022-00477-5.
15. Liu, B., Qian, Y., and Wang, J. (2023). EDDSN-MRT: multiple rodent tracking based on ear detection and dual siamese network for rodent social behavior analysis. BMC Neurosci. *24*, 23. https://doi.org/10.1186/s12868-023-00787-3.
16. Han, Y., Chen, K., Wang, Y., Liu, W., Wang, Z., Wang, X., Han, C., Liao, J., Huang, K., Cai, S., et al. (2024). Multi-animal 3D social pose estimation, identification and behaviour embedding with a few-shot learning framework. Nat. Mach. Intell. *6*, 48–61. https://doi.org/10.1038/s42256-023-00776-5.
17. Romero-Ferrero, F., Bergomi, M.G., Hinz, R.C., Heras, F.J.H., and de Polavieja, G.G. (2019). idtracker.ai: tracking all individuals in small or large collectives of unmarked animals. Nat. Methods *16*, 179–182. https://doi.org/10.1038/s41592-018-0295-5.
18. Walter, T., and Couzin, I.D. (2021). TRex, a fast multi-animal tracking system with markerless identification, and 2D estimation of posture and visual fields. Elife *10*, e64000. https://doi.org/10.7554/eLife.64000.
19. Luxem, K., Mocellin, P., Fuhrmann, F., Kürsch, J., Miller, S.R., Palop, J.J., Remy, S., and Bauer, P. (2022). Identifying behavioral structure from deep variational embeddings of animal motion. Commun. Biol. *5*, 1267. https://doi.org/10.1038/s42003-022-04080-7.
20. Hsu, A.I., and Yttri, E.A. (2021). B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. Nat. Commun. *12*, 5188. https://doi.org/10.1038/s41467-021-25420-x.
21. Goodwin, N.L., Choong, J.J., Hwang, S., Pitts, K., Bloom, L., Islam, A., Zhang, Y.Y., Szelenyi, E.R., Tong, X., Newman, E.L., et al. (2024). Simple Behavioral Analysis (SimBA) as a platform for explainable machine learning in behavioral neuroscience. Nat. Neurosci. *27*, 1411–1424. https://doi.org/10.1038/s41593-024-01649-9.
22. Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), pp. 3634–3640. https://doi.org/10.24963/ijcai.2018/505.
23. Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, *32*. https://doi.org/10.1609/aaai.v32i1.12328.
24. Mitz, A.R., Philyaw, T.J., Boccuto, L., Shcheglovitov, A., Sarasua, S.M., Kaufmann, W.E., and Thurm, A. (2018). Identification of 22q13 genes most likely to contribute to Phelan McDermid syndrome. Eur. J. Hum. Genet. *26*, 293–302. https://doi.org/10.1038/s41431-017-0042-x.

25. Peça, J., Feliciano, C., Ting, J.T., Wang, W., Wells, M.F., Venkatraman, T.N., Lascola, C.D., Fu, Z., and Feng, G. (2011). Shank3 mutant mice display autistic-like behaviours and striatal dysfunction. Nature *472*, 437–442. https://doi.org/10.1038/nature09965.

26. Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019). Two-stream adaptive graph convolutional networks for kkeleton-based action recognition. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12018–12027. https://doi.org/10.1109/CVPR.2019.01230.

27. Zhang, L., Gao, J., Xiao, Z., and Fan, H. (2023). AnimalTrack: A benchmark for multi-animal tracking in the wild. Int. J. Comput. Vis. *131*, 496–513. https://doi.org/10.1007/s11263-022-01711-8.

28. Jiang, Z., Liu, Z., Chen, L., Tong, L., Zhang, X., Lan, X., Crookes, D., Yang, M.H., and Zhou, H. (2023). Detecting and Tracking of Multiple Mice Using Part Proposal Networks. IEEE Transact. Neural Networks Learn. Syst. *34*, 9806–9820. https://doi.org/10.1109/TNNLS.2022.3160800.

29. Chen, Z., Zhang, R., Fang, H.S., Zhang, Y.E., Bal, A., Zhou, H., Rock, R.R., Padilla-Coreano, N., Keyes, L.R., Zhu, H., et al. (2023). AlphaTracker: a multi-animal tracking and behavioral analysis tool. Front. Behav. Neurosci. *17*, 1111908. https://doi.org/10.3389/fnbeh.2023.1111908.

30. Ray, S., and Stopfer, M.A. (2022). Argos: A toolkit for tracking multiple animals in complex visual environments. Methods Ecol. Evol. *13*, 585–595. https://doi.org/10.1111/2041-210x.13776.

31. Pedersen, M., Haurum, J.B., Bengtson, S.H., and Moeslund, T.B. (2020). 3D-ZeF: A 3D zebrafish tracking benchmark dataset. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2423–2433. https://doi.org/10.1109/CVPR42600.2020.00250.

32. Dendorfer, P., Ošep, A.a., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., and Leal-Taixé, L. (2021). MOTChallenge: A benchmark for single-camera multiple target tracking. Int. J. Comput. Vis. *129*, 845–881. https://doi.org/10.1007/s11263-020-01393-0.

33. Huang, G., Liu, Z., Maaten, L.V.D., and Weinberger, K.Q. (2017). Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

34. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

35. Zhou, K., Yang, Y., Cavallaro, A., and Xiang, T. (2019). Omni-scale feature learning for person re-identification. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3701–3711. https://doi.org/10.1109/ICCV.2019.00380.

36. Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., and Wei, Y. (2021). LayerCAM: Exploring hierarchical class activation maps for localization. IEEE Trans. Image Process. *30*, 5875–5888. https://doi.org/10.1109/TIP.2021.3089943.

37. Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. (2019). Mask Scoring R-CNN. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6402–6411. https://doi.org/10.1109/CVPR.2019.00657.

38. Maji, D., Nagori, S., Mathew, M., and Poddar, D. (2022). YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2636–2645. https://doi.org/10.1109/CVPRW56347.2022.00297.

39. Farnebäck, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. Image Analysis, 363–370.

40. Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., and Chen, K. (2022). RTMDet: An empirical study of designing real-time object detectors. Preprint at arXiv. https://doi.org/10.48550/arXiv.2212.07784.

41. Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking. In Proceedings of the European conference on computer vision (ECCV), pp. 466–481. https://doi.org/10.1007/978-3-030-01231-1_29.

42. Hui, T.W., Tang, X., and Loy, C.C. (2021). A lightweight optical flow CNN - Revisiting data fidelity and regularization. IEEE Trans. Pattern Anal. Mach. Intell. *43*, 2555–2569. https://doi.org/10.1109/TPAMI.2020.2976928.

43. Cao, J., Pang, J., Weng, X., Khirodkar, R., and Kitani, K. (2023). Observation-Centric SORT: Rethinking SORT for robust multi-object tracking. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9686–9696. https://doi.org/10.1109/CVPR52729.2023.00934.

44. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. In European Conference on Computer Vision (ECCV), pp. 1–21. https://doi.org/10.1007/978-3-031-20047-2_1.

45. Maggiolino, G., Ahmad, A., Cao, J., and Kitani, K. (2023). Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification. In 2023 IEEE International Conference on Image Processing (ICIP), pp. 3025–3029. https://doi.org/10.1109/ICIP49359.2023.10222576.

46. Aharon, N., Orfaig, R., and Bobrovsky, B.-Z. (2022). BoT-SORT: Robust Associations Multi-Pedestrian Tracking. Preprint at arXiv. https://doi.org/10.48550/arXiv.2206.14651.

47. Cheng, H., Oh, S.W., Price, B., Schwing, A., and Lee, J. (2023). Tracking anything with decoupled video segmentation. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1316–1326. https://doi.org/10.1109/ICCV51070.2023.00127.

48. Padilla-Coreano, N., Batra, K., Patarino, M., Chen, Z., Rock, R.R., Zhang, R., Hausmann, S.B., Weddington, J.C., Patel, R., Zhang, Y.E., et al. (2022). Cortical ensembles orchestrate social competition through hypothalamic outputs. Nature *603*, 667–671. https://doi.org/10.1038/s41586-022-04507-5.

49. Bolya, D., Zhou, C., Xiao, F., and Lee, Y.J. (2019). YOLACT: Real-time instance segmentation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9156–9165. https://doi.org/10.1109/ICCV.2019.00925.

50. Bolya, D., Zhou, C., Xiao, F., and Lee, Y.J. (2022). YOLACT++ Better Real-Time Instance Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. *44*, 1108–1121. https://doi.org/10.1109/TPAMI.2020.3014297.

51. Kaufman, L., and Rousseeuw, P.J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis (John Wiley & Sons), pp. 92–102.

52. Yi, F., Danko, T., Botelho, S.C., Patzke, C., Pak, C., Wernig, M., and Südhof, T.C. (2016). Autism-associated SHANK3 haploinsufficiency causes Ih channelopathy in human neurons. Science *352*, aaf2669. https://doi.org/10.1126/science.aaf2669.

53. Ren, W., Huang, K., Li, Y., Yang, Q., Wang, L., Guo, K., Wei, P., and Zhang, Y.Q. (2023). Altered pupil responses to social and non-social stimuli in Shank3 mutant dogs. Mol. Psychiatr. *28*, 3751–3759. https://doi.org/10.1038/s41380-023-02277-8.

54. Tian, R., Li, Y., Zhao, H., Lyu, W., Zhao, J., Wang, X., Lu, H., Xu, H., Ren, W., Tan, Q.Q., et al. (2023). Modeling SHANK3-associated autism spectrum disorder in Beagle dogs via CRISPR/Cas9 gene editing. Mol. Psychiatr. *28*, 3739–3750. https://doi.org/10.1038/s41380-023-02276-9.

55. Yang, T., Bayless, D.W., Wei, Y., Landayan, D., Marcelo, I.M., Wang, Y., DeNardo, L.A., Luo, L., Druckmann, S., and Shah, N.M. (2023). Hypothalamic neurons that mirror aggression. Cell *186*, 1195–1211.e19. https://doi.org/10.1016/j.cell.2023.01.022.

56. Wei, D., Osakada, T., Guo, Z., Yamaguchi, T., Varshneya, A., Yan, R., Jiang, Y., and Lin, D. (2023). A hypothalamic pathway that suppresses aggression toward superior opponents. Nat. Neurosci. *26*, 774–787. https://doi.org/10.1038/s41593-023-01297-5.

57. Bayless, D.W., Davis, C.H.O., Yang, R., Wei, Y., de Andrade Carvalho, V.M., Knoedler, J.R., Yang, T., Livingston, O., Lomvardas, A., Martins, G.J., et al. (2023). A neural circuit for male sexual behavior and reward. Cell *186*, 3862–3881.e28. https://doi.org/10.1016/j.cell.2023.07.021.

58. Guo, Z., Yin, L., Diaz, V., Dai, B., Osakada, T., Lischinsky, J.E., Chien, J., Yamaguchi, T., Urtecho, A., Tong, X., et al. (2023). Neural dynamics in the limbic system during male social behaviors. Neuron *111*, 3288–3306.e4. https://doi.org/10.1016/j.neuron.2023.07.011.

59. Fan, Z., Chang, J., Liang, Y., Zhu, H., Zhang, C., Zheng, D., Wang, J., Xu, Y., Li, Q.J., and Hu, H. (2023). Neural mechanism underlying depressive-like state associated with social status loss. Cell *186*, 560–576.e17. https://doi.org/10.1016/j.cell.2022.12.033.

60. Wang, F., Zhu, J., Zhu, H., Zhang, Q., Lin, Z., and Hu, H. (2011). Bidirectional control of social hierarchy by synaptic efficacy in medial prefrontal cortex. Science *334*, 693–697. https://doi.org/10.1126/science.1209951.

61. Zhou, T., Zhu, H., Fan, Z., Wang, F., Chen, Y., Liang, H., Yang, Z., Zhang, L., Lin, L., Zhan, Y., et al. (2017). History of winning remodels thalamo-PFC circuit to reinforce social dominance. Science *357*, 162–168. https://doi.org/10.1126/science.aak9726.

62. Li, K., Zhou, T., Liao, L., Yang, Z., Wong, C., Henn, F., Malinow, R., Yates, J.R., 3rd, and Hu, H. (2013). betaCaMKII in lateral habenula mediates core symptoms of depression. Science *341*, 1016–1020. https://doi.org/10.1126/science.1240729.

63. Yang, Y., Cui, Y., Sang, K., Dong, Y., Ni, Z., Ma, S., and Hu, H. (2018). Ketamine blocks bursting in the lateral habenula to rapidly relieve depression. Nature *554*, 317–322. https://doi.org/10.1038/nature25509.

64. Bodla, N., Singh, B., Chellappa, R., and Davis, L.S. (2017). Soft-NMS — Improving object detection with one line of code. Proceedings of the IEEE international conference on computer vision (ICCV), 5561–5569. https://doi.org/10.1109/ICCV.2017.593.

65. Boppana, R., and Halldórsson, M.M. (1992). Approximating maximum independent sets by excluding subgraphs. BIT Numer. Math. *32*, 180–196. https://doi.org/10.1007/BF01994876.

66. Tang, Z., Naphade, M., Liu, M.-Y., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D., and Hwang, J.-N. (2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8797–8806. https://doi.org/10.1109/CVPR.2019.00900.

67. Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., and Soundararajan, P. (2007). The CLEAR 2006 evaluation. Multimodal Technologies for Perception of Humans, 1–44. https://doi.org/10.1007/978-3-540-69568-4_1.

68. Luiten, J., Os Ep, A.A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., and Leibe, B. (2021). HOTA: A higher order metric for evaluating multi-object tracking. Int. J. Comput. Vis. *129*, 548–578. https://doi.org/10.1007/s11263-020-01375-2.

69. McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. Preprint at arXiv. https://doi.org/10.48550/arXiv.1802.03426.

70. Tang, C. (2024). segTracker_supp_materials.zip. figshare. https://doi.org/10.6084/m9.figshare.25341856.v1.

71. Tang, C. (2024). segCluster_supp_materials.zip. figshare. https://doi.org/10.6084/m9.figshare.25341859.v1.

72. Tang, C. (2024). STCS code. figshare. https://doi.org/10.6084/m9.figshare.25594242.v2.

73. Tang, C. (2024). Code/models for benchmarking different animal tracking tools. figshare. https://doi.org/10.6084/m9.figshare.25804654.v2.

74. Tang, C. (2024). Rodents pose tracking results from various algorithms. figshare. https://doi.org/10.6084/m9.figshare.25779546.v3.