scientific reports

Check for updates

OPEN

A high precision method of segmenting complex postures in *Caenorhabditis elegans* and deep phenotyping to analyze lifespan

Bingyue Dong & Weiyang Chen✉

In-depth exploration of the effects of genes on the development, physiology, and behavior of organisms requires high-precision phenotypic analysis. However, the overlap of body postures in group behavior and the similarity of movement patterns between strains pose challenges to accuracy analysis. To address this issue, we designed the WormYOLO model based on the YOLO architecture, which improves the segmentation performance of *C .elegans* and effectively handles overlapping poses in images. In detection and segmentation tasks, WormYOLO performs well on the more overlapping Mating dataset, with its object detection performance improving by 24.1% ($mAP_{0.5:0.95}$) compared to Deep-worm-tracker, and its segmentation performance improving by 9.3% ($mAP_{0.5:0.95}$) compared to WormSwin. In addition, we propose a more accurate novel bending counting algorithm. In experiments, WormYOLO segmented images, followed by a feature point extraction algorithm to identify changes in worm skeleton positions, ultimately quantifying behavioral features with a counting algorithm. We conducted analytical experiments on various mutant strains based on their motion characteristics, investigating behavioral differences among the strains and assessing the correlation between high-dimensional phenotypic traits and relative lifespan.

Model organisms are essential materials for scientific research. Compared to *C. elegans* short lifecycle, easy maintenance and reproduction, and observable traits[1,2]. Analyzing the movement behavior of *C. elegans* allows for screening and evaluating anti-aging drugs[3–5]. Additionally, *C. elegans* exhibit genetic susceptibility and sensitivity to environmental stress, making them suitable for exploring the toxicity and mechanisms of various chemical substances at the organismal level[6,7]. However, the most straightforward experimental quantification of behavior involves calculating the frequency of body bends in *C. elegans*[8].

In recent years, researchers have proposed various methods for *C. elegans* segmentation and bending count. The earliest worm tracking algorithm was proposed by Baek et al.[9], who designed a single-worm tracker. They first calculated the mean and standard deviation of pixel values in the image, set appropriate thresholds, binarized the grayscale image, and then calculated the worm's positional movements in the binary image for quantitative behavioral assessment. Cronin et al.[10] divided the body of the worm into n segments, analyzing the degree of body bending by calculating differences in curvature angles for each segment. Restif et al.[11] introduced measures of swim locomotion that utilize a deformable model approach and a novel mathematical analysis of curvature maps, thereby calculating the number of body bends. Javer et al.[12] developed the multi-target tracking Tierpsy Tracker to monitor *C. elegans*. This method includes an HDF5 video reader for easy accuracy assessment of segmentation, and it uses motion data to describe worm locomotion characteristics. Koopma et al.[13] designed the WF-NTP platform, which can simultaneously analyze multiple behavioral parameters of hundreds of worms, though it often struggles with videos containing dark backgrounds. Zhang et al.[14] proposed an advanced bending count method but did not strictly follow the standard of advancing the count only when the maximum bend is reached in the direction opposite to the previous bend, leading to inaccurate results. Additionally, this method is only applicable for segmentation and behavior analysis of single worms with a simple background.

With the emergence of machine learning and the increasing demands of experiments, the next generation of more accurate *C. elegans* segmentation and tracking methods has also emerged. Hebert et al.[15] proposed a method for estimating challenging postures of *C. elegans*, though it only applies to single worms and requires the use of the Tierpsy Tracker for motion analysis. Banerjee et al.[16] presented "Deep-worm-tracker" for detecting and tracking the behavior of *C. elegans*, utilizing the original YOLOv5 model; however, it exhibits limitations in

School of Cyber Science and Engineering, Qufu Normal University, Qufu, China. ✉email: chenweiyang@qfnu.edu.cn

detecting smaller worm targets. Deserno et al.[17] proposed a more accurate method for detecting and segmenting worm poses, but it requires significant hardware resources, long training times, and slow inference speed.

Our aim is to address these challenges and further enhance the performance of YOLO[18] in worm segmentation while improving the accuracy of the bending count algorithm. To enhance segmentation performance, we first addressed the issue of the small receptive field in the YOLO model by integrating the large kernel convolutions of RepLKNet[19], allowing for better capture of the overall morphology of the worms. Secondly, we tackled the issues of inadequate accuracy and target confusion in small object and crowded object segmentation by proposing Attentional Scale Sequence DySample Fusion (ASDF), which flexibly integrates spatial and scale features. In terms of detail capture, we replaced Concat with the Depthwise Convolution-based Semantics and Detail Infusion (DSDI) Module, which not only preserves more detailed information but also reduces the computational cost of the Semantics and Detail Infusion Module[20,21]. Such detailed information is crucial for capturing subtle changes in worms under a microscope, contributing to improved segmentation accuracy.

To improve the accuracy of bending count, our algorithm strictly follows the standard in WormBook[22], where the count advances when the posterior region behind the pharynx reaches the maximum bend in the opposite direction of the previous count. This counting method is based on the worm's locomotion pattern and authentically reflects peristaltic behavior. Based on these approaches, we developed an algorithm for accurately measuring the movement behavior of worms. We compared our approach with mainstream worm detection and segmentation models, evaluating the performance of WormYOLO in detecting and segmenting worms, followed by validating the robustness of the counting algorithm. Our counting algorithm detected differences in worm movement patterns, which vary unevenly across strains. Additionally, we explored the relationship between parameters such as head bend count, tail bend count, maximum speed, length, and omega duration, and worm lifespan.

Results

The performance of WormYOLO in nematode detection and segmentation tasks

We evaluated mainstream worm detection and segmentation models on three independent datasets: CSB-1 dataset, Mating dataset(MD), and BBBC010[17,23]. These datasets, originating from different laboratories, contain varying amounts and degrees of overlapping *C. elegans* (Fig. 1). In the comparative experiments, we trained the WormYOLO model and the Deep-worm-tracker model under consistent experimental environment and model parameters. The results for the WormSwin model were obtained from the experimental data and paper provided by the authors. The results were primarily reported using the average precision (AP) from the COCO metric[24].

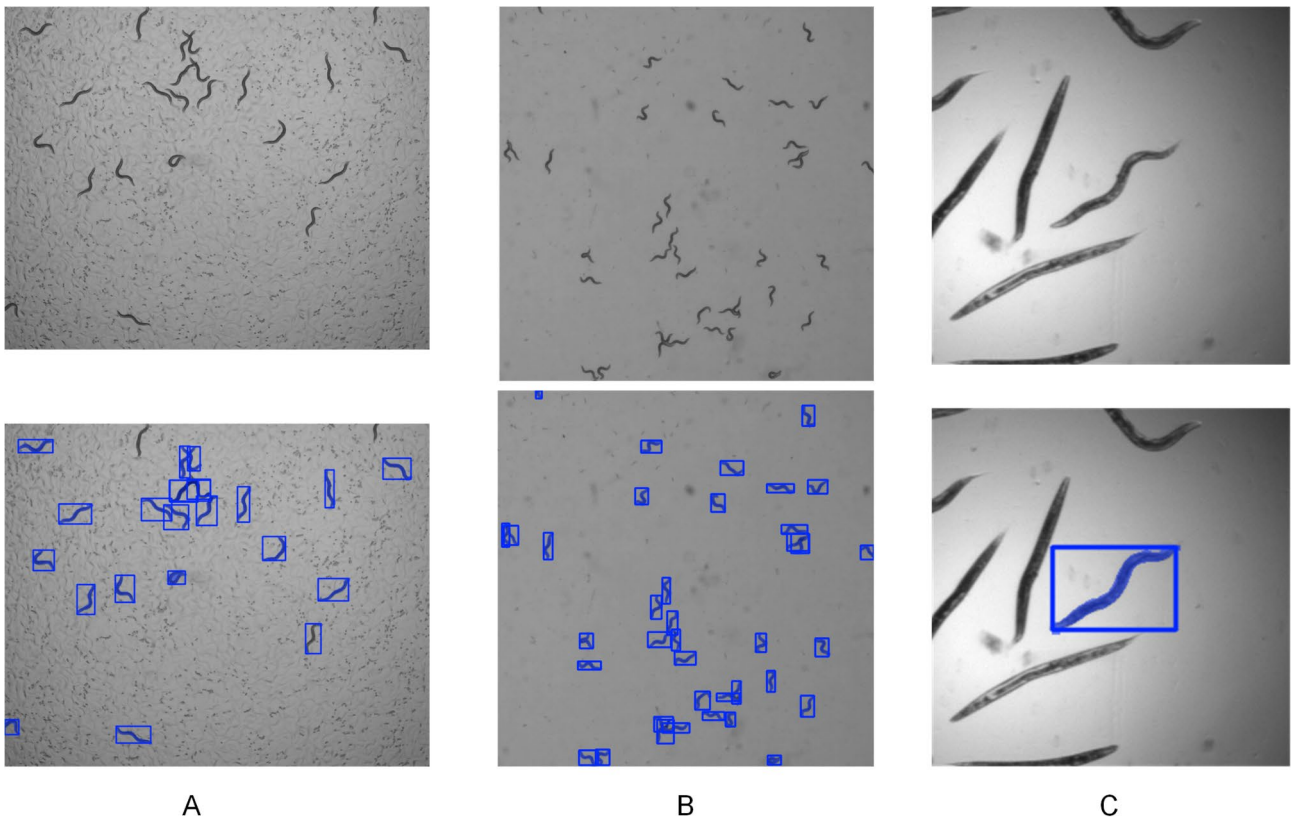


Fig. 1. The dataset images and segmentation examples used in this study. (**A**) The CSB-1 dataset. (**B**) The Mating dataset, which contains more overlapping worms. (**C**) The BBBC010 dataset, where only one worm is annotated in each image.

Dataset	Model	Accuracy (%)	$AP_{0.5}$ (%)	$AP_{0.75}$ (%)	$mAP_{0.5:0.95}$ (%)
CSB-1	Deep-worm-tracker[16](box)	96.8	98.7	94.9	80.9
	WormSwin[17](box)	-	99.0	97.6	81.9
	WormYOLO(box)	**98.5**	**99.2**	**98.0**	**85.6**
	WormSwin[17](mask)	-	99.0	67.5	58.5
	WormYOLO (mask)	**97.6**	**99.0**	**78.8**	**65.6**
MD	Deep-worm-tracker[16](box)	93.6	80.7	75.2	61.7
	WormSwin[17](box)	-	99.0	96.8	83.2
	WormYOLO(box)	**97.7**	**99.1**	**97.9**	**85.8**
	WormSwin[17](mask)	-	98.0	55.1	54.2
	WormYOLO (mask)	**96.9**	**98.3**	**78.0**	**63.5**
BBBC010	Deep-worm-tracker[16](box)	95.1	97.0	82.6	72.1
	WormSwin[17](box)	-	98.5	94.9	82.3
	WormYOLO(box)	**95.2**	**98.9**	**93.6**	**81.6**
	WormSwin[17](mask)	-	96.4	81.5	62.9
	WormYOLO (mask)	**95.9**	**97.1**	**82.6**	**67.8**

Table 1. Test results of various worm detectors and segmenters. "Box" and "mask" refer to the detection accuracy of the bounding box and segmentation mask, respectively. Significant values are in bold.

Task	Way	*p* value
Box	Datasets (CSB-1 VS MD VS BBBC010)	5.20493E-8
	Models (Deep-worm-tracker VS WormSwin VS WormYOLO)	1.43064E-25
	Interaction Effect (Datasets & Models)	3.11711E-26
Mask	Datasets (CSB-1 VS MD VS BBBC010)	4.24821E-5
	Models (WormSwin VS WormYOLO)	6.39207E-9
	Interaction Effect (Datasets & Models)	2.89384E-10

Table 2. Statistical tests on average precision were conducted for both detection and segmentation tasks. The interaction effect refers to the combined influence of the model and dataset on average precision when they are considered in conjunction.

AP is the area under the precision-recall curve, with values between 0 and 1; the higher the AP, the better the performance.

On three datasets, WormYOLO outperformed Deep-worm-tracker by 4.7%, 24.1%, and 9.5% in $mAP_{0.5:0.95}$, respectively, with superior performance in other metrics as well (Table 1). Although WormSwin demonstrates relatively high detection and segmentation capabilities, we found that WormYOLO's performance still showed varying degrees of improvement across the three public datasets. Particularly in the MD dataset, which contains a large number of mating worms with strong overlap and parallel alignment, the segmentation task is quite challenging (Fig. 1B). In this dataset, WormYOLO's segmentation performance outperforms WormSwin by 22.9% ($AP_{0.75}$) and 9.3% ($mAP_{0.5:0.95}$). In addition, the WormSwim model was trained on 4 Nvidia Tesla V100-SMX2 32 GB GPUs, 6 Intel Xeon Gold 6248 CPUs, and 100 GB of memory. In contrast, the WormYOLO model was trained on a single 3090 GPU with 24 GB of processing power and 60 GB of memory. During testing, the WormSwim model had an evaluation speed of 2.7 images/s on the test set, while WormYOLO reached 44.7 images/s. WormYOLO has clear advantages in hardware requirements and inference time, demonstrating good efficiency.

To relate the results to variations between datasets and models, we performed statistical tests on the detection and segmentation tasks separately. In the detection task, we used three models: Deep-worm-tracker, WormSwin, and WormYOLO. In the segmentation task, we used WormSwin and WormYOLO. A two-way analysis of variance (ANOVA) was conducted with the three datasets and different models as independent variables to examine their effects on the results. The results show that all P-values are less than 0.01, indicating significant differences in the performance of different models in detection or segmentation tasks (Table 2). This suggests that the performance improvements of WormYOLO across multiple datasets are valid. Furthermore, the characteristics of different datasets may have an impact on the detection or segmentation performance of each model.

Model	RepLKNet	ASDF + DSDI	$AP_{0.5}$(box)	$AP_{0.75}$(box)	$mAP_{0.5:0.95}$(box)	$AP_{0.5}$(mask)	$AP_{0.75}$(mask)	$mAP_{0.5:0.95}$(mask)
			98.3%	94.1%	79.3%	97.2%	52.9%	54.7%
YOLOV8-N	√		98.8%	95.3%	80.2%	98.0%	65.4%	59.3%
	√	√	**99.0%**	**95.9%**	**81.8%**	**98.5%**	**69.5%**	**62.6%**
			98.2%	94.8%	80.3%	97.5%	62.8%	58.8%
YOLO11-N	√		98.8%	95.0%	80.6%	98.1%	68.6%	61.5%
	√	√	**99.0%**	**95.7%**	**81.3%**	**98.4%**	**67.0%**	**62.7%**

Table 3. Ablation study of YOLOv8-N and YOLO11-N on the CSB-1 dataset. ASDF + DSDI refers to the combined effect of attentional scale sequence dysample fusion (ASDF) and depthwise Convolution-Based semantics and detail infusion (DSDI). Significant values are in bold.

Version	TASK	Model	*p* value
YOLOV8-N	Box	None VS RepLKNet	1.51741E-6
		RepLKNet VS WormYolo	9.16499E-8
	MasK	None VS RepLKNet	1.57975E-5
		RepLKNet VS WormYolo	2.01053E-6
YOLO11-N	Box	None VS RepLKNet	0.00243
		RepLKNet VS WormYolo	4.76615E-5
	MasK	None VS RepLKNet	4.2262E-10
		RepLKNet VS WormYolo	6.33079E-8

Table 4. T-test of Ablation Experiments. Where 'none' refers to the original YOLO model, 'RepLKNet' refers to YOLO + RepLKNet, and 'WormYOLO' refers to YOLO + RepLKNet + ASDF + DSDI.

Ablation study

Table 3 Lists the ablation results based on YOLOv8-n and YOLO11-n. We selected the csb-1 dataset for training, with the experimental environment and model parameters being consistent. The results showed that after enhancing the backbone with RepLKNet, all metrics improved to varying degrees. The $mAP_{0.5:0.95}$ increased by 4.6% and 2.7%, respectively, for YOLOv8-N and YOLO11-N in segmentation tasks. After integrating the DSDI and ASDF modules, all metrics further improved. In the end, in the segmentation task, the model's performance in $mAP_{0.5:0.95}$ improved by 7.9% and 3.9%, compared to the original YOLOv8 and YOLO11, respectively.the t-test report indicates that the introduction of the architecture we proposed is effective for both YOLOv8 and the latest YOLO11 (Table 4). Additionally, it was observed that all datasets in this study were more suited to the YOLO-M model. The report shows that the accuracy of the improved YOLOv8-N and YOLO11-N models is comparable; however, YOLO11 requires more GPU resources when using the M model. Therefore, in comparisons with other detection and segmentation methods, the improved YOLOv8-M model was used.

Evaluate the performance of the feature point extraction algorithm

We improved the feature point extraction algorithm proposed by Zhang et al., which distinguishes the head and tail based on sharpness but is susceptible to environmental influences. Generally, the head exhibits a higher swinging frequency. We added a comparison of the swinging frequencies between the head and tail to enhance its performance. Additionally, our algorithm supports worm identification in different scenarios. For instance, in single-worm scenarios, the worms in the field of view are generally larger, whereas in denser scenes, the worms are smaller. Zhang et al.'s algorithm fails to correctly analyze targets when the ratio of the bounding box area to the image area is less than 1.3% (Fig. 1A, B). In contrast, our algorithm supports worm sizes greater than 0.5%, maintaining good adaptability and performance across typical scenarios. In the evaluation of the feature point extraction algorithm, more than 200,000 images of *C. elegans* were used in the experiments. These images, originating from different laboratories, contained worms of varying sizes and postures. Feature point extraction algorithms were used to identify feature points (head, throat, tail, inflection points, peak points) and skeleton points in consecutive image frames (for detailed definitions, please refer to the Materials and Methods section). Feature points were ignored when evaluating skeleton points.

The reported results primarily include the detection rate and accuracy of key point coordinates across all image sets, as well as the detection rate and accuracy in the absence of self-occlusion behavior. Detection rate refers to the proportion of images where key points were detected, while accuracy measures the degree to which identified key points match the true key points. In this study, a margin of error of less than 1.67% of the worm's body length was allowed for key point positions. The results show that our method demonstrates good generalizability across datasets from different laboratories, achieving a skeleton points detection rate of 99.30% and an accuracy of 98.10% (Table 5). When self-occlusion occurs, the detection rate and accuracy of the feature points slightly decrease. To investigate this, we performed t-test to explore the impact of self-occlusion on the

	Detection rate (no self-occlusion)	Detection rate	Accuracy (no self-occlusion)	Accuracy
Skeleton point	99.30%	99.28%	98.10%	98.06%
Head	99.23%	99.19%	97.20%	96.91%
Pharynx	99.29%	99.24%	98.06%	97.98%
Tail	99.25%	99.20%	97.41%	96.89%
Inflection point	99.11%	98.95%	98.01%	97.23%
Peak point	99.04%	98.94%	97.66%	97.02%

Table 5. Evaluation results of skeleton points and feature points. Detection rate and accuracy of key point coordinates across all image datasets, as well as detection rate and accuracy for datasets without self-occlusion behavior. In the experiments, we allowed an error of less than 1.67% of the Worm's body length between the identified key points and their true positions.

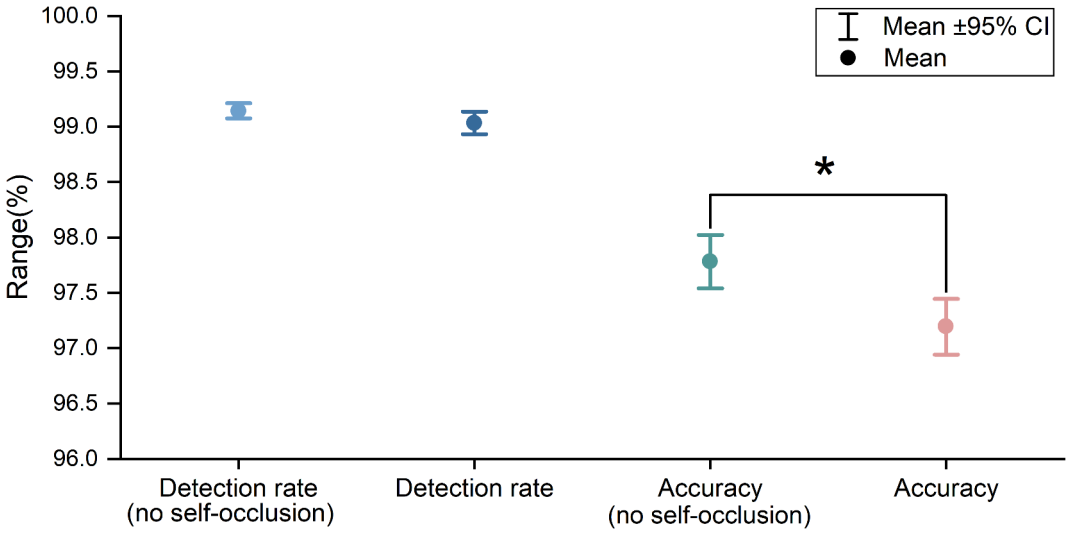


Fig. 2. The impact of self-occlusion on the detection rate and accuracy of all the feature points, with * indicating $p < .05$.

detection rate and accuracy of all feature points. The results indicate that self-occlusion has no significant effect on the detection rate, but it significantly affects the accuracy (Fig. 2).

The robustness of the counting algorithm was verified by manual counting

We randomly selected videos of different strains from the database and extracted 1-minute or 30-second segments to evaluate the proposed algorithm[25].

After extracting key point information using the feature point extraction algorithm, the number of body bends and head and tail bends was estimated by calculating the change in the coordinates of the points. We compared the results of the experiment with the counts of trained researchers. The standards for manual counting were strictly based on the counting methods provided by WormBook[26] for statistical analysis. Every time the part of the worm just behind the pharynx reaches a maximum bend in the opposite direction from the bend last counted, advance the count one. All the automatic counts were concentrated near a red 0-error line (Fig. 3). The number of body bends was lower, closer to the 0-error line (Fig. 3B). The number of head and tail bends were higher, and the error increased slightly (Fig. 3A, C, and D).

In addition, we found that the head and tail were more active than the body, mainly because the body's fluctuations began at the two ends during swimming. The head was bent the most frequently, reflecting the fact that the head of *C. elegans* guides active perception and motor behavior responses.

Reliability of counting algorithms

We compared our method with the methods proposed by the Tierpsy Tracker[27] and Zhang et al.[14]. For an objective comparison, we used the sample data provided by the Tierpsy Tracker to evaluate their detection performance for single worm motion. The background of the video was relatively simple, containing a *C. elegans* that moved sinusoidally. We conducted multiple experiments and found that the Tierpsy Tracker frequently overestimated the true values, while the method in reference[14] often underestimated them.

To explore the differences between the different counting methods, we visualized the motion data generated during an experiment, with one data point per frame, for a total of 9,000 frames of continuous data. Figure 4C

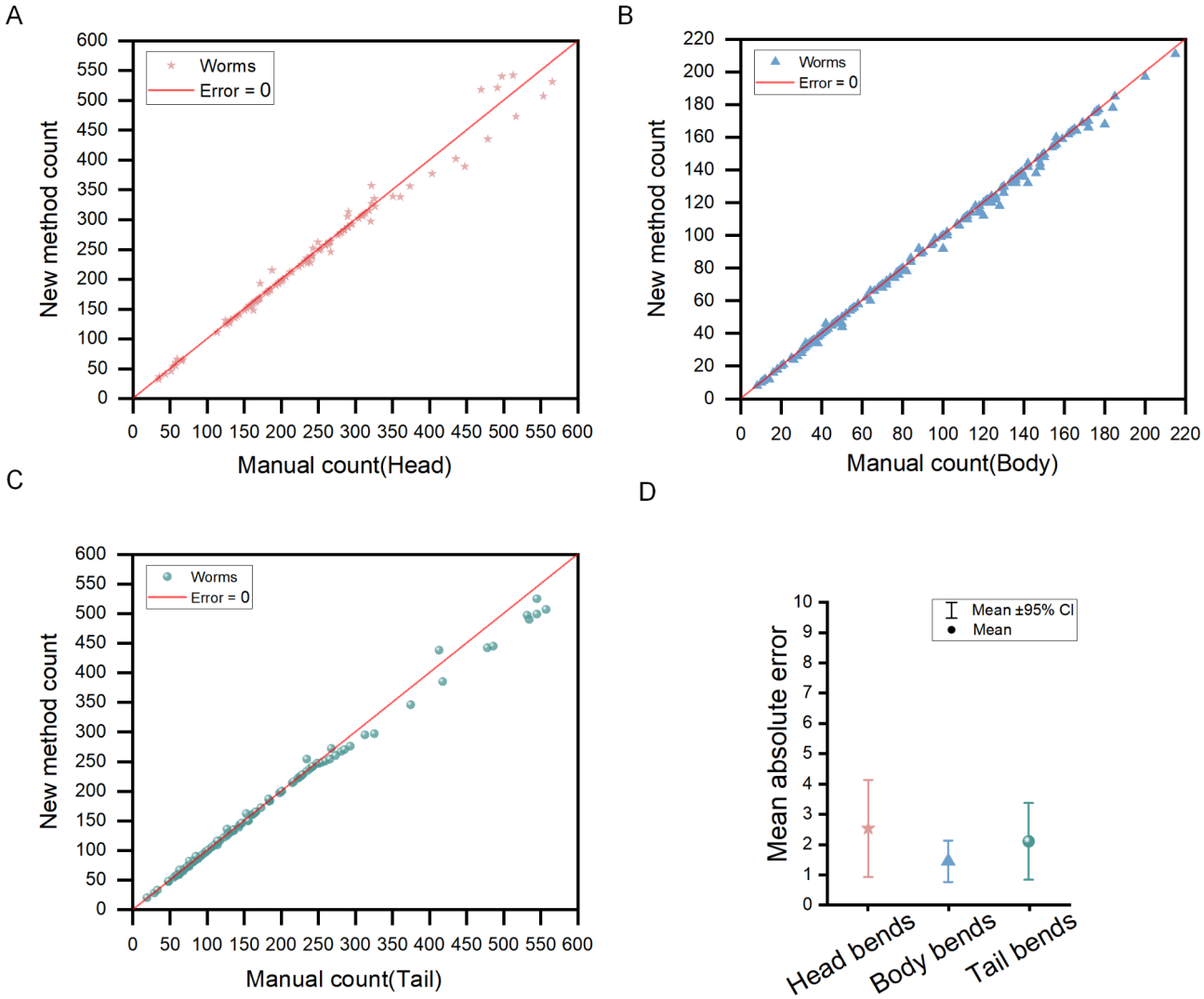


Fig. 3. Manual and automatic counting results of the bending frequency in different parts of the nematode body. (**A**, **B**, and **C**) The number of bends in various parts of the nematode within 1 min or 30-second, counted manually and automatically. The horizontal axis represents the manual counting results, and the vertical axis represents the automatic counting results. The red solid line is the zero-error line. (**D**) The mean absolute error and 95% confidence interval for the bend counts of the head, body, and tail, respectively.

shows the data for our new method, with the vertical axis representing the distance from the maximum peak point to the line connecting the pharynx and the tail. The positive and negative values of the distance represent the bending direction of *C. elegans*. When the peak point is in the counterclockwise direction of the head of *C. elegans*, it is defined as a positive value, and in the clockwise direction, it is a negative value. Different distances reflect the degree of bending in the positive and negative directions. When the part behind the worm's pharynx reaches the maximum bending degree in the opposite direction to the last count, it is recorded as a bend[22]. In the data graphs generated by the method from reference[14], it is evident that the bending frequency is slower (Fig. 4B). This is because when interference data occurs, their algorithm immediately loses the previous bending data and starts recalculating from the data after the interference, ignoring the changes between the data segments immediately before and after the interference. This results in the overall count being less than the actual value. For instance, in frames 1632–1725 in the Fig. 4B, when interference data appears, the distance suddenly drops to zero and starts counting again from zero, losing the count immediately before and after the interference data. However, in the original images of the corresponding frames, the movement state of *C. elegans* clearly changes and should be counted as a bend according to the standard in reference[22] (Fig. 4A(a), (b)).

Our algorithm perfectly skips the interference data, retaining all the bending data, achieving more accurate counting (Fig. 4C). Additionally, they did not count according to the standard in reference[22], but rather set it to count as a bend immediately when the bending direction changes, which may also lead to counting deviations.

In most existing worm Trackers, the bending count is calculated by determining the directional change of the average curvature of the skeleton points of various parts of *C. elegans*[27,29]. Taking the Tierpsy Tracker as an example, we plotted the average curvature data of the midbody in the tracking results for analysis (Fig. 4D). The

positive and negative values of the average curvature represent the direction of the worm's bending, and a bend is recorded when the direction changes. From the data plots of the Tierpsy Tracker method, it is evident that the bending frequency is relatively high. Through motion data analysis, we found that although the movement state of the nematode between certain frames changed little, the Tierpsy Tracker recorded a significant curvature change, resulting in an overall counting result higher than the true value.For example, consider the two frames in Fig. 4A (c) and (d). The worm's posture has hardly changed, but the curvature data in Fig. 4D, corresponding to frames 6030 to 6034, showed that the direction of the curvature had changed, which was evidently inaccurate. This phenomenon may be due to the algorithm's excessive sensitivity to curvature changes. In Fig. 4C, our bending data is all positive, which is more consistent with the movement state of *C. elegans*. However, this is not an occasional phenomenon, as the same situation occurs in many frames, which may be one of the reasons for the overcounting by this algorithm. In addition, We randomly selected several single-worm movement videos from an online database[25] containing different strains to evaluate various counting methods, and the results confirmed the analysis above (Fig. 4E). The accuracy of Tierpsy Tracker is relatively poor, with an average absolute error of 32.7, while the method proposed by Zhang et al. has an error of 4.7. In contrast, our counting algorithm performed excellently in the tests, with an error of only 1.6 (Fig. 4F). We also performed t-test between their algorithm and the algorithm presented in this paper, with $p < .01$. These results demonstrate the reliability of our counting algorithm.

Comparison of counting performance with multi-worm trackers

We selected sample data from Tierpsy Tracker to evaluate tracking performance. Using weights trained on the CSB-1 dataset, we completed the tracking process and generated position files for each detection box (Video files are provided in S1). The evaluation was performed with TrackEval[30], employing metrics such as MOTA, MOTP, and IDF1[31]. The results indicate that WormYOLO performs favorably (Table 6).

Subsequently, we compared the performance of our method with Tierpsy Tracker and WF-NTP in terms of bending count. During the experiment, we first corrected the errors in worm identity switching across all algorithms, and then proceeded with the counting. We found that the head and body bends counting results of the Tierpsy Tracker were consistently higher than the true values (Fig. 5A, C).

The average absolute error for the number of head and body bends was 21.778 and 12.44, respectively. Our errors were 2.146 and 1.389, respectively. The average error in the number of body bends for WF-NTP is 14.383 (Fig. 5B, D). In general, the results of the counting in the Tierpsy Tracker were higher than the real values, but the values of worm2 and worm14 in the figure were much lower than the real values. We inspected the tracking results in the Tierpsy Tracker Viewer and found that when the worms overlap, their method typically identifies them as a single entity, failing to capture accurate skeleton information, resulting in the loss of motion data (Fig. 5E). WF-NTP faces the same issue. In contrast, our method accurately identifies them as two distinct worms in most cases, ensuring the robustness of subsequent analyses.

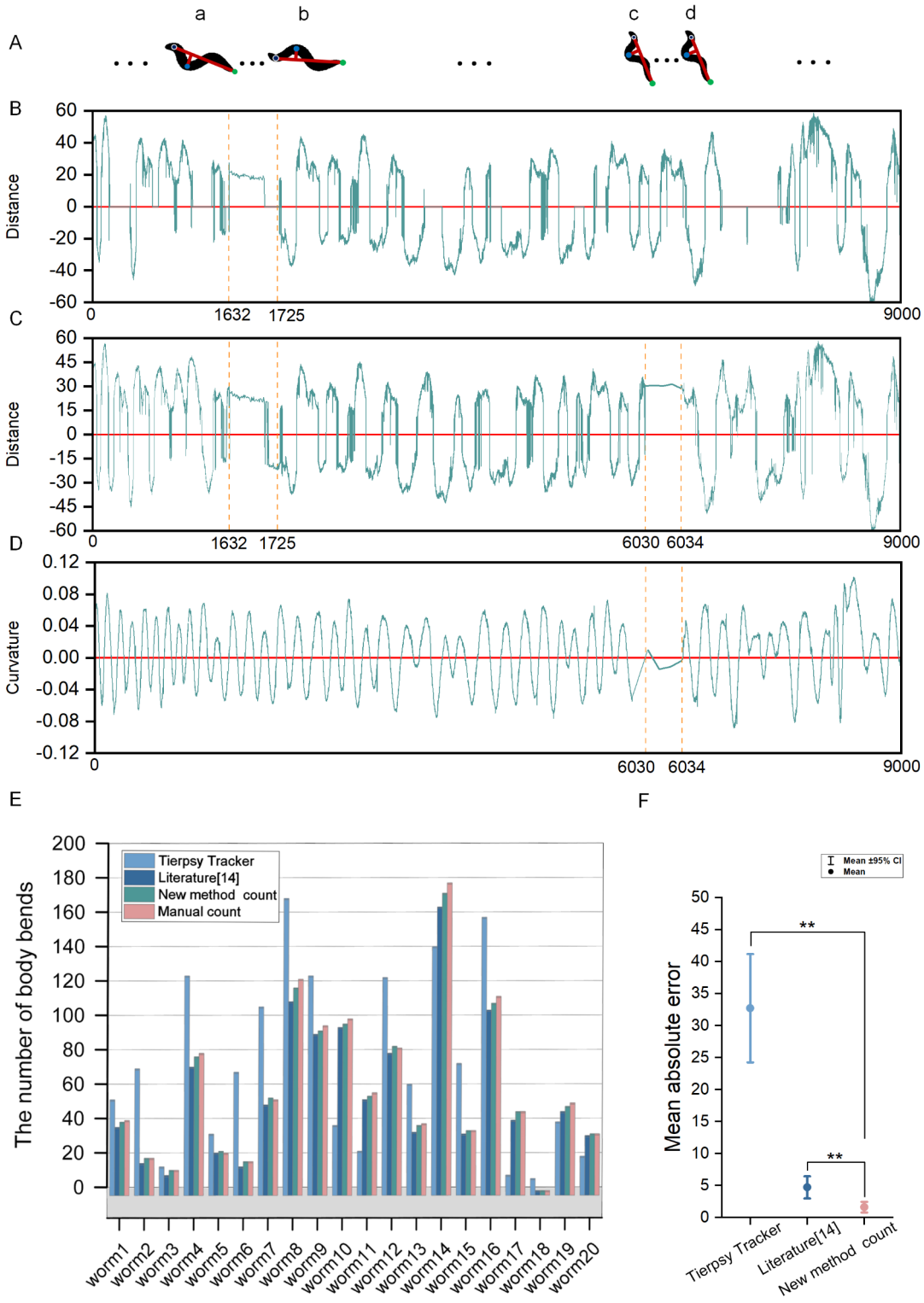
The model is able to effectively capture the movement differences between different strains

To investigate whether the counting algorithm can capture the motion differences between different strains, we selected four strains of *C. elegans* from the database: *unc-4*, *dpy-20*, N2 and LSJ1. We then extracted a 30-second clip from the video for the experiment. To more intuitively show the differences between the strains, we plotted the high-frequency bending and low-frequency bending data as a time-distance curve (Supplementary Fig. s1). The parameters in the graph were roughly the same as those described earlier. The degree of fluctuation of the curve represents the speed of the bending frequency. It has been reported that the dysfunction of *unc-4* motor neurons leads to motor defects[32,33]. *Dpy-20* exhibits motor defects due to excessive contraction of the body wall muscles attached to the external cuticle[34]. Supplementary Fig. S1A, B show that their curves fluctuated very sparsely, indicating a low bending rate. In addition, LSJ1 and N2 are derived from the same wild isolate, but after a long period of cultivation, the two strains have shown differences[35]. Supplementary Fig. S1D, J show that the curve of LSJ1 fluctuates more violently and its speed is higher compared to N2, confirming the results of the literature[25]. The above discussion shows that our model can effectively capture the movement differences between different strains.

We also observed from the video information in the database that the movement patterns of *gpa-16*, *acr-7*, *trp-2*, *npr-12*, *odr-3*, and *flp-16* showed differences compared to N2. Through extensive experiments on these strains, we measured their average bending frequencies as 29.4, 40.1, 44.3, 50.6, 51.2, and 55.4, respectively (Fig. 6). In addition to frequency differences, these strains also showed consistent patterns in low-frequency and high-frequency bending(Supplementary Fig. S1). The analysis indicates that our model can replicate the movement patterns of *C. elegans* to some extent.

Multidimensional phenotypic analysis of *C. elegans* with different lifespans

During the aging process, *C. elegans* undergo degenerative changes at the organ, tissue, cell, and molecular levels[36–38]. These changes not only affect their physiological functions but are also reflected in their phenotype. Multidimensional phenotypes have been shown to classify mutants and accurately reflect the effects of external stimuli[9,39–41]. We hypothesize that multidimensional phenotypic features can also serve as effective indicators for quantifying lifespan. To this end, we selected free-moving video data from 10 strains of *C. elegans*: *unc-60*, *gpa-8*, *daf-3*, *egl-8*, *hcf-1*, *gpa-2*, *trpa-1*, *mir-124*, *pkg-1*, and *ser-1* from the database[25]. Under uniform experimental conditions, we used a counting algorithm to record kinematic features, including body length, head bending speed, body bending speed, tail bending speed, Omega duration, and maximum bending speed. These kinematic parameters are considered biomarkers of the nematode aging process[42–44]. Although body length is closely



related to the aging process, it lacks direct correlation with lifespan[45]. In this study, we focus on the other five features. We used N2 as a control group and generated phenotype fingerprint maps for *C. elegans* with different lifespans (Fig. 7).

The figure shows that the mutants *unc-60*, *gpa-2*, *trpa-1*, *mir-124*, *hcf-1*, *ser-1*, and *pkg-1* are distinguishable from the wild type. Although there are no obvious phenotypic differences between the mutants daf-3, gpa-8, and egl-8 when examined by eye, the model can still distinguish them after careful analysis of the characteristic data (Fig. 7B-D).

phenotypic differences between the mutants *daf-3*, *gpa-8*, and *egl-8* when examined by eye, the model can still distinguish them after careful analysis of the characteristic data (Fig. 7B-D). As expected, the *unc-60*, *daf-3*, and *gpa-8* strains, which have a shorter lifespan than the wild type, have lower comprehensive indicators, while the *gpa-2*, *trpa-1*, *mir-124*, *hcf-1*, *ser-1*, and *pkg-1* strains, which have a longer lifespan, have higher

◀**Fig. 4**. Comparison of counting algorithms with other single-worm Trackers. (**A**) Images of the nematode at different frame numbers in the original video. (a) The posture of the nematode at frame 1632, corresponding to the data in (**B**) and (**C**) at frame 1632. (b) The posture of the nematode at frame 1725, corresponding to the data in (**B**) and (**C**) at frame 1725. (c) The posture of the nematode at frame 6030, corresponding to the data in (**C**) and (**D**) at frame 6030. (d) The posture of the nematode at frame 6034, corresponding to the data in (**C**) and (**D**) at frame 6030. (**B**-**C**) Continuous motion data of the nematode recorded in the experiment. The vertical axis represents the distance from the maximum peak point to the line connecting the pharynx and the tail, while the horizontal axis represents the frame number. The positive and negative values of the distance indicate the bending direction of *C. elegans*. (**B**) Motion data of the nematode tracked using the method proposed by Zhang et al. (**C**) Motion data recorded using our new method. (**D**) Continuous midbody curvature data recorded by the Tierpsy Tracker. The vertical axis represents the average curvature value, with the positive and negative values indicating the direction of the worm's bending. Each change in direction is counted as one bend, while the horizontal axis represents the frame number. (**E**) Multiple videos were randomly selected from public datasets to evaluate the performance of single-worm trackers. (**F**) The average absolute error and t-test results of different trackers in analyzing worm movement behavior. ** denotes $p < .01$.

Video File	MOTA	MOTP	IDF1
Video1	97.36%	70.21%	91.34%
Video2	98.12%	70.19%	99.05%
Video3	91.53%	70.00%	91.44%
Video4	90.89%	70.00%	75.47%

Table 6. Tracking performance of WormYOLO.

comprehensive indicators. However, we found that the omega duration is more random and does not show this correlation. To determine which of the features listed above are related to lifespan, we calculated the Spearman correlation coefficients between each parameter and lifespan (Table 7). Except for omega duration, lifespan is highly correlated with head bend speed, body bend speed, tail bend speed, and maximum bend speed.

The results show that these multi-dimensional phenotypes can distinguish *C. elegans* with different lifespans. Although they were observed to have markedly different dynamics during the process of quantifying lifespans, they had similar correlations with lifespan. Additionally, we found that a weighted combination of head bending speed and maximum bending speed could better predict the lifespan of *C. elegans* (Table 7).

Discussion

In this study, a novel architecture, WormYOLO, is proposed to address several challenges faced by the YOLO series models in worm detection and segmentation tasks. These challenges include insufficient small object detection capabilities, difficulty in segmenting overlapping postures, and a lack of ability to recognize subtle structures. It was demonstrated through comparative evaluation with existing worm detection and segmentation models that WormYOLO exhibits performance advantages, particularly in handling complex scenarios such as overlapping postures (Table 1).

We conducted a comprehensive comparative study of the performance of the new counting algorithm. When tracking the movement of *C. elegans*, WF-NTP and Tierpsy Trackers often lose the identity, posture, and skeleton information of the worm due to body occlusion or complex backgrounds. In contrast, our system is able to retain more complete movement information, providing better support for deeper behavioral research.

Our counting algorithm can sensitively capture and quantify the subtle motion differences between various *C. elegans* strain. These differences are often caused by genetic variations. By comparing the locomotor phenotypes of different strains, researchers can efficiently detect and classify these mutants[46], accelerating the process of gene function identification. This is of great significance for understanding the relationship between genes and locomotor behavior, as well as exploring new gene functions.

To elucidate the correlation between multidimensional phenotypes and lifespan, we attempted to establish a connection using tracked locomotion data. Under our conditions, head bend speed, body bend speed, tail bend speed, maximum bend speed, and lifespan were positively correlated (see Table 7). Lifespan is determined by genes[47], and understanding the genes related to nematode aging can help in understanding the regulatory genes of human aging. This provides intervention targets and novel strategies for studying human aging delay and the prevention and treatment of aging-related diseases.

However, our method does have certain limitations. For instance, although WormYOLO demonstrates better segmentation performance compared to existing methods on the mating dataset, its accuracy is still insufficient for reliable tracking in highly overlapping scenarios (such as the high overlap and parallel positioning between individuals). Additionally, under self-occlusion, feature extraction algorithms may struggle to accurately analyze the worm's posture, which could lead to the loss of shape feature information. In the future, applying posture estimation and identity re-identification techniques to worm research may effectively solve these issues.

In conclusion, we have developed a high-precision, high-efficiency worm segmentation model and proposed a new worm bending count algorithm. This system provides researchers with a powerful tool to analyze worm

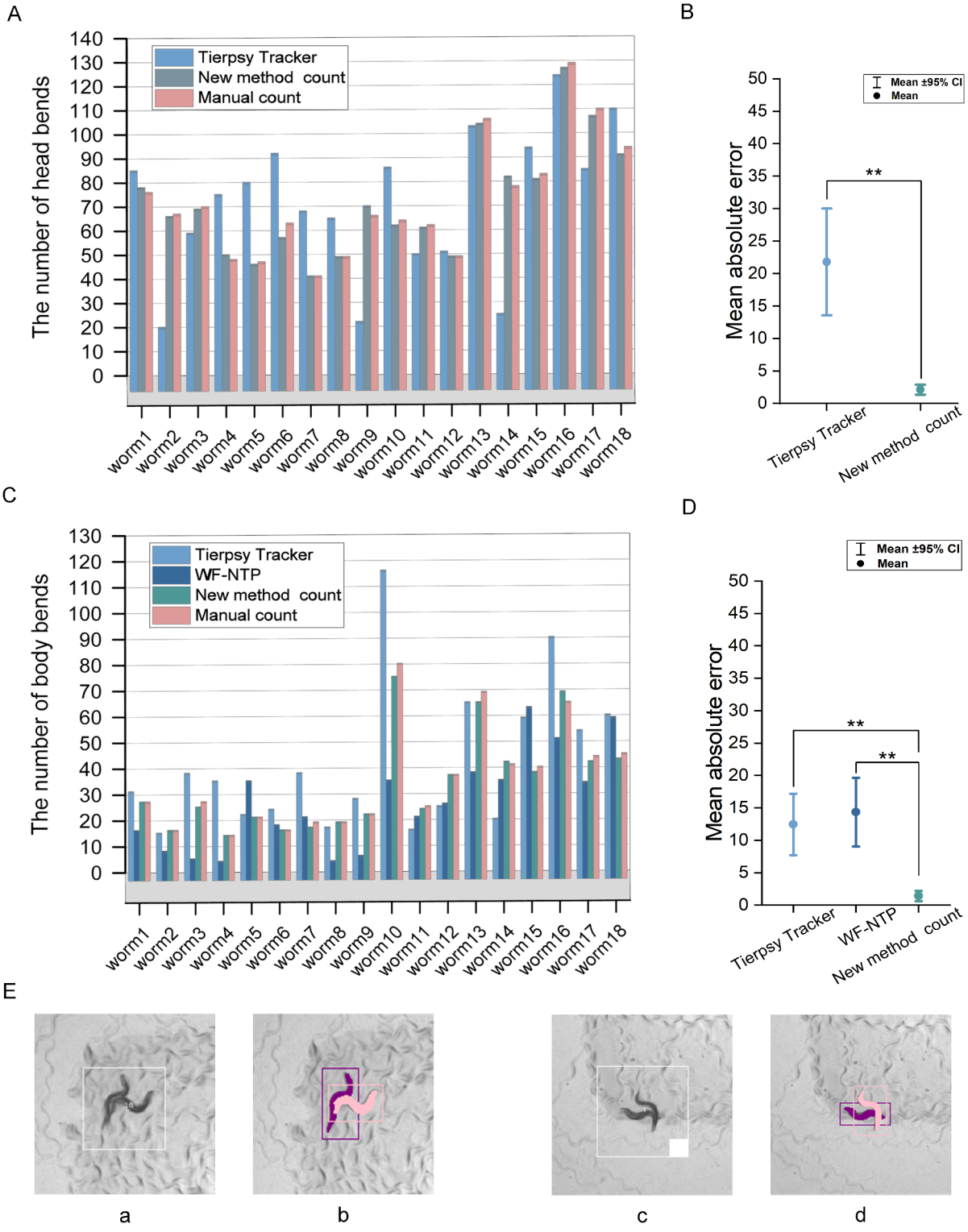


Fig. 5. Performance comparison of multiple worm trackers. (**A** and **C**) Bend count statistics for different methods. (**B** and **D**) Average absolute error and t-test results for different methods, with ** indicating $p < .01$. (**E**) The performance of the Tierpsy tracker and our method in tracking complex movement behaviors. (a) and (c) show the recognition results of the Tierpsy tracker when tracking overlapping and entangled behaviors; (b) and (d) show the recognition results of our method under the same conditions.

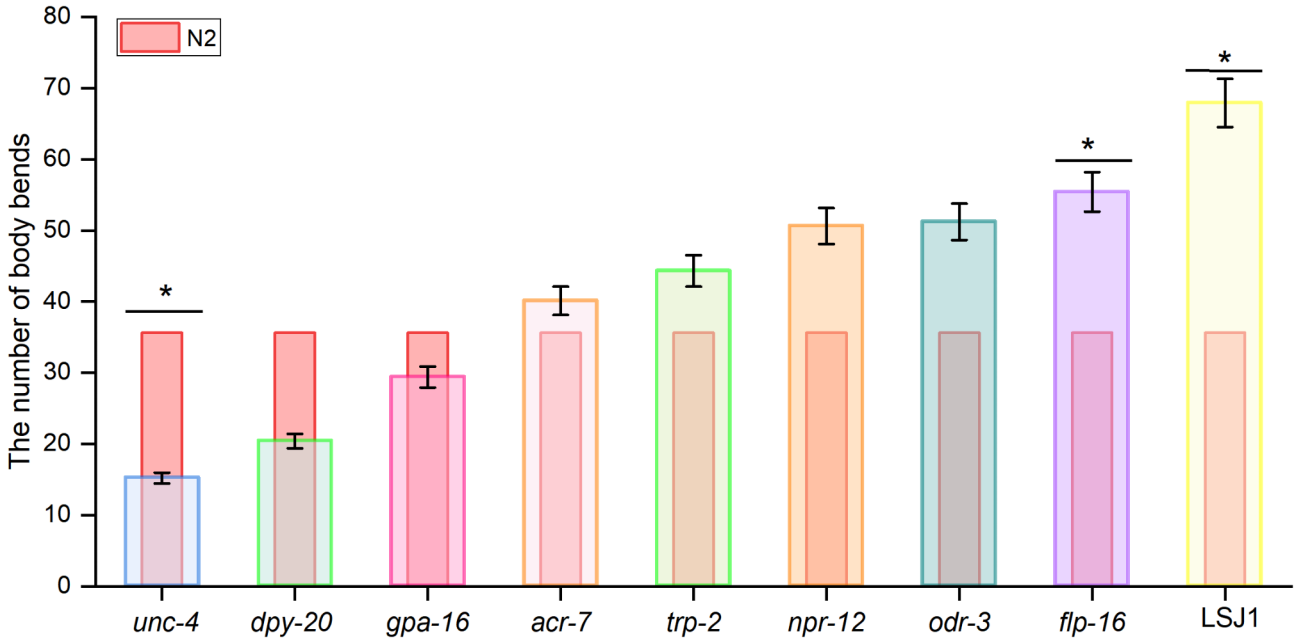


Fig. 6. The average bending counts of multiple worms from different strains within 30 s (8x speed) are listed, with N2 serving as the control group. * indicates a statistically significant difference compared to N2, with $p < .05$.

movement behavior more accurately. At the same time, the advancements in these technologies offer valuable references for tracking and quantifying mating behavior.

Methods

In this section, we outline the procedural framework of the proposed method, as illustrated in Fig. 8. Initially, the WormYOLO model based on deep learning is used to segment worm images to obtain binary images of complete worm bodies in complex scenes. Then, a feature point extraction algorithm is used to mark key points on the worm's centerline. Finally, the maximum distance between several key points and various parts of the body is calculated, and the number of bends and other features are determined based on the changes in these distances.

Datasets

The CSB-1 dataset[17] consists of 10 videos, each approximately 1.5 min long, with a frame rate of 5 Hz and a frame size of 912 × 736 pixels. Nine videos are used for training, and one is used for testing. These videos do not contain any visible petri dish edges, and they feature different backgrounds and varying numbers of worms. The dataset contains annotations for over 60,500 worms. During annotation, the contours of worms located at the edges of the images are ignored.

The Synthetic dataset[17] is generated from the training set of the CSB-1 dataset and contains 10,000 grayscale images, each with a size of 912 × 736 pixels. Over 175,000 worms are annotated in total. Random-sized grayscale rings are added to the center of each image to simulate the edges of the petri dish.

The Mating dataset[17] is derived from a 10-minute long video with a frame rate of 25 Hz and a frame size of 3036 × 3036 pixels. The video contains freely moving and mating worms. During the annotation process, the video sampling rate was reduced to 5 Hz, and 50 frames were randomly selected for annotation. A total of over 3,900 worm contours were annotated, including only mature worms, while worms located at the edges of the images were ignored. Each image has a size of 1012 × 1012 pixels.

The BBBC010 dataset[23] consists of 100 brightfield microscope images, each containing multiple worms that may overlap or cluster. We crop each image to 256 × 256 pixels centered on each worm and annotate only the worm located at the center of the image.

The data used to analyze worm movement behavior were all sourced from the *C. elegans* Behavioral Phenotype Database[25]. It includes clips from different strains, such as N2 (Schafer Lab N2), *unc-4(e120)*, *dpy-20(e1282)*, *gpa-16(ok2349)*, *acr-7(tm863)*, *trp-2(gk298)*, *npr-12(tm1498)*, *odr-3(n2150)*, *fLP-16(ok3085)*, *unc-60(e723)*, *gpa-8(pk345)*, *daf-3(e1376)*, *egl-8(n488)*, *hcf-1(ok559)*, *gpa-2(pk16)*, *trpa-1(ok999)*, *mir-124(n4255)*, *pkg-1(n478)*, *ser-1(ok345)*, and LSJ1. Each video is an 8x speed version of the original, with a total duration of 15 min, a frame rate of 30 Hz, and a resolution of 640 × 480 pixels. Our movement data were measured under these 8x speed conditions.

The Singleworm dataset consists of 324 images of a single worm in different poses, each with a size of 1496 × 1500. It is sourced from https://universe.roboflow.com/robot-cfrpa/unet-1-n7s0g.

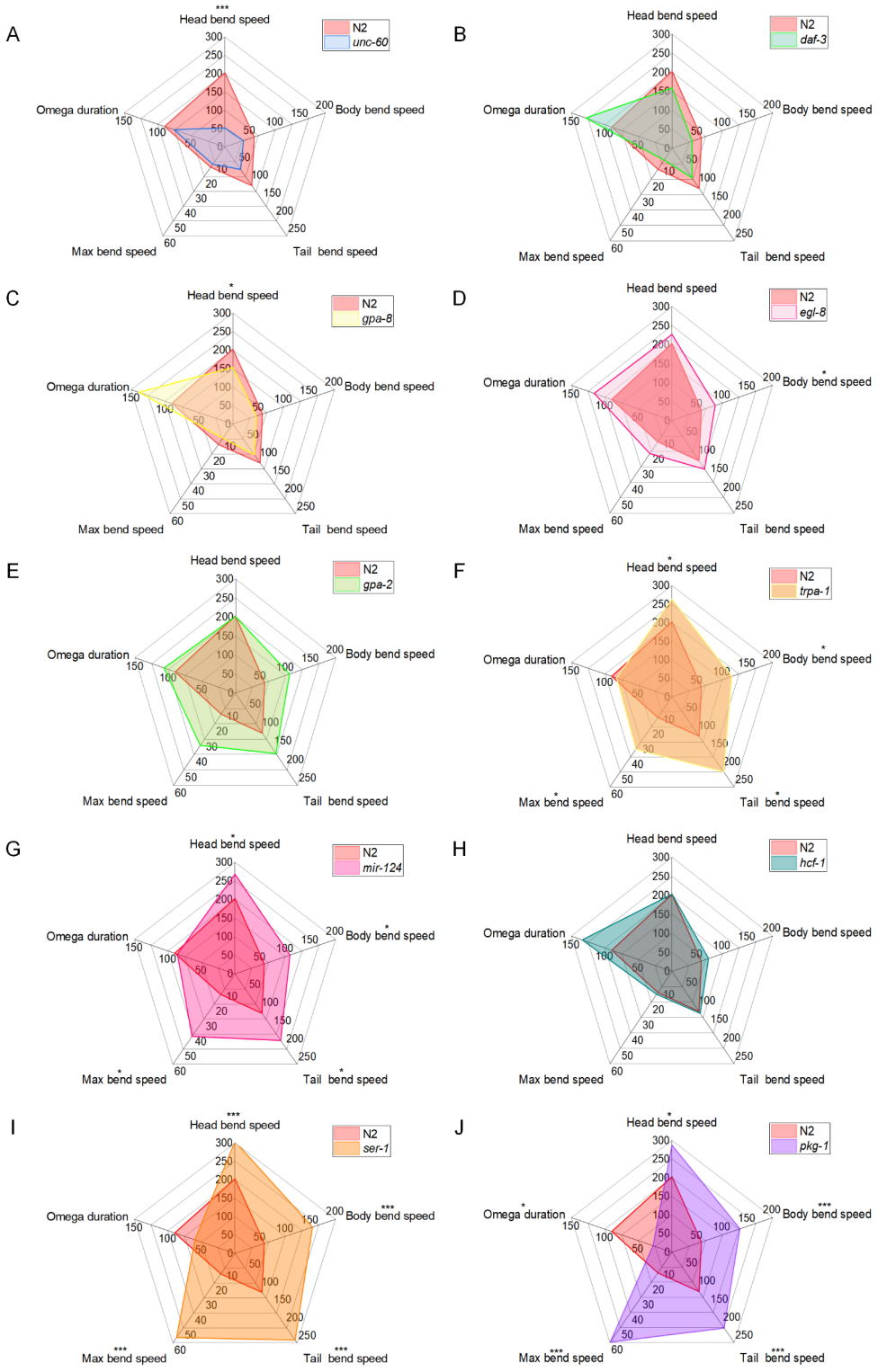


Fig. 7. Relationship between five-dimensional motion phenotype and relative lifespan. (**A-J**) The values corresponding to head velocity, body velocity, and tail velocity represent the average bending frequency of this strain of *C. elegans* within one minute, where the one minute is under 8x speed conditions. A bend occurring within less than five consecutive frames is defined as a rapid bend, and the maximum bend speed is the total number of these rapid bends. Omega duration refers to the total number of frames containing omega bends. The pink area represents the composite metrics measured for N2 in the experiment, while other colors represent the composite metrics of worms with different lifespans. We used N2 as the control group and plotted the composite metrics for *C. elegans* with different lifespans. The larger the area, the higher the composite metrics, with the area size reflecting differences in these metrics. * indicates a statistically significant difference compared to N2, with * indicating $p < .05$, ** indicating $p < .01$, and *** indicating $p < .001$.

	Head bend speed	Body bend speed	Tail bend speed	Max bend speed	Omega duration	Head bend speed & Max bend speed (1:9 ratio)
Coefficient	0.8909	0.9152	0.8788	0.9030	−0.4182	0.9273
p value	1.38E-03	4.67E-04	1.98E-03	8.80E-04	2.32E-01	1.30E-04

Table 7. Spearman correlation between lifespan and different kinematic parameters.

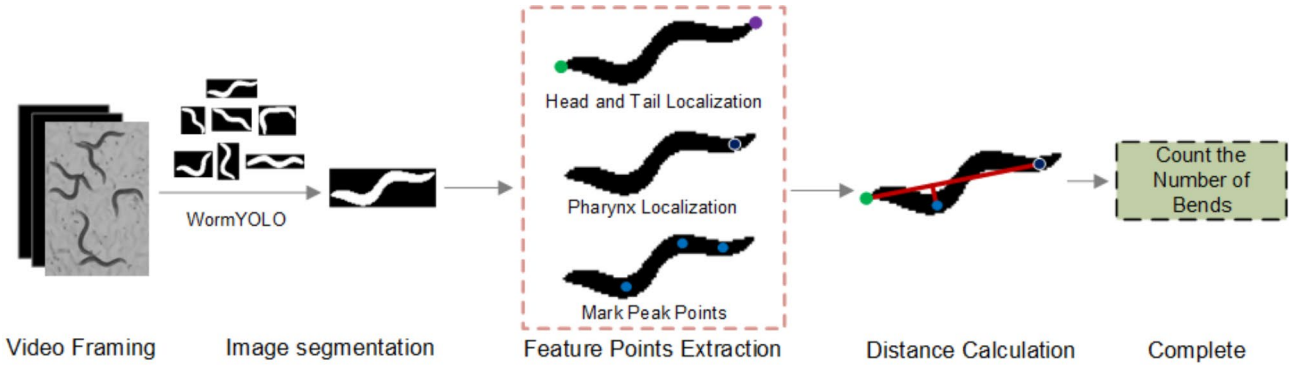


Fig. 8. The flowchart of the proposed method. Tracking algorithm utilizes the WormYOLO model to segment worms in complex scenes. Then, a feature point extraction algorithm is used to analyze the shape and features of the worms. Finally, the phenotypic data are recorded by calculating the change in the position of key points.

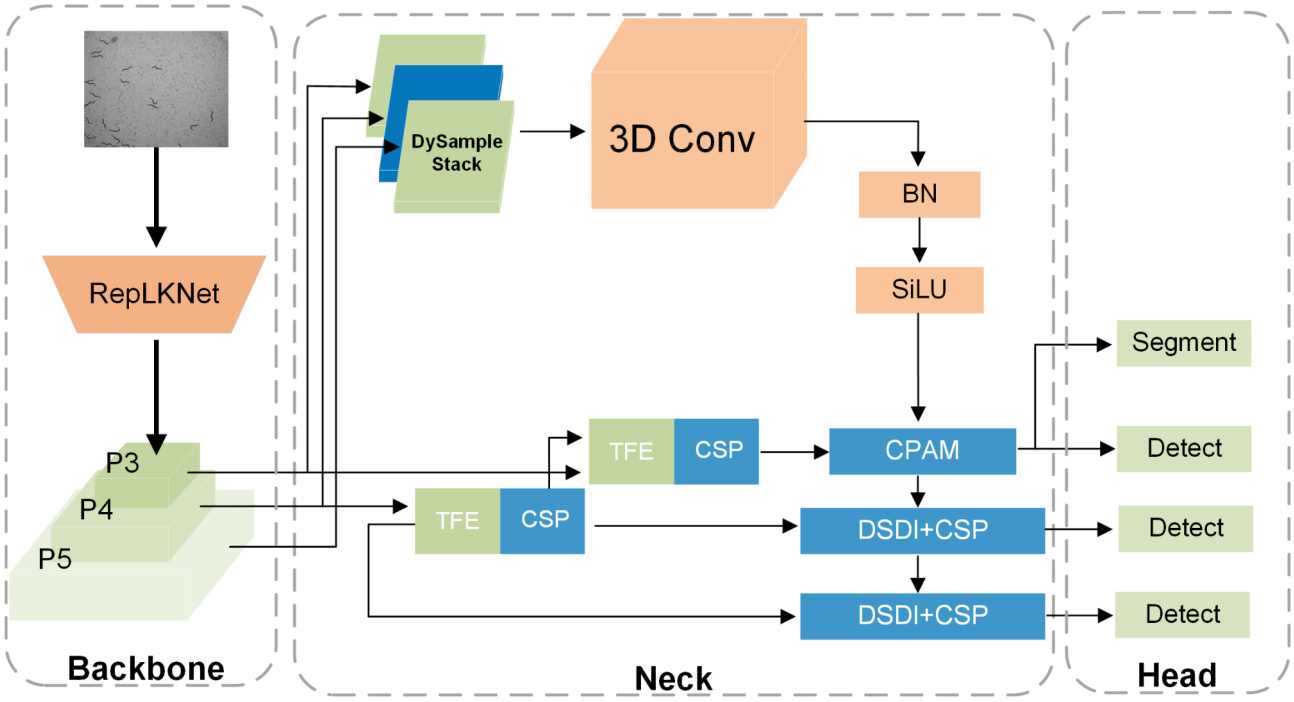


Fig. 9. The network architecture of WormYOLO incorporates RepLKNet as the backbone, integrating the ASDF module and the DSDI module.

Image segmentation algorithm

When using YOLO models for worm detection, information about small worm targets is often lost on the feature map due to a limited receptive field. The varying scales of worms across different developmental stages and the complexity of their posture changes also degrade the detection performance of YOLO models, particularly in imbalanced sample conditions. In segmentation tasks, YOLO fail to adequately retain detailed features, such as the fine structures of the worms. Therefore, a new model architecture, WormYOLO, is proposed (Fig. 9).

To more accurately obtain an effective receptive field, RepLKNet[19] was employed to enhance the backbone of the original model, using re-parameterized large convolutions to establish a broader receptive field. However, large convolution kernels introduce the challenge of a significant number of parameters and floating-point operations. To address the issue of parameter count and floating-point operations, Depthwise (DW) separable convolution is used by RepLKNet. Unlike traditional convolution, each kernel in DW convolution only processes one feature map, reducing both parameter count and computational cost to 1/n (where n is the number of input channels). Pointwise convolution, a 1×1 convolution with n output channels, is then used to compensate for the lack of feature interaction in DW convolution. Without an identity shortcut, deep networks struggle to capture local details, so RepLKNet incorporates identity shortcuts to allow the network to benefit from a larger receptive field while still preserving the ability to capture small-scale patterns. Additionally, RepLKNet employs small kernel re-parameterization to address optimization issues.

In addition to the issue of small object detection, the model structure of YOLO struggles to effectively segment complex poses involving overlap and occlusion in dense scenes. We propose a ASDF method based on the Attentional Scale Sequence Fusion module, which dynamically fuses feature maps from the P3, P4, and P5 layers to enhance the model's performance in multi-scale tasks. Moreover, the Attentional Scale Sequence Fusion module focuses on information channels and small objects related to spatial positions, effectively improving the accuracy of crowded object segmentation.

Finally, we introduced Depthwise Convolution-Based Semantics And Detail Infusion, a module capable of integrating semantic information from high-level features with detailed information from low-level features across feature maps at all levels, thereby capturing intricate details. However, as Semantics And Detail Infusion originally employed standard convolutions, we replaced them with Depthwise separable convolutions to make the module more lightweight and efficient.

Training

We used a synthetic dataset for pretraining, and then loaded the pretrained model to train on the CSB-1 dataset, Mating dataset, BBBC010 dataset, and Singleworm dataset. During the training and testing process, we adjusted the imgsz and epochs parameters based on the actual resolution and size of the dataset. The batch size was set to 8, and the Adam optimizer was used with an initial learning rate of 1e-3. A cosine learning rate scheduler was enabled, and mosaic augmentation was applied throughout all steps. All other parameters were left at their default values without modification.

Tracking

In the experiment, the WormYOLO model uses the BoT-SORT tracking algorithm. The algorithm first obtains the bounding boxes detected by WormYOLO along with their corresponding confidence scores. Next, the Kalman filter is used to predict the next motion state of all targets while awaiting target matching. After matching the detection boxes with the predicted boxes using the Hungarian algorithm, the appearance features provided by the ReID model are used to further refine the matching process. For unmatched targets, the decision to add or remove them is based on the continuous frame loss state, and the final tracking results are output.

We designed two tracking algorithms: single-target tracking and multi-target tracking. The single-target tracking algorithm requires loading the detection or segmentation weights generated by WormYOLO during training. WormYOLO is then used to predict the bounding boxes for each target, extract the regions of interest from the image, and apply OTSU's thresholding method to obtain masks for each region. The multi-target tracking algorithm requires loading the segmentation weights generated by WormYOLO during training, directly predicting the mask for each target.

In this paper, we used the weights trained on the Singleworm dataset for single worm analysis and the weights trained on the CSB-1 dataset for multi-worm analysis.

Feature point extraction algorithm

Upon completion of the segmentation algorithm, a noise-free binary image is obtained, and the search for feature points begins on this image[48] (Fig. 10). The algorithm requires consecutive image frames to identify the feature points. We perform linear interpolation on the contour points of the worm to achieve resampling. The resampled points are denoted as $Pi(i = 1,\ldots n)$. The angle between a contour point and its two neighboring points is calculated as follows

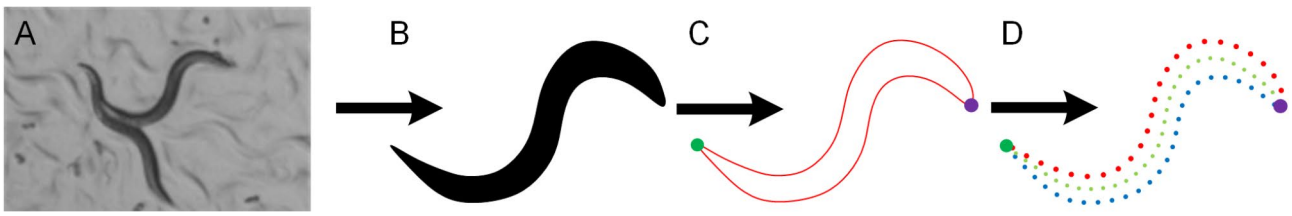


Fig. 10. The process of calculating the coordinates of the head and tail. (**A**) The original grayscale image. (**B**) A clean binary image of the worm. (**C**) The contour of the worm marked with head and tail, with the head indicated by a purple point and the tail by a green point. (**D**) The dorsal and ventral sides of the worm, with red representing the dorsal side and blue the ventral side.

$$T_{k,i} = (P_{i+k} - P_i) \cdot (P_{i-k} - P_i) \doteq l_k^2 \cos\theta_i \tag{1}$$

Where k is the contour point index, l_k is the vector length, and θ_i is the acute angle between two vectors. Since the tail is the sharpest point on the contour of the worm's body, the position of the tail can be found by calculating the angles. However, the accuracy in identifying the sharp point is dependent on the choice of k, where both larger and smaller values of k can affect the precision of the tail's position. To minimize errors, the tail's position should not be simply set as the maximum value of θ_i. The optimal position for P_t is determined through a more nuanced method.

$$P_t = \arg\max_i \left\{ T_{l_1,i} + l_1^2/l_2^2 \cdot T_{l_2,i} \right\}, \quad i \in \{1, \ldots n\} \tag{2}$$

The head of the worm is the sharpest point other than the tail, and the worm head P_h is denoted as

$$P_h = \arg\max_i \left\{ T_{l_1,i} + l_1^2/l_2^2 \cdot T_{l_2,i} \right\}, i \in \{1, \ldots n\} - \{t-a, \ldots t+a\} \tag{3}$$

The range from t-a to $t+a$ represents the restriction area, where the head coordinates are identified by searching the remaining contour points. In the experiment, set $a = n/4$, $l_1 = n/40$, $l_2 = n/100$.

Furthermore, the head has a higher swing frequency. We have added the oscillation frequency to improve the accuracy of identifying the head and tail. After obtaining the coordinates of the head and tail (Fig. 10C), the body contour of *C. elegans* is divided into ventral contour points and dorsal contour points. The ventral contour points are in the clockwise direction from the head, while the dorsal contour points are in the counterclockwise direction. The set of ventral contour points is denoted as $Vi(i = 1,\ldots s)$, and the set of dorsal contour points is denoted as D_j (Fig. 10D).

The midpoint of the line connecting the dorsal point to the corresponding ventral point is noted as the skeleton point of the worm and is expressed using the formula

$$C_i = \frac{1}{2}\left(\mathrm{V}_i + \mathrm{D}_j\right), \quad i = \{1, \ldots s\} \tag{4}$$

where

$$j = \arg\min_m \left\{ \left(\mathrm{V}_{i+k} - \mathrm{V}_{i-k}\right)\left(\mathrm{D}_m - \mathrm{V}_i\right) \right\}, m = \{i-a, \ldots i+a\} \tag{5}$$

Where k is the contour point index, m is the range of traversed contour points, and a is set to 10. After each skeleton point of the worm is determined, the distance between neighboring skeleton points is accumulated to obtain the length L of the skeleton line, which is expressed as

$$L = \sum_{i=1}^{n-1} |C_{i+1} - C_i| \tag{6}$$

In the experiment, the number of skeleton points n was set to 100. Given that the body length of *C. elegans* is approximately 1 mm, and the length of the pharynx is about 100 μm. The coordinates of the pharynx, denoted as p, can be calculated based on its proportional position within the body (Fig. 11A).

$$p = C_{\left[\frac{L}{10}\right]} \tag{7}$$

C. elegans moves forward by propagating dorsal-ventral body bending waves from head to tail. The real-time curvature at each point along the body's centerline can quantify the worm's bending state. First, the curvature

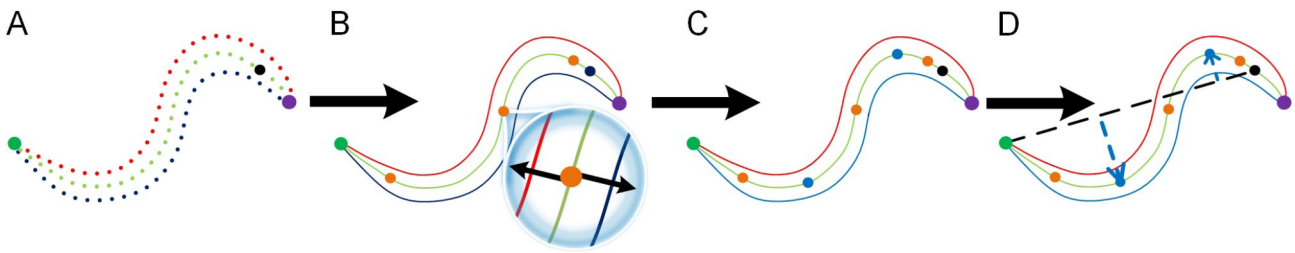


Fig. 11. The feature point extraction algorithm. (**A**) The pharynx of the worm is denoted by a black dot. (**B**) Inflection points along the worm body's central line are marked with orange dots. (**C**) Peak points along the worm body's central line are indicated by blue dots. (**D**) A black dashed line represents the connection between the worm's pharynx and tail, with blue dashed lines denoting the distance from the peak points to the black line. Blue arrows symbolize the direction of bending of various parts of the body.

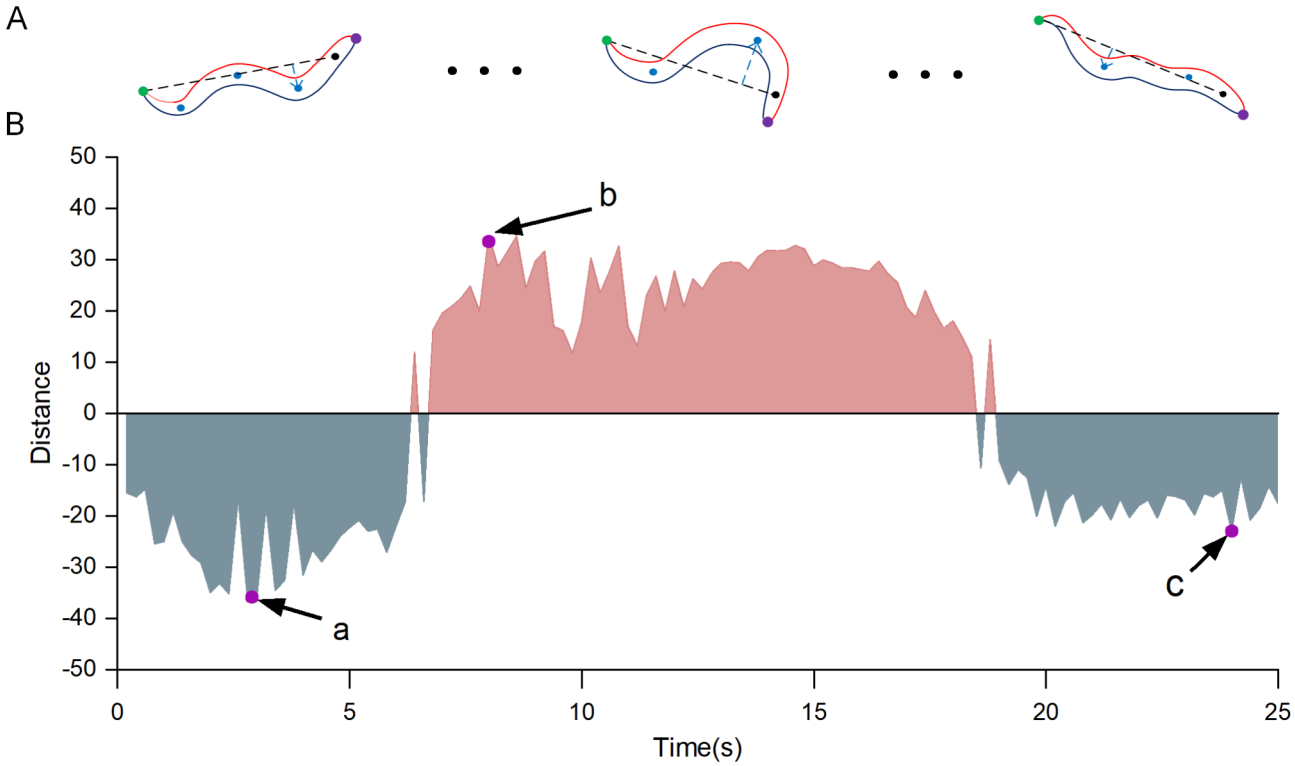


Fig. 12. Changes in the body posture of the worm and the corresponding degree of body bending within 25 s. (**A**) The state of the worm's body with each count of bending. (**B**) The variation in the worm's body curvature progresses with the passage of time. The degree of bending is represented by the maximum distance from the peak points to the line connecting the pharynx and tail. The red section indicates the maximum distance starting from the head in a counterclockwise direction, denoted by a positive sign; the green section represents the maximum distance starting from the head in a clockwise direction, marked by a negative sign.

radius ($R = 1/k$) is used to determine the curvature k for each point. Then, by normalizing the curvature values, k is represented as follows

$$K = \kappa(i) \cdot \frac{d_i}{L} \tag{8}$$

Where d is the actual distance of each skeleton point from the head of the worm and L is the total body length. During continuous motion, *C. elegans* exhibits positive and negative curvature across different segments of its body. When the sign of the curvature changes, it is marked as an inflection point (Fig. 11B). Thereafter, we analyze the curvature of each point along the centerline of *C. elegans*' body to identify the point exhibiting the maximum absolute curvature value situated between two consecutive inflection points, which is then marked as the peak point (Fig. 11C). The total count of peak points is directly related to the number of inflection points identified in the preceding step, implying that a peak point is located between every pair of adjacent inflection.

Calculation of movement characteristics

This research employs statistical analysis based on the method for counting body bends in *C. elegans* as delineated in WormBook[22]. Following this, a bend is recorded when the segment of *C. elegans* posterior to the pharynx exhibits its maximal curvature in a direction contrary to the previous count. The initial step involves drawing a linear through the pharynx and tail of *C. elegans*. Subsequently, the perpendicular distance from each peak point of curvature to this baseline is measured (Fig. 11D).

Furthermore, to accurately identify the direction of bending, commencing from the head, peak points positioned counterclockwise to the line linking the pharynx and tail are classified as demonstrating a positive orientation, denoted by a positive perpendicular distance. Conversely, peak points in a clockwise direction are defined as exhibiting a negative orientation, with the perpendicular.

distance assigned a negative value. During continuous motion, each body region alternately exhibits positive and negative distances. A body bend is recorded when the direction of these values changes and reaches its maximum variation. Figure 12A illustrates the correlation between the worm's body posture and distance changes. Figure 12B displays the maximum distance for each frame over a continuous 25 s video, with counterclockwise direction corresponding to the red area, and clockwise direction to the green area. The transition from point (a) to point (b) is marked as the first bend, and the change from point (b) to point (c) as the second bend, each corresponding to the worm's motion state in Fig. 12A. During the second bend, point (b) represents the peak of

maximum distance in the counterclockwise direction, and point (c) represents the peak of maximum distance in the clockwise direction, strictly adhering to the criterion of reaching "the maximum bend in the opposite direction to the last count". When noise data are present, a voting mechanism is employed to reduce errors. We recorded body bends within fewer than 5 consecutive frames as one instance of maximum bend speed.

The method for counting head and tail bends is similar to the one described above. The head and tail each account for 1/6 of the body length. Using a straight line to connect the respective start and end coordinates, we then identify the peak points of the head and tail positions. The maximum distance between the peak points and the line is calculated, and by analyzing the changes in this maximum distance, the count is completed.

If the distance between the head and tail reached less than 20% of the animal's body length, the turn was classified as having a deep omega-like bend[25]. We statistically determined the number of frames during which *C. elegans* exhibited omega bends to determine the duration.

Data availability

All relevant data used in this paper is available on Google Drive at https://drive.google.com/drive/folders/1V7geOFGjwCFrkeGOdRbtrBTB32BJo87C? dmr=1&ec=wgc-drive-hero-goto. The code is available on GitHub at https://github.com/1490560350/WormYOLO-Count.

Received: 13 August 2024; Accepted: 7 March 2025
Published online: 14 March 2025

References

1. Giacomotto, J. & Ségalat, L. High-throughput screening and small animal models, where are we? *Br. J. Pharmacol.* **160**, 204–216 (2010).
2. Byerly, L., Cassada, R. C. & Russell, R. L. The life cycle of the nematode Caenorhabditis elegans. I. Wild-type growth and reproduction. *Dev. Biol.* **51**, 23–33. https://doi.org/10.1016/0012-1606(76)90119-6 (1976).
3. Alavez, S., Vantipalli, M. C., Zucker, D. J., Klang, I. M. & Lithgow, G. J. Amyloid-binding compounds maintain protein homeostasis during ageing and extend lifespan. *Nature* **472**, 226–229 (2011).
4. Habchi, J. et al. An anticancer drug suppresses the primary nucleation reaction that initiates the production of the toxic Aβ42 aggregates linked with Alzheimer's disease. *Sci. Adv.* **2**, e1501244 (2016).
5. Kim, E. J. E. & Lee, S. J. V. Recent progresses on anti-aging compounds and their targets in Caenorhabditis elegans. *Translational Med. Aging.* **3**, 121–124 (2019).
6. Wang, Y., Liu, S. S., Huang, P., Wang, Z. J. & Xu, Y. Q. Assessing the combined toxicity of carbamate mixtures as well as organophosphorus mixtures to Caenorhabditis elegans using the locomotion behaviors as endpoints. *Sci. Total Environ.* **760**, 143378 (2021).
7. McDonough, C. M., Guo, D. J. & Guo, T. L. Developmental toxicity of bisphenol S in Caenorhabditis elegans and NODEF mice. *Neurotoxicology* **87**, 156–166 (2021).
8. Gidalevitz, T., Krupinski, T., Garcia, S. & Morimoto, R. I. Destabilizing protein polymorphisms in the genetic background direct phenotypic expression of mutant SOD1 toxicity. *PLoS Genet.* **5**, e1000399 (2009).
9. Baek, J. H., Feng, C. P., Silver, Z. & Schafer, J. Using machine vision to analyze and classify Caenorhabditis elegans behavioral phenotypes quantitatively. *J. Neurosci. Methods*. **118**, 9–21 (2002).
10. Cronin, C. J. et al. An automated system for measuring parameters of nematode sinusoidal movement. *BMC Genet.* **6**, 1–19 (2005).
11. Restif, C. et al. CeleST: computer vision software for quantitative analysis of C. elegans swim behavior reveals novel features of locomotion. *PLoS Comput. Biol.* **10**, e1003702 (2014).
12. Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. High-throughput behavioral analysis in C. elegans. *Nat. Methods*. **8**, 592–598 (2011).
13. Koopman, M. et al. Assessing motor-related phenotypes of Caenorhabditis elegans with the wide field-of-view nematode tracking platform. *Nat. Protoc.* **15**, 2071–2106 (2020).
14. Zhang, H. & Chen, W. Automated recognition and analysis of body bending behavior in C. elegans. *BMC Bioinform.* **24**, 175 (2023).
15. Hebert, L. et al. Image synthesis and convolutional networks for pose Estimation in C. elegans. *PLoS Comput. Biol.* **17**, e1008914 (2021).
16. Banerjee, S. C., Khan, K. A. & Sharma, R. Deep-worm-tracker: deep learning methods for accurate detection and tracking for behavioral studies in C. elegans. *Appl. Anim. Behav. Sci.* **266**, 106024 (2023).
17. Deserno, M., Bozek, K. & WormSwin Instance segmentation of C. elegans using vision transformer. *Sci. Rep.* **13**, 11021 (2023).
18. Redmon, J. & Farhadi, A. YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271 (2017).
19. Ding, X., Zhang, X., Han, J. & Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11963–11975 (2022).
20. Wang, C. F. A basic introduction to separable convolutions. *Towards Data Sci.* **13**, 13 (2018).
21. Peng, Y., Sonka, M. & Chen, D. Z. U-Net v2: Rethinking the skip connections of U-Net for medical image segmentation. *arXiv preprint arXiv:2311.17791* (2023).
22. Girard, L. R. et al. WormBook: the online review of Caenorhabditis elegans biology. *Nucleic Acids Res.* **35**, D472–D475 (2007).
23. Ljosa, V., Sokolnicki, K. L. & Carpenter, A. E. Annotated high-throughput microscopy image sets for validation. *Nat. Methods*. **9**, 637–637 (2012).
24. Lin, T. Y. et al. Microsoft Coco: common objects in context. *Comput. Vision–ECCV 2014*, **8693**, 740–755 (2014).
25. Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. & Schafer, W. R. A database of C. elegans behavioral phenotypes. *Nat. Methods*. **10**, 877 (2013).
26. Schindelman, G., Fernandes, J. S., Bastiani, C. A., Yook, K. & Sternberg, P. W. Worm phenotype ontology: integrating phenotype data within and beyond the C. elegans community. *BMC Bioinform.* **12**, 1–16 (2011).
27. Javer, A. et al. An open-source platform for analyzing and sharing worm-behavior data. *Nat. Methods*. **15**, 645–646 (2018).
28. Barlow, I. et al. Megapixel camera arrays for high-resolution animal tracking in multiwell plates. *Commun. Biology*. **5**, 253 (2021).
29. Wang, S. J. & Wang, Z. W. Track-a-worm, an open-source system for quantitative assessment of C. elegans locomotory and bending behavior. *PloS One*. **8**, e69653 (2013).
30. Luiten, J. et al. A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vision*. **129**, 548–578 (2021). Hota.
31. Bernardin, K. & Stiefelhagen, R. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing* 1–10 (2008). (2008).

32. White, J., Southgate, E. & Thomson, J. Mutations in the Caenorhabditis elegans unc–4 gene alter the synaptic input to ventral cord motor neurons. *Nature* **355**, 838–841 (1992).
33. Lickteig, K. M. et al. Regulation of neurotransmitter vesicles by the homeodomain protein UNC-4 and its transcriptional corepressor UNC-37/Groucho incaenorhabditis elegans cholinergic motor neurons. *J. Neurosci.* **21**, 2001–2014 (2001).
34. Lewis, J. A., Wu, C., Berg, H. & Levine, J. H. The genetics of levamisole resistance in the nematode Caenorhabditis elegans. *Genetics* **95**, 905–928 (1980).
35. Gimond, C. et al. Natural variation and genetic determinants of Caenorhabditis elegans sperm size. *Genetics* **213**, 615–632 (2019).
36. Syntichaki, P., Troulinaki, K. & Tavernarakis, N. eIF4E function in somatic cells modulates ageing in Caenorhabditis elegans. *Nature* **445**, 922–926 (2007).
37. Koyuncu, S. et al. Rewiring of the ubiquitinated proteome determines ageing in C. elegans. *Nature* **596**, 285–290 (2021).
38. Hsin, H. & Kenyon, C. Signals from the reproductive system regulate the lifespan of C. elegans. *Nature* **399**, 362–366 (1999).
39. Parida, L. The locomotory characteristics of Caenorhabditis elegans in various external environments: A review. *Appl. Anim. Behav. Sci.* **255**, 105741 (2022).
40. Martineau, C. N., Brown, A. E. & Laurent, P. Multidimensional phenotyping predicts lifespan and quantifies health in Caenorhabditis elegans. *PLoS Comput. Biol.* **16**, e1008002 (2020).
41. Javer, A., Ripoll-Sanchez, L. & Brown, A. E. Powerful and interpretable behavioural features for quantitative phenotyping of C. elegans. *Philos. Trans. R Soc. Lond. B Biol. Sci.* **373**, 20170375 (2018).
42. Son, H. G., Altintas, O., Kim, E. J. E., Kwon, S. & Lee, S. J. V. Age-dependent changes and biomarkers of aging in Caenorhabditis elegans. *Aging Cell.* **18**, e12853 (2019).
43. Shen, L. L., Du, M., Lin, X. F., Cai, T. & Wang, D. Y. Genes required for the functions of olfactory AWA neuron regulate the longevity of Caenorhabditis elegans in an Insulin/IGF signaling-dependent fashion. *Neurosci. Bull.* **26**, 91 (2010).
44. Zhao, B., Khare, P., Feldman, L. & Dent, J. A. Reversal frequency in Caenorhabditis elegans represents an integrated response to the state of the animal and its environment. *J. Neurosci.* **23**, 5319–5328 (2003).
45. McCulloch, D. & Gems, D. Body size, Insulin/IGF signaling and aging in the nematode Caenorhabditis elegans. *Exp. Gerontol.* **38**, 129–136 (2003).
46. Pham, T. D. Classification of caenorhabditis elegans locomotion behaviors with eigenfeature-enhanced long short-term memory networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **20**, 206–216 (2022).
47. Uno, M. & Nishida, E. Lifespan-regulating genes in C. elegans. *NPJ Aging Mech. Disease.* **2**, 1–8 (2016).
48. Dong, X., Song, P. & Liu, X. An automated microfluidic system for morphological measurement and size-based sorting of C. elegans. *IEEE Trans. Nanobiosci.* **18**, 373–380 (2019).

Acknowledgements

AcknowledgementsThis work is supported by the National Natural Science Foundation of China - Tianyuan Fund for Mathematics (12026210).

Author contributions

B.D. planned and conducted experiments, carried out data analysis, wrote and co-edited the paper. W.C. supervised the research, assisted with experimental planning and data analysis, and co-edited the paper.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-93533-0.

Correspondence and requests for materials should be addressed to W.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.