UNIVERSITÄT
ZU KÖLN

Learning Spatiotemporal Representations of C. elegans From Bright-Field Microscopy Data Using Deep Learning Methods

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Maurice Deserno

aus

Aachen, Deutschland

Jahr der Veröffentlichung: 2025

Berichterstattende (Gutachter):	Prof. Dr. Katarzyna Bozek
	Prof. Dr. Oya Beyan
Prüfungsvorsitzender:	Prof. Dr. Andreas Beyer
Tag der mündlichen Prüfung:	16. Mai 2025

Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind als solche kenntlich gemacht. Ich versichere an Eides statt, dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen und eingebundenen Artikeln und Manuskripten - noch nicht veröffentlicht worden ist sowie, dass ich eine Veröffentlichung der Dissertation vor Abschluss der Promotion nicht ohne Genehmigung des Promotionsausschusses vornehmen werde. Die Bestimmungen dieser Ordnung sind mir bekannt. Darüber hinaus erkläre ich hiermit, dass ich die Ordnung zur Sicherung guter wissenschaftlicher Praxis und zum Umgang mit wissenschaftlichem Fehlverhalten der Universität zu Köln gelesen und sie bei der Durchführung der Dissertation zugrundeliegenden Arbeiten und der schriftlich verfassten Dissertation beachtet habe und verpflichte mich hiermit, die dort genannten Vorgaben bei allen wissenschaftlichen Tätigkeiten zu beachten und umzusetzen. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Teilpublikationen:

[1] Maurice Deserno and Katarzyna Bozek. WormSwin: Instance segmentation of C. elegans using vision transformer. *Scientific Reports*, 13(1):11021, July 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-38213-7. URL `https://doi.org/10.1038/s41598-023-38213-7`

[2] Maurice Deserno and Katarzyna Bozek. Unsupervised representation learning of c. elegans poses and behavior sequences from microscope video recordings. *bioRxiv*, 2025. doi: 10.1101/2025.02.14.638285. URL `https://www.biorxiv.org/content/early/2025/02/19/2025.02.14.638285`

Datum	Name	Unterschrift

Abstract

Recent biomedical research has led to a deep level of understanding of biological mechanisms we never reached before, enabling us to develop novel tests and cures for disease and improve our lives. However, there are still many disease and mechanisms to be researched and fully understood. One important element of this research are experiments with model organisms.

Model organisms play a crucial role in biomedical research and thanks to their wide spread use in science, we understand these organisms in a level of detail that was not reached for any other organism yet. Researchers use model organisms in their experiments to study disease like Alzheimer's and cancer, to understand aging and sleep and its underlying biomedical mechanisms. One model organism is the small roundworm Caenorhabditis elegans (C. elegans). Proposed as model organism by Sidney Brenner in the 1960s it quickly became a highly researched organism. The transparent body allows effortless observations of in vivo organs and inner processes, especially when applying stains that attach to specific biomolecules, highlighting them for improved observations. Using modern technologies, scientists can introduce all kinds of genetic mutations into an organism e.g. to understand the influence and interplay of specific genes in their experiments. Their findings do not only help to understand C. elegans but bring insights in human biology. Additionally, compared to other organisms C. elegans are easy to breed and cultivate under laboratory conditions making it a cheap and practical model organism. These and other factors result in C. elegans popularity in research and wide use in experiments.

One of the main parts of the experiments conducted with C. elegans is quantifying its behavior. As behavior is an output of the organism's neural network, it gives scientists valuable insights and helps them to understand the effects of their experiments. Together with the aforementioned benefits, behavior quantification of C. elegans enables broad possibilities for analysis and research. Unfortunately, traditional quantification of behavior is done by hand during time consuming observations of the nematodes under a microscope. Therefore, there is the need to automate this process with the promise to speed-up experiments, allowing scientists to spend more time on other tasks like interpretation of the results, obtained by the automated analysis, and conducting more experiments.

Recently, Machine Learning (ML) and Deep Learning (DL) methods specialized on C. elegans have been proposed for tasks like detection, segmentation, tracking, pose estimation and behavior quantization. At the same time, more and more high-resolution recordings of C. elegans become available, thanks to the increased level of automatization in science. Although recent methods are implementing automation in this domain with increasing success, state-of-the-art approaches struggle when it comes to more challenging poses of C. elegans like coiling and self-intersecting or complex behavior like mating and aggregation. Additionally, many state-of-the-art approaches rely on hand-engineered features, omitting one of DLs strongest abilities: to find robust, discriminating, and possibly previously unknown features, suitable for the task. Based on this we see great potential in additional research into behavior quantization using DL, to find solutions for the aforementioned challenges and we aim to tackle them with this work.

Here, we present our work, focusing on closing the gap between high-resolution behavior recording and time consuming and incomplete behavior quantification due to inaccessible poses and behavior of C. elegans. We present our novel instance segmentation approach, trained on synthetic data for segmentation of C. elegans in challenging scenes. We test our method on video data including C. elegans with coiling and heavy bending poses, as well as multiple individuals moving closely in parallel or overlapping each other. Additionally, we designed a tracking algorithm to present the abilities of our contribution. Our approach is capable of segmenting C. elegans in video frames depicting multiple individuals in challenging scenarios where previous methods failed to retrieve correct segmentation information. Our contribution allows for more detailed information required in downstream tasks and therefore enables more precise quantification and studies of the phenotype.

Next, we present our self-supervised representation learning method for behavior sequences. By now, we focused on the spatial level of behavior by segmenting individual C. elegans in image data to extract pose information on a pixel level. In this work, we include the temporal component of behavior by learning how a pose changes in time using video recordings of C. elegans. First, we train a contrastive learning network to embed pose information without relying on curve or keypoint estimations. Second, we use the pre-trained contrastive learning network to learn representations of behavior sequences. We demonstrate the abilities of our approach by visualizing the embedding space and coloring it using hand-engineered features computed by state-of-the-art methods. These visualizations reveal that our network is able to capture hand-engineered features without explicitly enforcing them during training. Thanks to the absence of explicit features, our new approach is not limited to these but is rather able to capture properties previously inaccessible. Additionally, as our method is self-supervised, it does not require pose or behavior annotations and can directly be applied on videos of C. elegans, bridging the gap between fast data acquisition and slow data labeling. Combining both approaches allows to surpass the limitations of previous state-of-the-art methods and enables quantization of challenging behavior that other methods left unsolved or only partially solved.

Acknowledgments

First, I would like to thank my supervisor Prof. Dr. Katarzyna Bozek. When I joined the Bozek Lab we were a small group: just three students and you, Kasia, our supervisor. Everything was new and all of us had to find our way, but in shortest time you were able to build up a large group with fantastic people. I still remember how surprised I was by all the trust and independence you granted me and everyone else from day one on. You helped me grow as a scientist but also in my private life and I am really thankful for the time I had in our lab.

This brings me to the next point, my colleagues and labmates. During the last years we rode this rollercoaster together. We helped each other mastering all kinds of challenges, visited conferences and collected many wonderful memories. I learned so many new things on a personal level, about many different cultures, hobbies, sports... In the lab I found new friends and I hope you will accompany for the rest of my life. Thanks to all of you!

To my friends in Cologne: Thank you for listening to my detailed and way too long explanations about deep learning and "this worm". I know you heard many funny stories and weird background information about it during the last few years. Thank you for dragging me back to reality with our game nights.

Thank you to all my friends in Aachen and anywhere else. I know that we meet less often since I started my PhD but when we do, it still feels like almost no time has passed. You never stopped checking in on me and helped me deal with stressful times.

To my brother: thank you for always listening to my stupid problems, complaints and worries. You always understood me and gave me valuable advice. Our weekly evenings, talking about life and playing all sorts of video games helped me cool down and sort my head to see what I really want.

To my caring parents: I know, I already said that after I finished my master studies and probably also after my bachelor studies... but again, I would like to honestly thank you for all the support and love you gave me. For all the advice you gave me, for encouraging me to do what I like and for listening to me when I was in need. You gave me the power to believe in myself. Thanks to you I went all that way, from Abitur to the PhD.

Finally, I would like to thank my wonderful girlfriend. You accompanied me during this wild ride, made sure I'm buckled up tight and that I stay seated. You gave me the emotional support during my downs and celebrated my ups. You lifted me up with your love and never-ending positivity and brightened every day with a smile. I know it wasn't always easy but thanks to you I made it through and I am so happy to start this new chapter, together, at your side. I love you.

Oh and thanks for reading the acknowledgments. These people and many more helped me finishing my PhD and they deserve much more than being listed on a page that many people won't read.

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

– Marie Curie

"Non est ad astra mollis e terris via."

– Lucius Annaeus Seneca

Publication Preface

This thesis and its contributions are largely based on the following manuscripts:

Published manuscripts:
[1] Maurice Deserno and Katarzyna Bozek. WormSwin: Instance segmentation of C. elegans using vision transformer. *Scientific Reports*, 13(1):11021, July 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-38213-7. URL `https://doi.org/10.1038/s41598-023-38213-7`

Pre-prints or manuscripts under review:
[2] Maurice Deserno and Katarzyna Bozek. Unsupervised representation learning of c. elegans poses and behavior sequences from microscope video recordings. *bioRxiv*, 2025. doi: 10.1101/2025.02.14.638285. URL `https://www.biorxiv.org/content/early/2025/02/19/2025.02.14.638285`

Contribution Statement

Maurice Deserno is the main author of all the publications and manuscripts included in this thesis. As the main author, he took the primary responsibility for design, implementation, data collection and analysis, as well as publishing the results in peer-reviewed venues. His and the co-authors' contributions to the included publications/manuscripts are described in the following using the Contributor Roles Taxonomy (CRediT)[1]:

[1] **Maurice Deserno:** conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, validation, visualization, writing - original draft, writing - reviewing and editing. **Katarzyna Bozek:** conceptualization, funding acquisition, project administration, resources, supervision, data and funding acquisition, writing - reviewing and editing.
[2] **Maurice Deserno:** conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, validation, visualization, writing - original draft, writing - reviewing and editing. **Katarzyna Bozek:** conceptualization, funding acquisition, project administration, resources, supervision, data and funding acquisition, writing - reviewing and editing.

[1] `https://credit.niso.org`

Contents

List of Figures

List of Abbreviations

C. elegans	Caenorhabditis elegans	v
CGC	Caenorhabditis Genetics Center	2
CRediT	Contributor Roles Taxonomy	xiii
CV	Computer Vision	2
DL	Deep Learning	vi
E. coli	Escherichia coli	1
FN	False Negative	3
FP	False Positive	3
HTC	Hybrid Task Cascade	6
ML	Machine Learning	vi
MLM	Masked Language Modeling	6
NGM	Nematode Growth Medium	1
NLP	Natural Language Processing	6
UMAP	Uniform Manifold Approximation and Projection	6
VAE	Variational autoencoder	32
ViT	Vision Transformer	6
3D	three dimensional	6

Chapter 1

Introduction

1.1 Caenorhabditis elegans as a model organism

The term model organism is not clearly defined, but generally describes a non-human organism which is widely used to study biomedical concepts and mechanisms. Ideally, they are easy to maintain under laboratory conditions and the findings gained by their studies are transferrable to other, possibly more complex, organisms like humans. The list of model organisms spans from viruses to bacteria to plants, vertebrates and invertebrates. Widely used model organisms include, but are not limited to [4]:

- Mus musculus (mouse)
- Rattus norvegicus (rat)
- Danio rerio (Zebrafish)
- Drosophila melanogaster (fruit fly)
- C. elegans (nematode)
- Arabidopsis thaliana (thale cress)

C. elegans is an important model organism in modern research that is used for all kinds of biomedical topics, ranging from studies about aging to disease and toxicology [5, 6, 7, 8]. Researching C. elegans not only brings insights into this model organism, but results in a better understanding of various types of human diseases [9, 10, 11]. Compared to other model organisms, C. elegans is a relatively simple organism. Thanks to this fact and its wide spread use in research, it is the first organism to have its entire connectome mapped [12] and the first multicellular organism whose entire genome was sequenced [13]. This can be compared to drawing the blueprint of a machine and handing it to a engineer, who tries to understand its functionality. C. elegans has two sexes, hermaphrodite and male, a high fertility and is easy to cultivate and to be kept in a laboratory environment. It is maintained on Nematode Growth Medium (NGM) petri plates with multiple other C. elegans and fed with Escherichia coli (E. coli). This allows scientists to breed organisms

in high numbers which lowers the costs of experiments, enabling to record large amounts of data for later analysis.

Contrary to other model organisms, the transparent body of C. elegans allows anatomical observations in vivo under the microscope [5, 14], granting scientists important insights, that other model organisms only offer using complicated techniques. When using stains that attach to and highlight selected biomolecules, scientists obtain even deeper insights into the organsim. Besides the strain *N2* which is found in the wild (also called wild-type), scientists have produced many other frequently used strains by introducing mutations. These strains all come with different, unique properties that can influence their behavior and help researchers understand biomedical mechanisms. To speed up experiments further and to improve comparability, institutes like Caenorhabditis Genetics Center (CGC) provide services for scientists to request and receive specific strains, previously produced by other research groups, via mail for their own experiments [14]. Additionally, compared to some other model organisms the use of C. elegans comes with no ethical constraints [5]. These factors contribute to the wide spread use of this nematode in different research domains.

1.2 Behavior as window into the organism

Behavioral phenotypes of organisms give scientists crucial insights into a model's nervous system. Quantifying behavior helps understanding the effects of genetic mutations or the organism's environment e.g. when screening drugs or toxins [7, 8, 15]. Conducting such studies with C. elegans traditionally means time consuming observations of different strains and individuals under the microscope. This labor-some process involves observing a roundworm individually or in a setup with multiple organisms for extended time periods under the microscope, while evaluating their behavior by e.g. quantifying their crawling speed or bending behavior. Recent methods have introduced more automatization and enabled broad systematic studies by publishing large amounts of behavior recordings and enabling the comparison of different strains [11, 16, 17]. This not only speeds up research, but also enables the development of novel methods bringing new insights. Early methods used classical machine learning approaches in combination with hand-engineered features [16, 18, 19]. The success of DL and Computer Vision (CV), alongside with the growing amount of available data, resulted in the development and application of new methods to data of C. elegans. In the following, we give an outline about different challenges and our contributions, developing and applying DL methods to video data of C. elegans to analyze their behavior.

1.3 Challenges

CV methods such as classification, detection or segmentation build up on low level features like colors or edges and combine them to more complex features like patterns. This is common practice in CV and works well when applied to natural color images of humans, animals or objects like cars and buildings. Objects in natural images often set off from the background because of their separation. This is due to clean edges, different colors and even distinct alignments, like horizontal or vertical lines. When working with biomedical images, these properties are often not present and even the color setting is different.

Some of the properties that make C. elegans interesting for researchers, like its transparent body or the fact of being kept in high numbers in petri dishes, result in challenging conditions for CV. When observing C. elegans with a bright-field microscope in a petri dish, the color contrast between the background and the objects of interest can be relatively low. This makes it difficult for CV approaches to detect and separate individuals. Further, crawling C. elegans can leave traces when crawling on a bacterial lawn resulting in an uneven background. Another challenge is posed by dirt or even markings on the petri dish lid resulting in visible artifacts in the image data. As invertebrates, C. elegans can take on challenging poses like tightly coiling, heavy bending or self-intersection, resulting in movement and flexibility largely differing from those of vertebrates. Additionally, they express social behavior like aggregation, male mating, or social feeding [20, 21, 22]. Social behavior can cause overlaps of two or more individuals or worms lying in parallel to each other, making it challenging to tell them apart (see Fig. 1.1). Additionally, missing keypoints like joints or entire limbs exclude many of the exiting pose estimation methods as they where developed with focus on organisms featuring these properties [23, 24, 25]. These effects can lead to False Negatives (FNs), False Positive (FP) detections, faulty segmentation and pose estimations, limiting the success of quantifying an individual's behavior using DL methods.

Figure 1.1: C. elegans in challenging setups like overlapping and crawling in close parallel contact. Fine traces are visible where individuals crawled (images taken from [3]).

1.4 Related Work

Baek et al. [26] proposed a behavior quantification method tracking and recording individual C. elegans under the microscope. Using classic CV methods like connected-components, binarization, morphological operations [27] and skeletonization, the object of interest is segmented from the background for the following feature calculation. The individual is represented by a curve describing its body pose. The method computes features of that curve, such as the worm size or the movement speed, using the recorded video data to get temporal information. Additionally, the proposed method facilitates the CART algorithm [28] to classify strains based on their locomotion. Swierczek et al. [29] proposed Multi-Worm Tracker to quantify behavior of multiple C. elegans in video data. For detection and tracking of individuals, the method applies thresholding to find foreground objects considered to be worms. These objects are then segmented using flood-fill and, if passing a specified size threshold, are considered as C. elegans. Objects are tracked through the video by defining a rectangular search area in each frame, depending on the position of the object in the previous frame. Multi-Worm Tracker calculates different features, e.g. average curvature, covered area, speed and angular speed. While this approach already adds important improvements like real-time capability and tracking of multiple C. elegans, it struggles when it comes to challenging conditions like overlapping individuals as their identities get lost in this case. Stephens et al. [30] discovered that around 95% of total variance in angles along the C. elegans approximated center-line are captured by four eigenvalues. The work introduces the term *eigenworms* to describe these "templates" of possible poses. Eigenworms are a widely used measure to quantify pose and behavior of tracked worms. Javer et al. [18, 19] proposed Tierpsy, a multi-worm tracking software. Like previous methods, C. elegans are segmented from the background by thresholding. The foreground object is skeletonized to estimate the center-line. Based on this curve, Tierpsy calculates hand-engineered features, describing pose and motion of C. elegans. While this method gives scientist an easy-to-use software, it is not able to quantify challenging poses like coiling or track worms with overlapping behavior. While previous methods focused on tracking multiple organisms in a single petri dish, Barlow et al. [31] introduced a setup consisting of an array of six cameras, able to record C. elegans on a 96-well plate with high resolution, allowing behavioral studies. This approach is able to record video data in large scale and therefore speeds up phenotypic studies.

Previous methods applied threshold-based approaches to segment C. elegans from the background. While this is a relative simple but effective way to tackle the task of segmentation, it comes with many downsides. As illumination or background can change in recordings, the threshold needs to either be adapted by hand or using an algorithm. Additionally, this approach is sensitive to imperfect set threshold values, as already small changes can result in over- or under-segmentation, meaning that either too much is considered as foreground or the object of interest is cut off and not fully extracted from the background. While thresholding approaches like Otsu's method [32] can automatically determine a suiting threshold value, some general downsides of thresholding algorithms

still persist. Artifacts or other objects in the image, like dirt or marks on the petri dish lid can not be distinguished from C. elegans using thresholding if they have similar pixel values. Finally, thresholding alone is not able to separate overlapping or touching individuals of same appearance as it only allows binary segmentation. Therefore, scientist searched for alternative approaches to accurately segment individuals. The rise of CV and widely available video data of C. elegans sparked the development of new methods resulting in more accurate segmentation and tracking of challenging poses and behavior. Compared to classic ML algorithms, CV often takes more context into consideration, resulting in a higher accuracy.

Banerjee et al. [33] presented a DL-based detection and tracking approach. The method first detects C. elegans using YOLOv5 [34] before passing the detected areas to a thresholding algorithm for segmentation and StrongSORT [35] for tracking. Hebert et al. [36] published a ResNet-based [37] approach that allows estimating coiling poses of individual C. elegans. Using video data with center line annotations for frames prior to challenging coiling poses, the method is able to estimate pose information for these challenging behavior. For training, the authors facilitate synthetically generated images of C. elegans with different bending behavior. To resolve different center-line predictions by the network, caused by a swapped head/tail estimation, the approach is trained to predict two possible center-lines. Using the same synthetic image generation approach that also generated the training dataset, images of artificial C. elegans are generated for each predicted center-line. During evaluation, these images are then compared to the true input image to find the best predicted center-line. Alonso et al. [38] presented a pose estimation and tracking method for slender and overlapping bodies like C. elegans. The method is tested on swimming C. elegans in dense scenes. For training, the authors generate data using a physics-based model to avoid labeling data by hand. Their method estimates a center-line for each individual and tracks it in a frame using information of the adjacent frames. As this approach focuses on swimming worms, Weheliye et al. [39] build up on it and proposed DeepTangleCrawl for pose estimation and tracking of the more complex behavior of crawling C. elegans. They adapt the approach by Alonso et al. [38] to more challenging coiling and self-intersecting poses and train the method with new video data, containing annotations computed by Tierpsy [18, 19] and hand-made annotations for challenging poses. While this method accurately tracks swimming and crawling C. elegans and enables their phenotypic screening, the method still struggles when it comes to individuals persistently coiling tightly or worms in persistent tight parallel contact.

Based on the success of previous DL methods and motivated by the still existing challenges posed by coiling and overlapping C. elegans in dense scenes, we focused on researching new approaches to segment individual worms and extract behavior information. In the following section, we will highlight our contributions presented in this work.

1.5 Contributions

1.5.1 Instance segmentation of C. elegans

To quantify the behavior of an individual we need to extract spatial and temporal information from video data. The spatial component hereby is mainly the worm's body pose. Here [1], we decided to extract the pose of C. elegans on a pixel level by applying an instance segmentation approach, rather than estimating a curve or keypoints. We apply a combination of the Swin Transformer [40] together with the Hybrid Task Cascade (HTC) network architecture [41] and train it with a synthetic dataset. Generating our own synthetic dataset allowed us to create challenging conditions by artificially overlapping individual C. elegans. Further, this enabled us to reduce the amount of data that needs to be labeled by hand. The hierarchical feature maps of Swin Transformer use small-sized patches which allow to focus on smaller features and result in a more detailed predictions compared to other Vision Transformers (ViTs) [40]. We evaluate our approach on different real (non-synthetic) datasets and compare it to other methods. Source code[1], model weights [42], datasets and annotations generated [3] in this work are available online.

1.5.2 Representation learning of C. elegans behavior

We present a self-supervised representation learning approach, combining contrastive learning and a Transformer-encoder architecture [43] to learn pose and behavior sequence embeddings. Since our approach is self-supervised, it does not require any hand-labeled ground-truth data. First, we apply VICReg [44] combined with ResNet-18 [37] to learn pose embeddings from video frames. We construct sequences of twelve pose embeddings of consecutive frames and mask the last five elements, similar to Masked Language Modeling (MLM) [45]. Using a Transformer encoder architecture similar to those used in Natural Language Processing (NLP) [45], we predict the masked elements to learn behavior sequence embeddings. To evaluate our approach, we reduce the pose and the behavior embedding space to three dimensional (3D) using Uniform Manifold Approximation and Projection (UMAP) [46] and color it using features calculated by Tierpsy [18, 19].

In the upcoming chapters we highlight our individual contributions in more detail, starting with describing our instance segmentation approach in chapter 2 followed by chapter 3 describing the representation learning approach. We conclude this work in chapter 4 with a summary and future work. Source code is available online[2].

[1] `https://github.com/bozeklab/worm-swin`

[2] `https://github.com/bozeklab/worm-behavior`

Chapter 2

Instance segmentation of C. elegans

The model organism *C. elegans* is often kept and observed in a petri dish with multiple organisms. Separating individuals accurately in recorded image data is crucial to enable uninterrupted (automated) analysis like phenotype studies. The worm's ability to bend, coil and overlap with other individuals makes this a challenging task. In the following publication [1], we present a vision transformer based approach to apply instance segmentation on bright-field microscopy frames from video data. We demonstrate the performance of our method by applying it on images depicting challenging behavior like overlapping *C. elegans*. Additionally we test the abilities of our approach when combined with a tracking algorithm.

www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

WormSwin: Instance segmentation of *C. elegans* using vision transformer

Maurice Deserno[1,2,4]✉ & Katarzyna Bozek[1,2,3]

The possibility to extract motion of a single organism from video recordings at a large-scale provides means for the quantitative study of its behavior, both individual and collective. This task is particularly difficult for organisms that interact with one another, overlap, and occlude parts of their bodies in the recording. Here we propose WormSwin—an approach to extract single animal postures of *Caenorhabditis elegans* (*C. elegans*) from recordings of many organisms in a single microscope well. Based on transformer neural network architecture our method segments individual worms across a range of videos and images generated in different labs. Our solutions offers accuracy of 0.990 average precision ($AP_{0.50}$) and comparable results on the benchmark image dataset BBBC010. Finally, it allows to segment challenging overlapping postures of mating worms with an accuracy sufficient to track the organisms with a simple tracking heuristic. An accurate and efficient method for *C. elegans* segmentation opens up new opportunities for studying of its behaviors previously inaccessible due to the difficulty in the worm extraction from the video frames.

Behaviour is the external output of an animal's nervous system. The possibility to systematically observe, extract, and quantify an animal's motion is a prerequisite to investigate and ultimately understand its behavioral repertoire. Alterations to an organism's natural behavior is a phenotypic readout of the neural and other molecular changes that are causing them. To fully understand the functioning of neural mechanisms it is therefore essential to dissect their effect on an animal's behavior.

Capturing behavior requires video acquisition systems allowing to either view or infer an entire posture of an organism and its change in time. One of the main challenges in obtaining complete and precise posture measurements are the occlusions of animal body parts in a 2D video recording, especially if more than one individual is being imaged. To resolve this, extensive 3D motion capture systems have been developed[1] as well as methods that allow to impute the occluded parts of the posture[2].

These challenges have not yet been resolved for the model organism *C. elegans*. While imaging the nematode's behavior is less complex than imaging of larger organisms and massively parallel recording systems allow to capture thousands of worms at a time[3,4], there are currently no end-to-end methods that resolve their postures when occlusions occur. The quantification of *C. elegans* strains' behavior and characterization of their phenotypes is therefore based on segments of worm motion in which it does not coil or intersect with another worm. As a result, a large portion of the worm behavior, including its group behavior, cannot be quantitatively analyzed.

Here we propose an automated method for *C. elegans* posture extraction from 2D video recordings. Based on deep learning transformer architecture and a classical instance segmentation training objective, our solution allows to correctly infer an outline of an individual worm body in overlapping and occluded configurations. We train the neural network on randomly generated image data, obtaining a solution that generalizes to various real datasets. With the segmentation outputs of our method we are able to correctly infer worm trajectories with a simple position matching heuristics. WormSwin opens up new opportunities to study the full repertoire of *C. elegans* behavior including behaviors such as mating that were previously inaccessible to quantitative analysis.

[1]Center for Molecular Medicine Cologne (CMMC), Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, North Rhine-Westphalia, Germany. [2]Institute for Biomedical Informatics, Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, North Rhine-Westphalia, Germany. [3]Cologne Excellence Cluster on Cellular Stress Responses in Aging-Associated Diseases (CECAD), University of Cologne, Cologne, North Rhine-Westphalia, Germany. [4]Faculty of Mathematics and Natural Sciences, University of Cologne, Cologne, North Rhine-Westphalia, Germany. ✉email: maurice.deserno@uni-koeln.de

Related work

Over the past years different methods for *C. elegans* detection and segmentation have been proposed, either as part of a general approach to tracking and behavioral studies, or as a stand-alone method.

One of the first methods for automated worm tracking and behavior quantification was proposed by Baek et al.[5]. The method used a computer-controlled tracker for single worms, recording grayscale videos. The grayscale frames of a video were binarized based on the mean and standard deviation of pixel intensities and a predefined threshold. The method computes features such as the area of foreground or the movement between two frames in the binarized videos and uses them as input to the algorithm[6] for classification of different *C. elegans* strains. Swierczek et al.[7] proposed a tracking approach called Multi Worm Tracker. The method calculates a background estimate using pixel intensity values. Moving objects are found by searching for pixels darker than the background by a specific threshold. In the next frame, the objects are searched for in the vicinity of their previous location.

The arrival of deep learning offered new opportunities to build more accurate methods for worm segmentation and tracking. Javer et al.[8] developed a multi-object tracking framework able to track *C. elegans* as well as fish and drosophila larvae. The method requires manual tuning of segmentation parameters to best perform with the given recorded data and comes with a graphical user interface for the ease of use and evaluation of the results. Using the motion data, the framework extracts a large number of features characterizing worm movement. Hebert et al.[9] proposed a pose estimation method for videos of single moving *C. elegans* in challenging poses like coiling. Using a ResNetV2[10]-like architecture the centerline of worms is predicted. With the help of temporal information the head and tail position is determined. Wählby et al.[11] proposed a phenotype analysis toolbox based on the open-source CellProfiler[12] project. To untangle clusters of worms the authors describe them as a mathematical graph and, using a learned model of worm postures, search for the best representation of true worms. The worm posture model is based on a training dataset of isolated single *C. elegans* shapes and on computed angle-based shape descriptors. One of the downsides of this approach is that unexpected phenotypes are likely to be discarded as debris. Banerjee et al.[13] introduced a deep learning *C. elegans* tracking method in which the detection is based on YOLOv5[14] and tracking on Strong SORT algorithm[15]. For each detected object the method outputs its bounding box, then threshold-based segmentation and skeletonization are applied to infer shapes of the detected objects. Fudickar et al.[16] developed a two-shot segmentation method based on Mask R-CNN[17] with ResNet-101[18] backbone, to segment *C. elegans* imaged in petri-dishes with a low-cost image capturing system. However, the method did not solve the problem of segmenting overlapping worms and segments them as one object. Mais et al.[19] developed a proposal-free instance segmentation method, called PatchPerPix, based on a convolutional neural network (CNN) trained to predict the shape of a *C. elegans* in a small patch of the whole image (local shape patches). The method uses a modified U-Net[20] deep neural network and patch affinity graphs to reconstruct individual worm shapes. For each pixel the method predicts which shape patch it belongs to and, using a patch affinity graph, merges the patches to form complete instance shapes. Lalit et al.[21] proposed an embedding-based instance segmentation method for 2D and 3D microscopy data, called EmbedSeg. The method is based on ERF-Net[22], predicting spatial embeddings of pixels. These embeddings are then clustered into object instances. To train this method, an additional step of pre-processing the dataset is required to generate object-centered image patches for every object. The method was tested on different datasets including the *C. elegans* BBBC010 dataset.

Among the methods described above there are one- and two-shot detectors. One shot-detector architectures like YOLO[23] detect objects in one step. Pre-defined boxes (also called anchors) are placed onto a grid, laid over the image. For each box, the network predicts if the box contains an object. On the other hand, two-shot detectors consist of a region proposal network (RPN) proposing regions of interest (RoI) to a second network, refining these proposals to form the actual predictions. One-shot object detection methods (like[13]) are in general less computationally expensive compared to two-shot approaches (e.g.[16]), although the latter ones achieve a higher precision especially in more challenging scenes. This is one of the reasons for the high popularity of two-shot networks such as Mask R-CNN in the domain of instance segmentation.

Usually more than one box is predicted per object. To only keep the best matching box, many methods apply non-maximum suppression (NMS). This approach consists of removing from the predicted highly overlapping bounding boxes those with lower probability values as potential false positive detections of the same object. However, NMS can lead to removal of correct detections, especially in dense scenes, where many objects in the image overlap.

In this paper we address the problem of segmenting objects in dense scenes by combining the well established architecture of two-shot detectors with state of the art vision transformer. To avoid the pitfalls of the NMS algorithm, we apply Soft Non Maximum Suppression (Soft-NMS)[24].

Methods

Datasets. CSB-1 dataset. The CSB-1 dataset consists of 56 videos with a length of $\sim$ 1.5 min, a frame rate of 5 Hz and frame size of 912 $\times$ 736 px which were generated to describe the new *C. elegans* csb-1 strain[25]. We annotated 10 of those videos, where nine videos were reserved for training and one for testing. The videos do not contain any visible petri-dish edges, have different backgrounds and varying numbers of worms. We extracted frames from the videos using *FFmpeg* (https://ffmpeg.org) and used them to generate our synthetic training dataset described below.

Worms were annotated individually with a binary mask labelling foreground pixels, resulting in one mask image per worm per frame. These separate masks allow to mark all the worms also in cases where pixels of individual *C. elegans* overlap. The labeled CSB-1 dataset contains more than 60,500 individual worm masks. *C.*

elegans at the image borders are ignored during the labelling process. Our data is available under https://doi.org/10.5281/zenodo.7456803 as a rich resource to develop better methods for animal tracking.

Synthetic dataset. For training the model we generated a synthetic dataset using the nine annotated videos from the CSB-1 dataset described above. We automatically cut out foreground objects from the original grayscale images, according to their polygon annotations and created patches with a worm in the foreground and transparent background. Additionally, we created background images as templates by removing all foreground objects using standard graphics software and filled them with patches of background, taken from the same background images.

The following pipeline was applied to create each image of the synthetic dataset:

1. Randomly select 5–30 foreground objects and a background template
2. Randomly flip and rotate foreground objects and their corresponding annotations
3. Apply blurring to foreground objects by averaging the pixel values using a 2 × 2 px kernel
4. Place foreground object patches on background image:
   (a) In 20% cases: place a foreground object on top of another one
   (b) In 80% cases: place a foreground object randomly on the background image

The generated training dataset consists of 10,000 grayscale images with a size of 912 × 736 px and more than 175,000 labeled *C. elegans* and additional 1000 images for testing (see Fig. 1a). We randomly added grayscale rings of random sizes surrounding the center of the images (see Fig. 1b) to make the network robust against similar artifacts (e.g. petri-dish edges) in other test datasets. Foreground objects might overlap with the artificial petri-dish edges, but are only placed on the inside of the rings. Using the object masks of the original data, for each foreground object we generated a binary mask corresponding to its artificially generated location and shape. These masks are used as ground truth for model training and testing on this dataset. Our synthetic training dataset is available at https://doi.org/10.5281/zenodo.7456803.

BBBC010. The "BBBC010—*C. elegans* live/dead assay"[26] (BBBC010) dataset consists of 200 images, divided into 100 bright-field and 100 green fluorescent protein (GFP) microscopy images of the same scene. The images have a size of 696 × 520 px and are saved as 16-bit grayscale TIFF files. For our experiments we converted the images to 8-bit grayscale PNG images. The images contain a black border surrounding the region of interest (ROI) with the *C. elegans* in the center (Fig. 1c) which makes up around 50% of the image. Ground truth consists of binary foreground/background images for each worm separately, allowing to disentangle the overlapping shapes.

The images show *C. elegans* exposed to *Enterococcus faecalis* with a negative control group containing dead worms and a positive control group, which was treated with ampicillin and includes alive worms. While the alive *C. elegans* have the natural curved shapes (Fig. 1c), the negative control group appear rod-like with an uneven texture (Fig. 1d).

Mating dataset. The mating dataset (MD) was created from a 10 min. long video with a frame rate of 25 Hz and a frame size of 3036 × 3036 px. It contains freely moving worms as well as mating ones. Mating behavior is particularly difficult to segment as the two individuals are strongly overlapping and parallel to one another (Fig. 1e). This dataset represents therefore the most challenging segmentation task for our method.

We downsampled the video to 5 Hz and selected 50 frames randomly for annotation and testing of our approach. More than 3900 individual worm postures were labeled in this dataset. The labeling includes only mature *C. elegans*, worms touching the image boundary were ignored. We split the frames into 450 images with a size of 1012 × 1012 px without overlap. The grayscale images show *C. elegans* in a petri-dish with the edges visible (see example patch in Fig. 1f).

Network architecture

To predict bounding boxes and instance segmentation masks we use the Hybrid Task Cascade (HTC)[27] neural network architecture, combined with Swin Transformer[28] as backbone (similar to[28]).

Swin Transformer is a Vision Transformer (ViT)-based backbone architecture[29], which can be applied to different vision-related tasks (e.g. classification, detection or segmentation). Previous ViTs divided the input image into relatively large patches and computed self attention among them. ViTs showed lower computational complexity, but did not account for small details in large images. To tackle this problem Swin Transformer introduced a **S**hifted **Win**dow approach to reduce the computational complexity of standard multi-head self attention (MSA) modules. Additionally, Swin Transformer builds hierarchical feature maps, merging image patches in deeper layers, enabling small-sized patches, leading to more detailed predictions. We chose Swin-L architecture variant in our study which was pre-trained on ImageNet-21K[30] with an image size of 384 × 384 px (similar to[28]).

HTC improves the architecture of Cascade Mask R-CNN[31] by introducing interleaved bounding box regression and instance segmentation mask prediction. The information flow is optimized by adding direct connections between the individual mask branches (Fig. 2). Additionally, a semantic segmentation branch is added to the original architecture to help to distinguish between foreground and background. In our experiments we do not use this additional semantic segmentation branch.

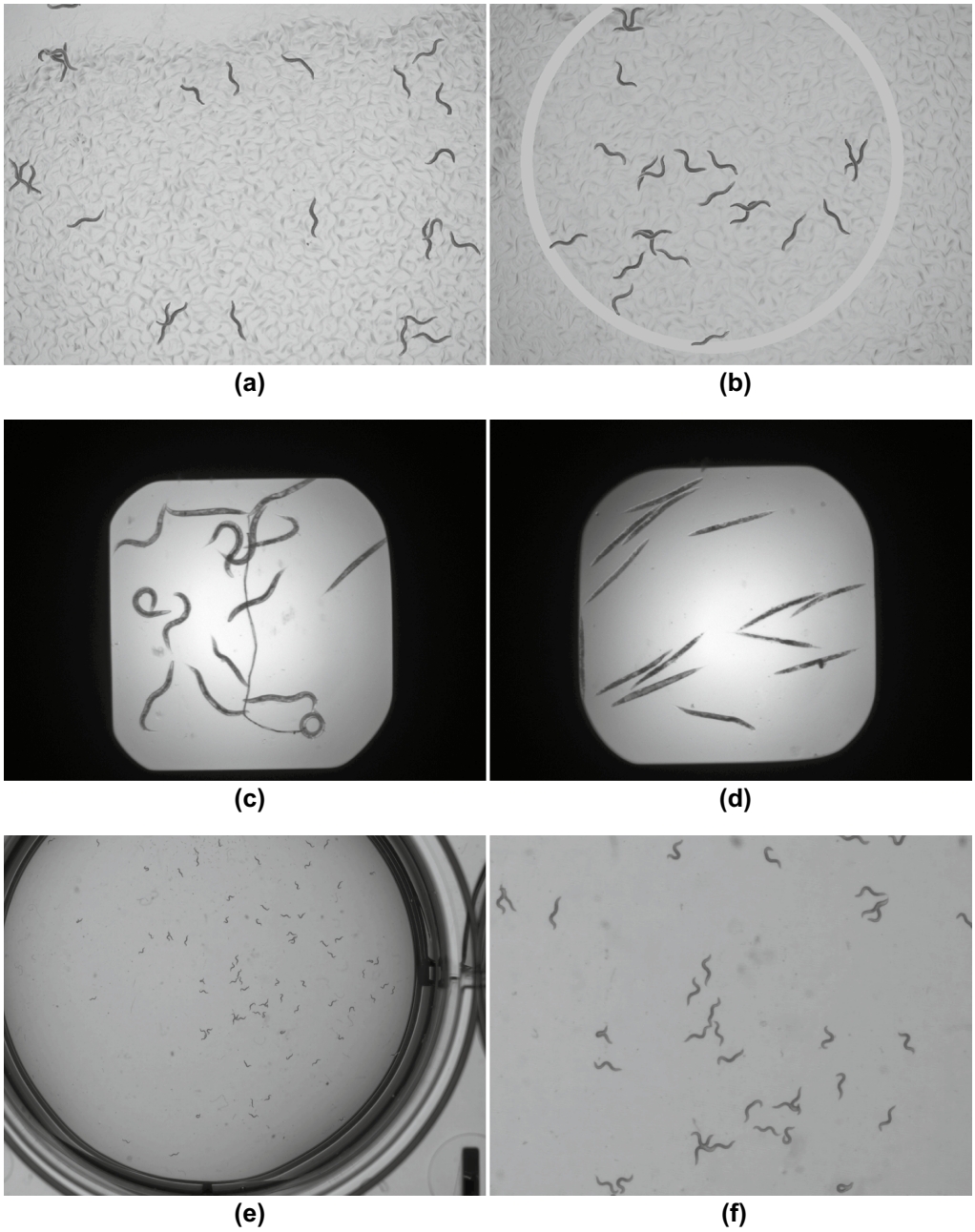


Figure 1. Example images from the datasets used in this study: (**a**) synthetic dataset example with added ring, (**b**) synthetic dataset without ring, (**c**) BBBC010 dataset example with mostly alive *C. elegans*, (**d**) BBBC010 dataset patch with mostly dead *C. elegans*, (**e**) mating dataset with petri-dish ring, (**f**) zoomed-in mating dataset patch with many overlaps.

To further improve the accuracy when training on small batches, we exchanged the default Batch Normalization (BN)[32] with Group Normalization (GN)[33] and Weight Standardization (WS)[34] in HTC (similar to[34]). We also replaced the Shared 2 Fully-Connected Bounding Box heads (Shared2FC) by Shared 4 Convolution + one

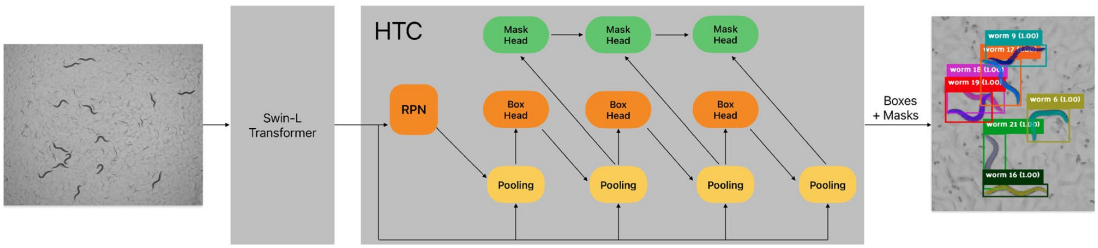


Figure 2. Network architecture based on Swin-L backbone and HTC. Batch norm (BN) layers in HTC are replaced by group norm (GN) + weight standardization (WS). Bounding box heads are changed from the original Shared2FC architecture to Shared4Conv1FC.

Fully-Connected Bounding Box head (Shared4Conv1FC) (as described in[33]). To suppress low quality detections but keep high quality predictions in dense and overlapping scenes we use Soft-NMS instead of the traditional NMS algorithm for the R-CNN during test time (see HTC++[28]).

Training. We used multi-scale training with a size between 480 and 800 px for the shorter side and 1333 px at most for the longer side, AdamW[35] as optimizer, Cosine Annealing Learning Rate Scheduler[36] and Linear Warm-Up[37] (similar to[28]). The learning rate was set to $2.5e^{-5}$ and weight decay to 0.1. The number of warm-up iterations of the linear warm-up and learning rate scheduler was set to 1000, warm-up ratio to 0.1 and minimum learning rate ratio to $1e^{-5}$. During training and testing the NMS threshold for the RPN was set to 0.7, the Soft-NMS of the R-CNN was set to 0.5 during test time. We used random flipping with a probability of 0.5 and AutoAugment[38] for multiscale resizing and cropping. Additionally, we used the pre-trained weights for the Swin backbone, trained on ImageNet-21K with an image size of 384 × 384 px (similar to[28]). We tested our approach on three different datasets: the publicly available BBBC010 dataset, MD and CSB-1 datasets. During testing all images were resized to 800 px on the short side and to no more than 1333 px on the longer side, preserving the original ratio. We excluded all instances touching image borders as incomplete *C. elegans* instances.

In all our experiments we used the MMDetection framework[39]. Our code and network configuration file for the MMDetection framework are available at https://github.com/bozeklab/worm-swin.

WormSwin was trained using 4 Nvidia Tesla V100-SMX2 32 GB GPUs, 6 cores of an Intel Xeon Gold 6248 CPU @ 2.50 GHz and 100 GB of RAM. With a batch size of four (one image per GPU) and two workers per GPU, training for 36 epochs took ∼ 19 h. Evaluation on the test set runs at a speed of 2.7 images/s.

Results

We trained WormSwin on data synthetically generated based on the CSB-1 dataset. The procedure of data generation allows us to control the degree of overlap among individuals and to train the network on a large number of images containing overlapping worms to improve segmentation of dense scenes. Once trained, we evaluated the model on a synthetic test set (see Table 1) as well as on three independent datasets: BBBC010, MD and CSB-1. These datasets come from different labs, show visual variability, and contain different number and degree of overlapping *C. elegans*. We report our results mostly as COCO Average Precision (AP)[40] calculated using pycocotools (https://github.com/cocodataset/cocoapi). For the BBBC010 dataset, we report our results as DSB AP for comparison to other methods. AP is the area under the precision-recall curve and its values are between 0 and 1, with a higher AP representing better performance. Precision and recall of the detection is calculated for instances that show intersection over union (IoU) with the ground truth above a predefined threshold. DSB mAP calculates a mean Jaccard Index by using different IoU thresholds. COCO mAP uses a more complex approach: detections are sorted by descending confidence score. The calculation iterates over all detections in this order, marks them as True Positive (TP) or False Positive (FP) and adds them to calculate the precision until a recall of 1.0 is reached or iterated over all detections. Different IoU thresholds are used to label detections as TP or FP.

We report our results mostly for two IoU thresholds: 0.5 and 0.75 as well as a mean AP (mAP) for thresholds from 0.5 to 0.95 with a step size of 0.05. One of the most challenging parts for instance segmentation of *C. elegans*, as well as other biological systems, are overlapping objects in dense configurations. To measure the accuracy of our approach explicitly for overlapping objects, we added a dedicated AP metric. We defined overlapping objects as those whose ground truth bounding boxes overlap by more than 25% or whose segmentation masks that any overlap (>0% IoU). We report the AP for all objects as well as for the overlapping objects separately (Table 2).

CSB-1 dataset. Although trained on synthetically generated data, our method generalizes fairly well to the real video data with a mAP of 0.819 and 0.585 for the bounding box and mask respectively, lower by only ∼ 0.09 mAP compared to the synthetic data. The same metric on the overlapping worms in the CSB-1 dataset are 0.551 and 0.527. While the mAP is lower for the overlapping *C. elegans* compared to the results on the entire dataset, the $AP_{0.50}$ of the bounding box and mask on the overlapping worms are 0.883 and 0.975, respectively. This result suggests that the worms are detected correctly in principle but there exist errors in mask prediction. What these mask prediction errors are, is however not clear at a first glance. Despite the difference between the $AP_{0.50}$ and $AP_{0.75}$ in the overlapping worms, we found that the segmentation masks align in general well with the ground truth (Fig. 3), however pixels on the edges of each object tend to be imprecisely segmented. Due to

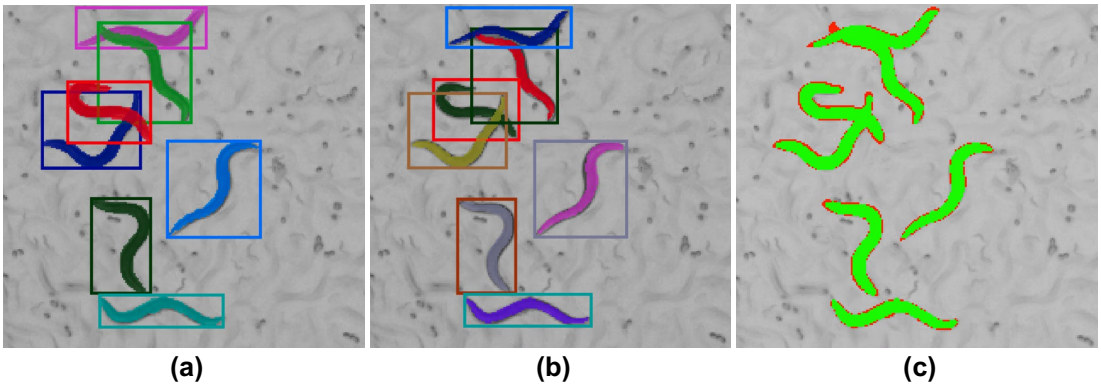


Figure 3. Example from the CSB-1 dataset (box and mask colors are selected randomly). (**a**) Ground truth annotations, (**b**) predicted bounding boxes and masks, (**c**) TP (green), FP and FN (red) pixels.

	$AP_{0.50}$	$AP_{0.75}$	$mAP_{0.50:0.95}$
CSB-1			
WormSwin (box)	0.990	0.976	0.819
WormSwin (mask)	0.990	0.675	0.585
Synthetic			
WormSwin (box)	0.989	0.978	0.909
WormSwin (mask)	0.977	0.918	0.679
BBBC010			
PatchPerPix ppp+dec (mask)	0.939	0.891	0.775
WormSwin (box)†	0.985	0.949	0.823
WormSwin (mask)†	0.954	0.801	0.622
WormSwin (mask)*, †	0.964	0.815	0.629
MD			
WormSwin (box)†	0.990	0.968	0.832
WormSwin (mask)†	0.980	0.551	0.542

Table 1. Test results on all instances. "Box" and "mask" refer to the accuracy of detection of the bounding box and segmentation mask, respectively. PatchPerPix ppp+dec refers to the network variant, introduced by[19]. (*Multi-scale testing, †additional training data).

	$AP_{0.50}$	$AP_{0.75}$	$mAP_{0.50:0.95}$
CSB-1			
WormSwin (box)	0.883	0.643	0.551
WormSwin (mask)	0.975	0.409	0.527
Synthetic CSB-1			
WormSwin (box)	0.983	0.958	0.853
WormSwin (mask)	0.959	0.821	0.613
BBBC010			
WormSwin (box) †	0.911	0.821	0.661
WormSwin (mask) †	0.873	0.565	0.488
WormSwin (mask)*, †	0.895	0.573	0.501
MD			
WormSwin (box)†	0.822	0.633	0.505
WormSwin (mask)†	0.893	0.079	0.355

Table 2. Test results for overlapping worms only (* multi-scale testing, †additional training data).

the small size of a worm mask with ∼ 500 px, errors at the edges of the predicted masks represent ∼ 30% of all foreground pixels.

To test the hypothesis that most error occur on the mask edges, we implemented an alternative version of the IoU: if a pixel in either ground-truth or predicted mask is at the border of an object (when the 4-way neighborhood is not fully foreground) then it is set to the value of the pixel at this position in the other mask. This way, object border pixels which otherwise would be considered as false negative (FN) or false positive (FP) do not influence the IoU calculation in a negative way. Using this calculation, the mean IoU on the test subset raised from 0.827 to 0.961 (+13.4% increase) on the CSB-1 dataset.

BBBC010 dataset. Because of the very limited number of training samples (50 images) the predictions of the network trained on BBBC010 were of poor quality. Therefore, we used the network pre-trained on our synthetic data and fine-tune it on 50 randomly selected images from the BBBC010 dataset. We compared the performance of our approach to two existing methods: PatchPerPix[19] and EmbedSeg[21]. To enable this comparison, instead of the COCO AP metric (see Table 1) we used (Data Science Bowl) DSB AP (https://www.kaggle.com/competitions/data-science-bowl-2018/overview/evaluation) as accuracy evaluation on this dataset which was used in the original EmbedSeg method publication[21] (see Table 3).

We used the alternative IoU calculation already used for the CSB-1 dataset, to calculate the DSB accuracy without considering object edges. With the IoU defined this way, using the DSB metric we achieve 0.769 mAP (+0.233), 0.929 $AP_{0.50}$ (+0.012) and 0.823 $AP_{0.80}$ (+0.487) (compare to Table 3).

Mating dataset. Finally, we tested WormSwin on the MD dataset using weights pre-trained on our synthetic dataset. In this dataset we annotated 50 images, which are larger in size and contain a higher number of *C. elegans* compared to the BBBC010 dataset. Further, we split them into patches of size 1024 × 1024 px. We report our results in Table 1). Despite the challenging configurations of worms in this dataset, our method correctly identifies the segmented objects, as indicated by the $AP_{0.50}$ which is comparable to the $AP_{0.50}$ in other datasets. However the $AP_{0.75}$ and $mAP_{0.50:0.95}$ suggest that, while correctly detected, the segmentation masks of the detected objects are imprecise. Similar to other datasets, we hypothesise that these errors occur on the boundaries of the segmentation masks (Fig. 4) as well as are due to the very challenging object overlaps in this dataset.

Tracking. To test if our segmentation results are sufficiently accurate to allow for worm tracking and further behavioral analysis, we implemented a simple IoU-based matching method (Fig. 5) and applied it on our predicted instance segmentation masks in the CBS-1 test set. Between two consecutive frames, objects with the highest overlap in mask are matched into a trajectory. Iterating the matching procedure over all video frames results in object trajectories. In this simple approach, if an object is not detected in a frame but detected in a subsequent frame its trajectory is disrupted and two separate trajectories are created instead. We attempt to reconnect such trajectories in a post-processing step: for 10 frames after loosing an object, starting points of new trajectories are compared with the endpoint of the lost trajectory. If the segmentation masks at these points overlap with at least 50%, the trajectories are reconnected. In the frames with missing segmentation masks the positions of *C. elegans* can be interpolated between two ends of reconnected trajectories.

While a tracking method is outside of the scope of this study, our simple approach allows to build trajectories of interacting mating worms (Fig. 5). Tracking these challenging *C. elegans* interactions opens up new possibilities of studying its behavior.

Discussion

In this work we present WormSwin, a deep learning approach for instance segmentation of microscopy images of *C. elegans*. Our method combines several recent improvements in deep learning and instance segmentation (Transformer Networks, HTC, Group Normalization, Weight Standardization, Soft-NMS) into a single approach trained end-to-end. WormSwin does not require any pre-processing of the image data, enabling researchers to directly apply it on their video or image data.

Together with our method we provide a large dataset of *C. elegans* images with instance mask annotations to help researchers develop better segmentation approaches in the future. The new dataset is by an order of magnitude larger compared to the BBBC010 dataset, enabling training deeper network architectures.

The small size of the BBBC010 benchmark dataset is a limiting factor to extensively train and test our method on this dataset. The accuracy of our method is lower on this dataset compared to the CSB-1 which might be

	$AP_{0.50}$	$AP_{0.60}$	$AP_{0.70}$	$AP_{0.80}$	$AP_{0.90}$	mAP
BBBC010						
PatchPerPix ppp+dec[19]	0.930	0.905	0.879	0.792	0.386	0.727
EmbedSeg[21]	0.965	0.934	0.896	0.762	0.326	–
WormSwin (mask)*, †	0.917	0.884	0.785	0.336	0.005	0.536
WormSwin (mask)*, †, ‡	0.929	0.920	0.890	0.823	0.483	0.769

Table 3. Test results using DSB metric (* multi-scale testing, †additional training data, ‡alternative IoU without object edges, mAP for IoUs in range 0.5–0.95, step size 0.05).

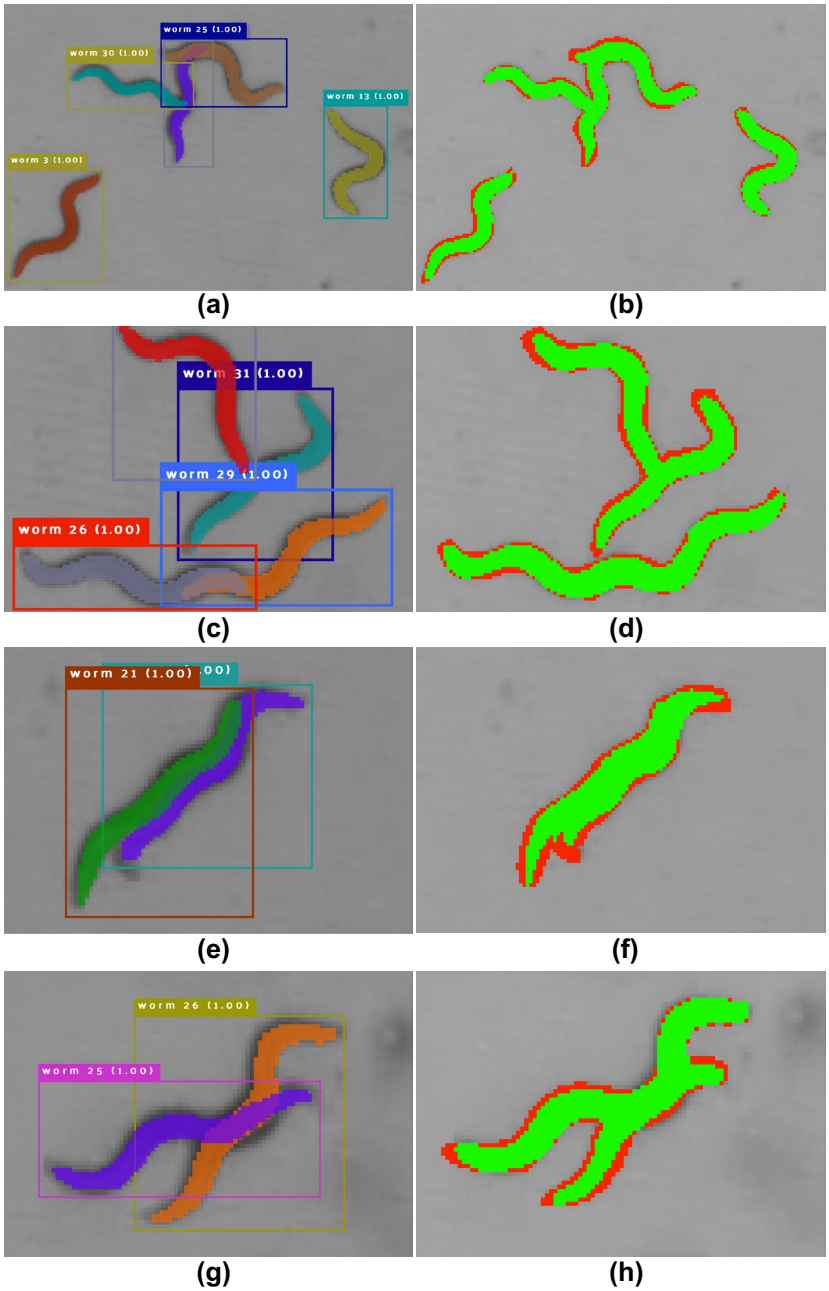
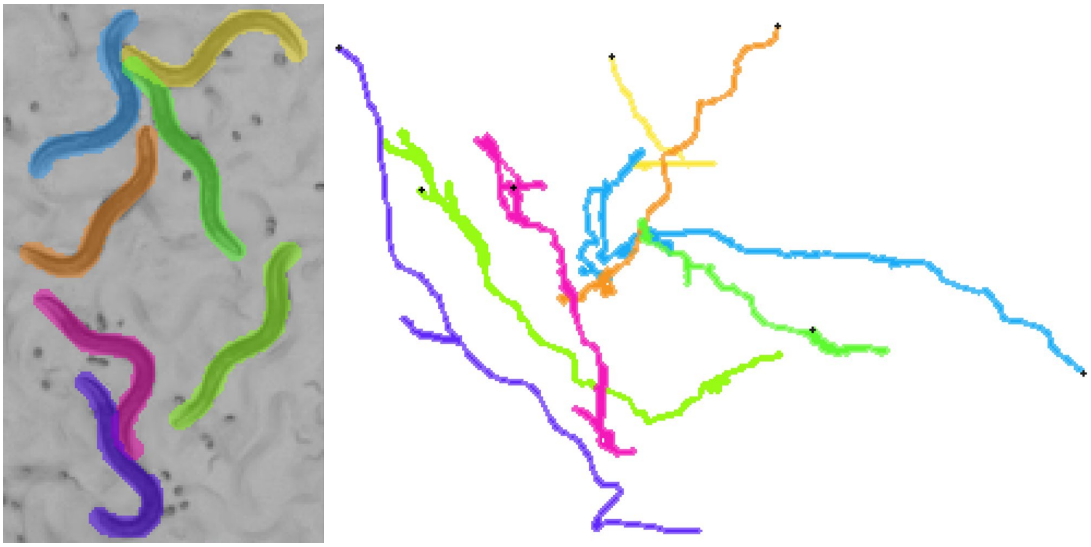


Figure 4. Results on the Mating Dataset (box and mask colors are selected randomly). (**a,c,e,g**) Segmentation results, (**b,d,f,h**) TP (green), FP and FN (red) pixels.

attributed to the differences in the color intensities, size and appearance of *C. elegans* between the two datasets. Since retraining of WormSwin on a small amount of BBBC010 images improved the methods performance, we suggest that to accurately segment datasets differing from CSB-1 characteristics, a similar retraining is necessary.

Notably, our method shows a decrease in AP in the higher IoU threshold categories (e.g. Table 3$AP_{0.80}$). Despite this precision drop, the segmentation masks appear overall correct (Figs. 3, 4). We therefore hypothesize that the major errors in the segmentation masks occur on the boundaries of the foreground area and further



(a) Tracked *C. elegans*. **(b)** Selected trajectories of interacting *C. elegans*.

Figure 5. Example of tracked *C. elegans*.

substantiate this by calculating accuracy metric that does not take into account boundary pixels. The reason for this type of error might be e.g. variation in human-generated labeling. We introduce blurring in the synthetic training data which might additionally change the appearance of the object contours. Despite these errors, individual *C. elegans* poses are captured by the predicted segmentation masks and can be subject to further quantitative analysis.

As a major future improvement of this work we see models exploring temporal information to improve segmentation of overlapping objects. Information on how *C. elegans* individuals arrive in a specific configuration is of great help in disentangling their postures. Previous work by Fontaine et al.[41] model *C. elegans* using planar curves and Central Difference Kalman Filter (CDKF) to track multiple worms. This approach shows good results even when occlusion occurs. Similarly, Alonso et al.[42] proposed a deep learning approach for detection and tracking in high density microscopy data, based on splines as shape descriptors. They test their approach on different dataset including videos of *C. elegans* and achieve high accuracy in dense scenes with a high degree of occlusion. Such methods are a step towards combining segmentation with tracking in a single training objective. While generating training datasets for multi-object tracking is a massive work burden, the accuracy of our segmentation approach allows to build preliminary trajectories in an automated fashion.

Data availability

The datasets (except for the BBBC010 dataset) generated during and/or analysed during the current study are available in the Zenodo—WormSwin: C. elegans Video Datasets repository, https://doi.org/10.5281/zenodo.7456803. The BBBC010 dataset is available at https://bbbc.broadinstitute.org/BBBC010. Source code and configuration files are available at https://github.com/bozeklab/worm-swin.

Received: 15 May 2023; Accepted: 5 July 2023
Published online: 07 July 2023

References

1. Marshall, J. D. *et al.* Continuous whole-body 3D kinematic recordings across the rodent behavioral repertoire. *Neuron* **109**, 420-437e8. https://doi.org/10.1016/j.neuron.2020.11.016 (2021).
2. Gosztolai, A. *et al.* LiftPose3D, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. *Nat. Methods* **18**, 975–981. https://doi.org/10.1038/s41592-021-01226-z (2021).
3. Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. X. & Schafer, W. R. A database of *Caenorhabditis elegans* behavioral phenotypes. *Nat. Methods* **10**, 877–879. https://doi.org/10.1038/nmeth.2560 (2013).
4. Barlow, I. L. *et al.* Megapixel camera arrays enable high-resolution animal tracking in multiwell plates. *Commun. Biol.* **5**, 253. https://doi.org/10.1038/s42003-022-03206-1 (2022).
5. Baek, J.-H., Cosman, P., Feng, Z., Silver, J. & Schafer, W. R. Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively. *J. Neurosci. Methods* **118**, 9–21. https://doi.org/10.1016/S0165-0270(02)00117-6 (2002).
6. Breiman, L., Friedman, J., Olshen, R. & Stone, C. Classification and regression trees. Wadsworth Int. *Group* **37**, 237–251 (1984).
7. Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. High-throughput behavioral analysis in *C. elegans*. *Nat. Methods* **8**, 592–598. https://doi.org/10.1038/nmeth.1625 (2011).

8. Javer, A. *et al.* An open-source platform for analyzing and sharing worm-behavior data. *Nat. Methods* **15**, 645–646. https://doi.org/10.1038/s41592-018-0112-1 (2018).
9. Hebert, L., Ahamed, T., Costa, A. C., O'Shaughnessy, L. & Stephens, G. J. WormPose: Image synthesis and convolutional networks for pose estimation in *C. elegans*. *PLoS Comput. Biol.* **17**, e1008914. https://doi.org/10.1371/journal.pcbi.1008914 (2021).
10. He, K., Zhang, X., Ren, S. & Sun, J. Identity mappings in deep residual networks. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 630–645 (Springer, 2016).
11. Wählby, C. *et al.* An image analysis toolbox for high-throughput *C. elegans* assays. *Nat. Methods* **9**, 714–716. https://doi.org/10.1038/nmeth.1984 (2012).
12. Stirling, D. R. *et al.* Cell Profiler 4: Improvements in speed, utility and usability. *BMC Bioinform.* **22**, 433. https://doi.org/10.1186/s12859-021-04344-9 (2021).
13. 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*. *Anim. Behav. Cogn.*https://doi.org/10.1101/2022.08.18.504475 *(2022).*
14. Jocher, G. YOLOv5 by Ultralytics. https://doi.org/10.5281/zenodo.3908559 (2020).
15. Du, Y., Song, Y., Yang, B. & Zhao, Y. Strongsort: Make deepsort great again. https://doi.org/10.48550/ARXIV.2202.13514 (2022).
16. Fudickar, S., Nustede, E. J., Dreyer, E. & Bornhorst, J. Mask R-CNN based *C. elegans* detection with a DIY microscope. *Biosensors* **11**, 257. https://doi.org/10.3390/bios11080257 (2021).
17. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (2017).
18. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778 (2016).
19. Mais, L., Hirsch, P. & Kainmueller, D. Patchperpix for instance segmentation. In *European Conference on Computer Vision*, 288–304 (Springer, 2020).
20. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).
21. Lalit, M., Tomancak, P. & Jug, F. Embedding-based instance segmentation in microscopy. In *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, 399–415 (PMLR, 2021).
22. Romera, E., Álvarez, J. M., Bergasa, L. M. & Arroyo, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst.* **19**, 263–272. https://doi.org/10.1109/TITS.2017.2750080 (2018).
23. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
24. Bodla, N., Singh, B., Chellappa, R. & Davis, L. S. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision*, 5561–5569 (2017).
25. Lopes, A. F. C. *et al.* A *C. elegans* model for neurodegeneration in Cockayne syndrome. *Nucleic Acids Res.* **48**, 10973–10985. https://doi.org/10.1093/nar/gkaa795 (2020).
26. Ljosa, V., Sokolnicki, K. L. & Carpenter, A. E. Annotated high-throughput microscopy image sets for validation. *Nat. Methods* **9**, 637–637. https://doi.org/10.1038/nmeth.2083 (2012).
27. Chen, K. *et al.* Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
28. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).
29. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021* (OpenReview.net, 2021).
30. Deng, J. *et al.* ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848 (2009).
31. Cai, Z. & Vasconcelos, N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*https://doi.org/10.1109/tpami.2019.2956516 *(2019).*
32. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456 (PMLR, 2015).
33. Wu, Y. & He, K. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19 (2018).
34. Qiao, S., Wang, H., Liu, C., Shen, W. & Yuille, A. Micro-batch training with batch-channel normalization and weight standardization. arXiv:1903.10520 (arXiv preprint) (2019).
35. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2018).
36. Loshchilov, I. & Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations* (2017).
37. Goyal, P. *et al.* Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv:1706.02677 (arXiv preprint) (2017).
38. Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V. & Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
39. Chen, K. *et al.* MMDetection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155 (arXiv preprint) (2019).
40. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014* (eds Fleet, D. *et al.*) 740–755 (Springer, 2014).
41. Fontaine, E., Burdick, J. & Barr, A. Automated tracking of multiple *C. Elegans*. In *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference***2006**, 3716–3719. https://doi.org/10.1109/IEMBS.2006.260657 (2006).
42. Alonso, A. & Kirkegaard, J. B. Fast spline detection in high density microscopy data. arXiv:2301.04460 (2023).

Acknowledgements

We would like to thank Matthias Rieckher for supplying us the videos of the CSB-1 dataset, as well as Xiao-Liu Chu for supplying the Mating Dataset video. We thank everyone who helped us annotating the data used in this publication. Maurice Deserno and Katarzyna Bozek were supported by the North Rhine-Westphalia return program (311-8.03.03.02-147635), BMBF program Junior Group Consortia in Systems Medicine (01ZX1917B) and hosted by the Center for Molecular Medicine Cologne. We thank the Regional Computing Center of the University of Cologne (RRZK) for providing computing time on the DFG-funded (Funding number: INST 216/512/1FUGG) High Performance Computing (HPC) system CHEOPS as well as support.

Author contributions

M.D.: methodology, software, experiments and analysis. K.B.: supervision, data and funding acquisition. M.D. and K.B. writing the article. All authors reviewed the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to M.D.

Chapter 3

Representation learning of C. elegans behavior

Quantifying an organism's behavior is an important part of scientific experiments, helping to understand complex biological mechanisms like effects of toxins, pharmaceuticals and even disease. Previous methods focus on computing hand-engineered features, based on a predicted center-line along C. elegans body. Here we present our deep learning approach [2] for self-supervised learning of C. elegans poses and behavior sequences. We project the learned embeddings into 3D space by applying UMAP [46] and color the embeddings by hand-engineered features to visualize similarities and patterns in the learned features. Using this method we are able to capture the same hand-engineered features other approaches compute, but without limiting our method to those features.

Unsupervised Representation Learning of C. elegans Poses and Behavior Sequences From Microscope Video Recordings

Maurice Deserno[1, 2, 4, *] **and Katarzyna Bozek**[1, 2, 3]

[1]Center for Molecular Medicine Cologne (CMMC), Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, North Rhine-Westphalia, Germany
[2]Institute for Biomedical Informatics, Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, North Rhine-Westphalia, Germany
[3]Cologne Excellence Cluster on Cellular Stress Responses in Aging- Associated Diseases (CECAD), University of Cologne, Cologne, North Rhine-Westphalia, Germany
[4]Faculty of Mathematics and Natural Sciences, University of Cologne, Cologne, North Rhine-Westphalia, Germany
[*]maurice.deserno@uni-koeln.de

ABSTRACT

Caenorhabditis elegans (*C. elegans*) is an important model system for studying molecular mechanisms in disease and aging. The nematode can be imaged in highly parallel phenotypic screens resulting in large volumes of video data of the moving worm. However converting the rich, pixel-encoded phenotypical information into meaningful, quantitative description of behavior is a challenging task. There is a range of methods for quantification of the simple body shape of *C. elegans* and the features of its motion. These methods however are often multi-step and fail in the case of highly coiled and self-overlapping worms. Motivated by the recent development of self-supervised deep learning methods in computer vision and natural language processing, we propose an unbiased, label-free approach to quantify worm pose and motion from video data directly. We represent worm posture and behavior as embedding vectors and visualize them in a unified embeddings space. We observe that the vector embeddings capture meaningful features describing worm shape and motion, such as the degree of body bend or the speed of movement. Importantly, using pixel values directly as input, our method captures coiled worm behaviors which are inaccessible to methods based on keypoint tracking or skeletonization. While our work focuses on *C. elegans*, the ability to quantify behavior directly from video data opens possibilities to study organisms without rigid skeletons whose behavior is difficult to quantify using keypoint-based approaches.

1 Introduction

Behavior is a window to an animal's nervous system. Precise quantification of behavior allows to determine fine phenotypic effects of genetic mutations or pharmacological interventions and, eventually, their underlying neural mechanisms. Keypoint tracking methods and motion tracking imaging systems have enabled acquiring precise information on animal posture and its change in time in natural settings[1–3]. It is however unclear how to quantitatively measure behavior of invertebrate species with flexible bodies and appendages. Organisms such as worms lack natural skeletons and hence distinct keypoints on their bodies. The shape of *C. elegans* is typically represented as its central body line and reduced to eigenworms[4] that enable quantification e.g. of the motion features and dynamics. However, this approach fails in the case of coiled or self-intersecting poses of *C. elegans* and current solutions apply multi-step approaches[5] to resolve these shapes.

Here we present a method for quantification of *C. elegans* motion based on video recordings directly. Unlike keypoint- or central body line-based approaches our method does not estimate the body structure but quantifies the behavior from the raw pixel values. Our method does not require any annotations but relies on self-supervised learning approach to learn sequence representations. This combination of self-supervision and keypoint-free pose estimation enables to forgo skeletonization and feature engineering which allows studying the full repertoire of *C. elegans* poses and behavior in a comprehensive manner.

2 Related Work

Previous tracking and pose estimation methods for *C. elegans* enabled a quantitative, automated analysis and a better understanding of its poses and behavior[4,6–11]. These methods allowed to comprehensively analyze worm behavior and to better understand phenotypic effects of genetic mutations, disease, or aging.

Stephens et al.[4] tracked *C. elegans* in microscopy videos and approximated their pose with a curve. They found that approximately 95% of the total variance in angles along the curve is represented by four eigenvalues. Based on these findings they introduced the term *eigenworms* as "templates" to describe the *C. elegans* poses. Javer et al.[9] developed a widely used single- and multiworm tracking software called Tierpsy. The software segments *C. elegans* and estimates their outlines and the skeleton. Additionally, it computes several hand-engineered features characterizing pose and motion of an individual e.g. the 6 eigenworms[4], the maximum amplitude of the skeleton's major axis, the degree of bend of different body segments, different body size measurements (such as length and width) or the motion mode (backward or forward). Both the eigenworms quantification and Tierpsy are based on classical computer vision approaches and do not allow to quantify coiled or overlapping poses of the worm. These poses are inaccessible to these methods.

Several methods address the challenge of accurate estimation of coiled and (self-)intersecting poses. WormPose[5] is a Residual Network[12](ResNet)-based method applying a multi-step approach that allows for estimating poses of coiling worms. To estimate the center line using equidistant keypoints, the method relies on video data with detected/annotated center lines (e.g. by Tierpsy) for frames prior to the occurrence of coiling behavior. The authors train their network with synthetically generated images of *C. elegans* to avoid time-consuming human labeling. The network learns to predict the two different centerlines resulting from different head/tail orientations. During evaluation a synthetic image is generated for each predicted centerline. By comparing the generated images to the input the best prediction is determined. Recent methods like DeepTangle by Alonso and Kirkegaard[13] and its extension DeepTangleCrawl[14] by Weheliye et al. enable robust skeletonization and tracking of *C. elegans* with overlaps and on a noisy background and allow better phenotypic screening. Still these methods fail when *C. elegans* are very tightly coiled or individuals lie parallel to each other over an extended time.

3 Methods

3.1 Datasets

All data we used for our experiments are publicly available on Zenodo[15] in the *Open Worm Movement Database*[9]. The data was downloaded using a python script filtering for specific parameters such as strain. For accessing the repository, we used the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). We created two datasets grouping genetic strains and long-term recordings of single individuals, respectively. The first dataset consists of seven genetic strains with one of them being the wild type. In the following we refer to this set as *strain dataset*.

This dataset includes a total of 165 videos of the following strains:

- N2
- AQ2932 (nca-2(gk5)III; unc-77(gk9)IV; nzIs29[punc-17::rho-1(G14V); punc-122::GFP])
- AQ2934 (nca-2(gk5);nzIs29[punc-17::rho-1(G14V); punc-122::GFP])
- TQ225 (trp-1(sy690)III)
- DG1856 (goa-1(sa734)I)
- DA609 (npr-1(ad609)X)
- VC731 (unc-63(ok1075)I)
- CB1141 (cat-4(e1141)V)

The second set consists of 71 videos of three strains with the same individuals recorded every day for multiple days (between 15 to 24 days per individual) during their adulthood. We call this last set the *aging dataset*. This dataset includes following genetic strains:

- AQ2947 (CGC N2 (Bristol, UK))
- OW940 (zgIs128[P(dat-1)::alpha-Synuclein::YFP])
- OW956 (zgIs144[P(dat-1)::YFP])

The data consists of video frames with masked background. Video data was recorded with frame rates varying between 25 frames per second (fps) and 32 fps. The videos of both sets have a length of almost 15 minutes each. Using Tierpsy[9,16] we calculated features of poses and motion in all our recordings. These features together with the worm genetic strain and age represent the metadata we use for interpretation of the image and sequence representations we developed in this study.

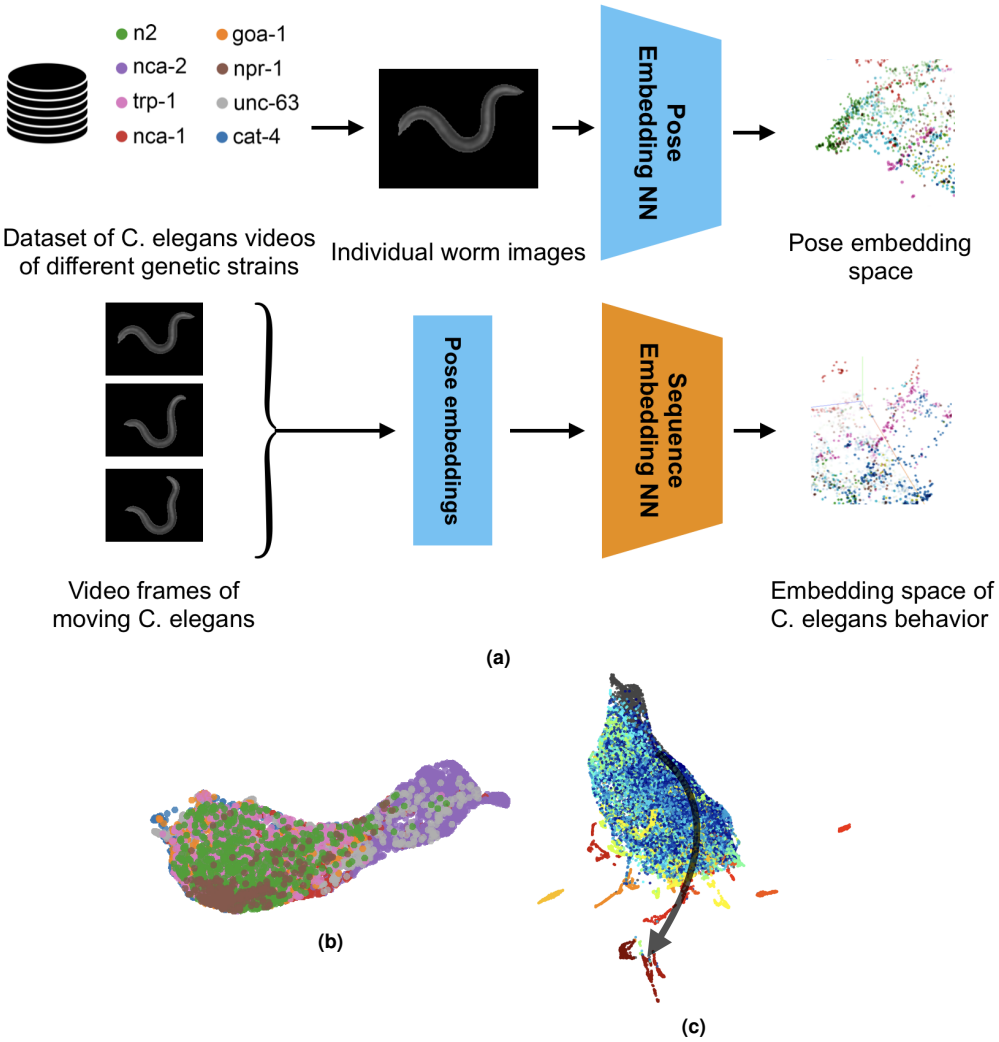


Figure 1. Processing pipeline and behavior representation. (a) Processing pipeline overview. We use a large set of video data of worm genetic strains and employ a contrastive learning approach to encode individual poses of the worm directly from the video frames. We next inspect these pose embeddings using their visualization in a 3D scatter plot. The trained pose embedding network is used to embed each video frame which is next an input to the sequence embedding network. Similarly to pose embeddings, we inspect the embedding space of worm behaviors using visualization techniques and motion features quantified with Tierpsy. (b) Visualization of the strain dataset behavior embedding space colored by the underlying genetic strain. (c) Visualization of the aging dataset behavior embedding space, illustrating the behavioral change with age in the direction of the arrow moving from young (blue) to old (red).

3.2 Data pre-processing

The background of the *C. elegans* images downloaded from Zenodo[15] is masked with black pixels. In some images the masking contains errors with background objects not masked out. Here we apply a combination of different methods including connected components and morphological operations to filter out smaller foreground blobs and to better match the background mask to the worm shape (similar to[5]). As a result we remove the errors and limit the foreground to one object only - the worm (see Fig. 2 "Data Preprocessing"). Further, we change the background mask pixel value from black (0) to gray (127). Next, we crop the foreground in the image, pad and resize it to a common image size of 128×128 pixels. This way, we remove excessive background pixels and center the object in the middle of the image while preserving its relative size. In the final step, we apply Principal Component Analysis (PCA) to rotate the object to a vertical orientation (Fig. 2).

We store the degree of rotation in addition to the strain and the day of adulthood in the aging dataset as the metadata. The metadata is not used in model training but in the model interpretation and visualization. We split the dataset into train, validation and test subsets with the proportions 0.76, 0.10, 0.14 and save the video frames as PyTorch[17] tensors in a .pt file per subset.

Following the data pre-processing, we train our deep learning approach, which includes two parts. The first part consists of a contrastive learning method to represent spatial poses of *C. elegans* based on their images. The second part is a Transformer encoder architecture that uses the learned pose representations to predict masked parts of a spatiotemporal sequence. In the following, we describe the two parts in detail.

3.3 Contrastive Learning for pose representations

We apply contrastive learning to learn representations of poses from *C. elegans* image data. It is a self-supervised approach that does not require labels. Specifically, we use a version of VICReg[18] adapted to our task. As backbone we chose ResNet18 over ResNet50 originally used in VICReg because of its smaller size. Our experiments suggested that the results do not improve using a larger feature extractor. We use a modified set of augmentations to ensure the network focuses on the important pose differences and learns to embed them rather than embedding the differences in e.g. lightning conditions or size of individuals. The output dimensionality was set to 64 with a hidden network dimensionality of 128.

To avoid having many similar poses in the training set, we subsample video frames by a factor of 10. We train the network using a batch size of 512 for 80 epochs on a NVIDIA Tesla V100-SXM2 with 32 GB of memory. As optimizer we chose AdamW[19] with a learning rate of 0.001. Additionally we use Cosine Annealing[20] as learning rate scheduler. The loss is calculated the same way as proposed by the authors of VICReg[18]: a weighted combination (compare with 1) of variance v, invariance s and covariance c loss with weights set to $\mu = 25.0$, $\lambda = 25.0$ and $\nu = 1.0$.

$$\ell(Z,Z') = \lambda s(Z,Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')] \tag{1}$$

3.4 Transformer encoder for sequence data imputation

To integrate the temporal component of behavior into the learned embeddings we employ a Transformer encoder neural network architecture[21–23]. The Transformer encoder consists of a multi-head attention block and a feed-forward network. This type of architecture has been used primarily in natural language processing (NLP) (e.g. by BERT[22]) and was later adapted to images (e.g. in Vision Transformer[23]). We attach the pre-trained pose representation network (see 3.3) as backbone to the Transformer network and freeze this backbone. We add a linear projection network to the last layer of the Transformer encoder network that infers embeddings of individual poses in the sequence.

During training, we input 12 ordered video frames as a sequence into the pre-trained pose representation network to generate pose embeddings. Here, we downsample the videos by a factor of 5 which is sufficient to capture the worm's motion in a smooth manner. With frame rates between $\sim 25 - 32$ fps (see section 3.1) this results in a sequence covering between ~ 2 - 1.6 seconds in real time. We store the ordered pose embeddings generated by the pose backbone as ground truth information for later evaluation.

Next, we construct sequences of 12 consecutive frame embeddings and attach frame rotation information generated during pre-processing. We mask the last 5 sequence elements by replacing them with zeroes (similar to[22,24]) before passing the sequence to the transformer network. We add sine-cosine positional encoding[21] and masked position encoding to the pose embeddings. The masked position encoding is a vector, indicating if a sequence element (frame) is masked (value 1 in the vector) or is not masked (value 0 in the vector)[24]. This vector is embedded and then added the same way as the positional encoding[25] (see Fig. 2). The pose embeddings together with positional and mask embeddings are the input to the transformer encoder. The Transformer network is trained to impute the missing values in the sequence. Using the linear projection network, pose embeddings and their rotations are predicted for each of the masked positions. We calculate the Mean Squared Error (MSE) loss between the embeddings generated by the pose representation network and the predictions of the linear projection network.

Pose representations have a dimensionality of 64 (see 3.3). The transformer uses a hidden dimensionality of 128 and consists of one encoding block and two heads. For training we use AdamW[19] as optimizer with a learning rate of 0.0005 for 250 epochs with a batch size of 64. The network was trained and tested on a NVIDIA Tesla V100-SXM2 with 32 GB of memory.

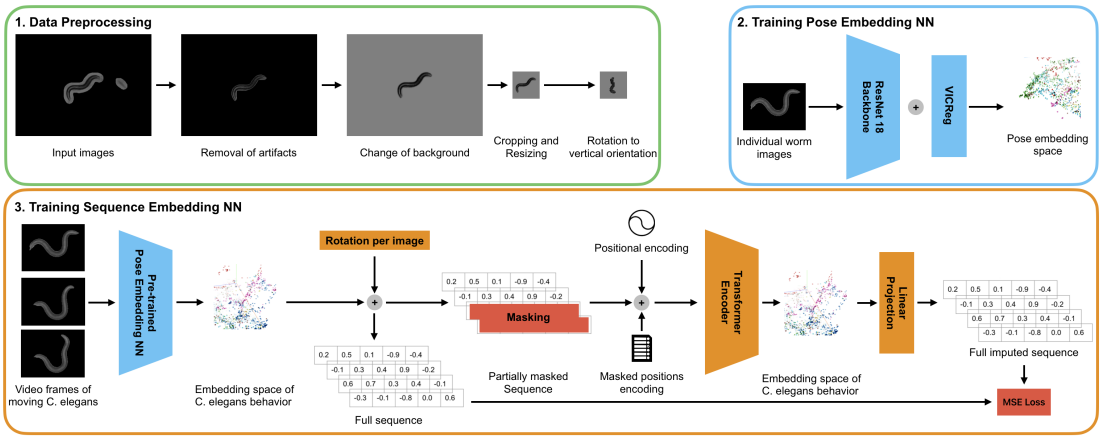


Figure 2. Data preprocessing and network architecture. 1) Data preprocessing pipeline: Artifacts are removed keeping only the worm as foreground object. We change the background to gray, crop the image to keep the worm centered and resize it to $128 \times 128px$. Finally, we rotate the worm to a vertical orientation. 2) A contrastive learning network is trained with images in random order to learn pose embeddings. 3) Using the ResNet-18 trained in (2) we embed sequences of 12 frames of moving *C. elegans*. Rotation information is concatenated with the encoded sequences and the last 5 frame embeddings are masked out. A Transformer-encoder learns behavior embeddings by imputing the masked sequence elements.

3.5 Visualization of pose and motion embeddings

To inspect the *C. elegans* pose and sequence embeddings we use the dimensionality reduction technique Uniform Manifold Approximation and Projection (UMAP)[26]. By applying UMAP, we reduce the embeddings to three dimensions to visualize them as scatter plots. We used the python implementation[1] of UMAP with the parameters *n_neighbors=30, min_dist=0.25, n_components=3* and *random_state=42* for the pose embedding space and for the behavior sequence embedding space.

4 Results

4.1 Pose representations

We first inspected the embeddings of individual worm poses. We project the embeddings in 3D using UMAP and inspect whether the embedding space reflects Tierpsy-based[9,16] pose features, as well as worm genetic strain. Figure 3a illustrates the pose embedding space of the strain datasets (see 3.1). This space shows a clear spatial ordering of poses according to their degree of bending (see Fig. 3a). While one end of the point cloud consists of strongly coiled worms, the opposite end clusters worms with poses close to a straight line. The points are colored according to the maximum amplitude of the bend along the worm body line. The straight poses have a low amplitude value, the more bent ones a higher one. There is a clear gradient of this value along the point cloud. However, the coiled worm shapes are missing this feature value (marked in gray color in Fig. 3a) as Tierpsy cannot resolve these poses[9,16]. This reveals an advantage of our approach: it allows to capture all worm poses, from straight to strongly coiled ones, in a uniform and smooth embedding space. Our approach groups coiling and bending poses together with a clear transition between them, whereas an important fraction of worm poses is not possible to quantify with skeletonization-based methods. Since our approach does not require any skeleton or keypoint estimations, it is robust against coiling and self-intersecting postures. A large number of embeddings in the gray area in Fig. 3a belongs to *C. elegans* of the AQ2934 strain.

4.2 Sequence representations

We next trained a Transformer-based approach to embed sequences of worm postures captured in a video recording. The Transformer network takes as input sequences of pose embeddings where the second half of each sequence is masked. The

[1] https://github.com/lmcinnes/umap

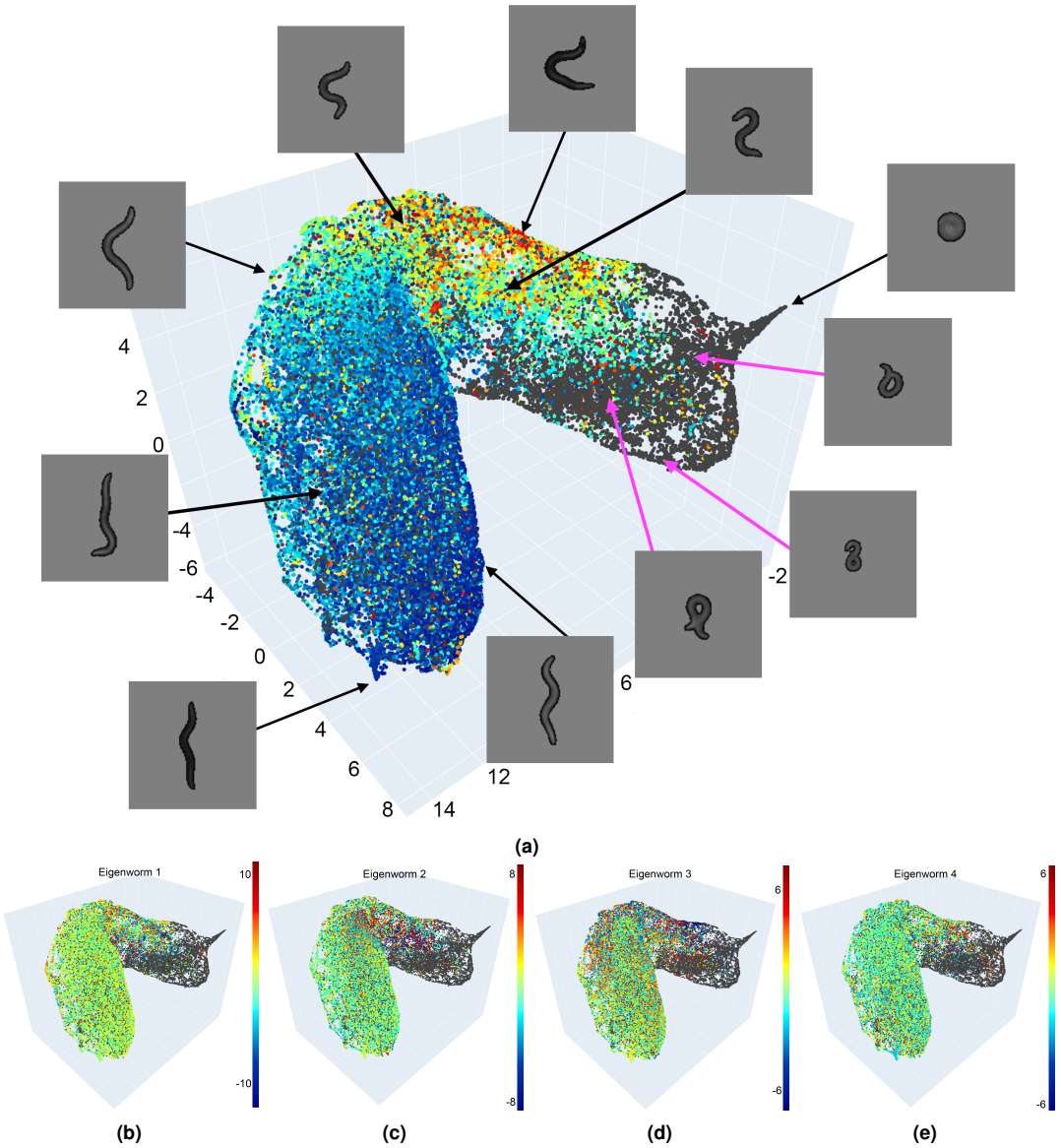


Figure 3. Visualization of the pose embedding space. (a) We reduced the embedding space to 3D using UMAP and colored it with the Tierpsy *max_amplitude* feature. Dark gray dots indicate poses for which this feature could not be quantified using Tierpsy. There is a gradient in coloring suggesting that similar poses occupy neighboring parts of the embedding space. Example images of poses are shown with an indication of their position in the embedding space. Strongly coiled and almost straight worms occupy opposite ends of the point cloud. (b-e) Pose embedding space colored according to their eigenworm 1 to 4 values.

network is trained to infer the masked part of the sequence as well as the rotation angle of the worm in the video. The MSE of the masked pose estimation in the strain dataset is 0.106 while of the rotation angle 0.0129, which represents an error of $\sim 20.47°$. Via this self-supervised approach the network learns representations of the sequences that encode worm posture, its change in time, and the dynamics of this motion in a comprehensive manner. Similar to the pose embeddings, we visualize the embedding space of *C. elegans* short-term behaviors in 3D (Fig. 4a).

This visualization shows a clear separation of sequences of the strain AQ2934 (labeled in orange) from sequences of the other strains. This separation was also present in the pose embedding space, and reflects the frequent and heavy coiling behavior of the AQ2934 strain. Behavior sequences of strain DA609 (marked in brown) are also grouped together in the embedding space. This strain is known for aggregating and burrowing behavior[27,28]. Next to the DA609 cluster is a larger area where behavior sequences of different strains mix. This likely occurs since most strains share common behaviors such as simple forward locomotion.

To further interpret the behavior embedding space, we colored it according to motion speed features quantified with Tierpsy (Fig. 4ab-c). We observed that sequences with faster movement are more frequent in the center of mass of the embedding space. This confirms our observation that crawling behavior, common to most of the strains, is located in this part of the embedding space.

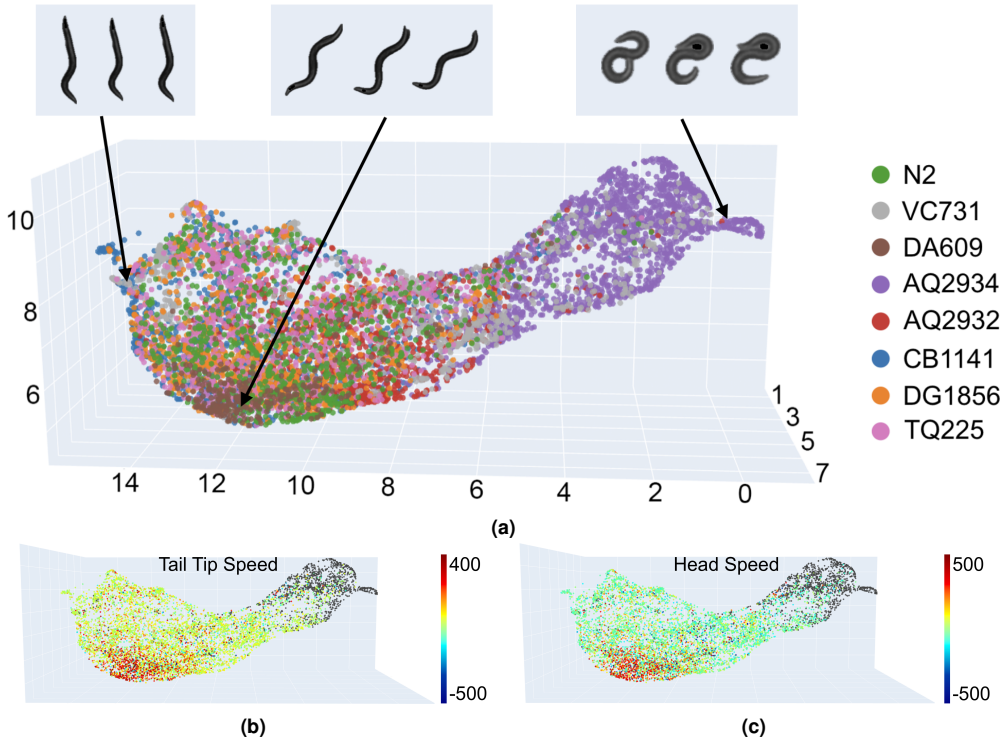


Figure 4. Behavior embedding space of the strain dataset. (a) Embedding space colored by strain. Worm images above correspond to 1st, 6th and 12th frame of three example sequences. (b) Embeddings space colored by tail tip speed and (c) head speed. Gray dots in (b) and (c) indicate sequences for which these Tierpsy features are missing.

4.3 Worm behavior changes with age

We next inspected the behavior embedding space of the aging dataset. This dataset contains 71 individuals that were recorded over their adulthood, for time span of up to 24 days. We employed our approach to inspect which parts of the embedding space those individuals occupy as they age (Fig. 5a). Young individuals appear to display a wide range of behaviors, while as they age their behavior repertoire reduces. Markedly, the patterns of aging in behavior are consistent among individuals. This can be seen in the embedding space labeled by day of adulthood (see Fig. 5a). Behaviors of individuals from day 1 to 10 span

a wide area in the space, while embeddings for day 10 to 15 cover much more limited areas at the bottom part of the point cloud. From day 15 onward the embeddings almost only form outlying groupings. The reason for these behaviors to localize on the outside of the embedding space can be two-fold. On the one hand these individuals move slower and assume fewer different poses which differ from those of more agile younger individuals with their typical crawling/swimming locomotion and coiling behaviors. On the other hand, old individuals are not included in the strain dataset on which the Transformer was trained. Although the strain dataset and the aging dataset were recorded in similar ways, the behaviors of older individuals were never seen by the network.

Additionally to the age color-coded embedding space, we plotted the trajectory in the embedding space of one individual of strain AQ2947 (Fig. 5b). This trajectory links behaviors of this worm as it ages. It illustrates the broad variety of the behavior of this individual up to day 15 after which the its behaviors are limited to the bottom part of the embedding space.

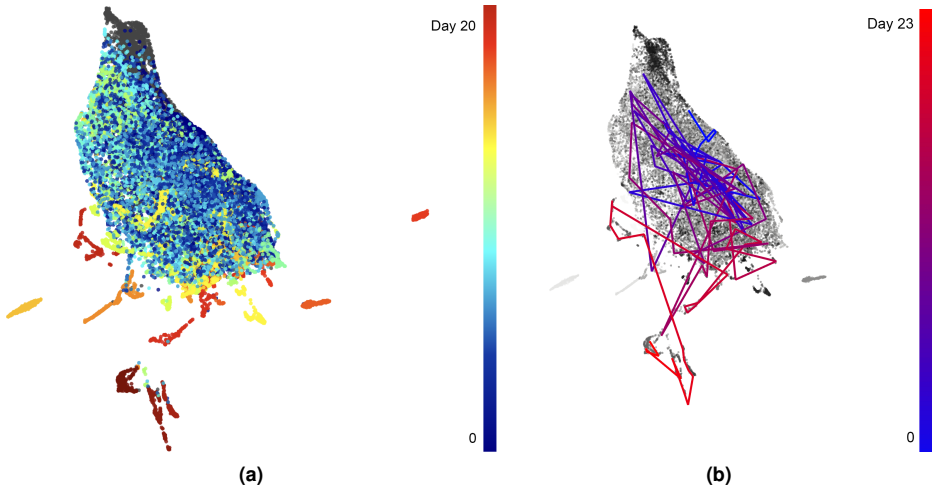


Figure 5. Behavior embedding space of the strain and aging datasets combined. (a) Embedding space colored by age. With age we refer to the day of adulthood of an individual *C. elegans*. Gray color indicates missing age data of worms from the strain dataset. (b) Behaviors of one individual linked over the course of its aging. Starting with blue at the last day of the L4 stage, progressing to red until the last recorded day (23) of adulthood.

5 Discussion

In this work, we presented a deep learning-based approach for representation learning of *C. elegans* poses and behavior sequences from bright-field microscopy videos without human annotations. Our method uses a combination of Contrastive Learning and a Transformer architecture originally developed for self-supervised learning in computer vision and NLP[22]. We draw inspiration from these methods to demonstrate that the pose and motion of *C. elegans* can be quantified in a meaningful manner without the use of labels. Contrary to previous approaches, our method does not require worm skeletonization, keypoints definition, or any pose or behavior categories. Our approach allows to embed all worm poses and pose sequences, from straight ones to the challenging poses of tightly coiling and strongly bending *C. elegans*. We demonstrate that, even though our methods are based exclusively on image pixel values, the resulting image and video embeddings reflect quantitative features describing the worm shape and its motion, such as degree of bend, eigenworms and speed of motion. We apply our method to the video data of different genetic strains as well as aging worms and illustrate the differences in behavior of worms of various strains and ages.

To summarize, the advantages of our approach are:

1. Embedding challenging poses without relying on annotations.
2. Quantifying previously inaccessible behaviors.
3. Capturing hand-engineered features without explicitly calculating them.

4. Ability to capture in a comprehensive manner properties of poses and behavior.

One limitation of our approach is the inability to distinguish between the head and tail of the worm. Head and tail movements are important elements of the worm behavior. Since *C. elegans* typically move head-first, the head/tail orientation can be estimated based on their direction of movement. However, for strains that frequently move backward, this rule would not apply and the head/tail orientation would need to be estimated based on their visual features. While this remains a challenging task, future work should incorporate predicted head/tail orientation as input to the network in our approach. Alternatively, video frames could be adjusted so that *C. elegans* always face head-up, rather than simply aligning all worms to a vertical orientation without considering head/tail direction.

While our approach offers many advantages over methods based on hand-engineered features, one drawback is its lower direct interpretability. For example, a feature such as head speed provides straightforward, low-level behavioral insights, whereas our embedding space visualizations combine all characteristics of the worm motion and are therefore more difficult to interpret, similar to eigenworm features[4]. On the other hand, the comprehensive motion embeddings derived from our method are a powerful representation for downstream tasks such as behavior or strain classification, reaching beyond analyses based individual motion features.

In this work we focused on behaviors spanning two seconds. Future experiments could explore embedding sequences with different time spans. Extending the length of the input video to four or eight seconds may allow to capture additional behaviors, from brief actions to prolonged activities such as mating. Longer videos can be incorporated in various ways, such as adjusting the step size between frames or increasing the sequence length.

Since pixel-based approaches like ours do not rely on skeleton or keypoints definition, they can be applied to any body form. This ability to quantify behavior directly from pixels opens possibilities to study a wide range of organisms, including cephalopods[29] or single-celled organisms with flagella or cilia[30,31] in a comprehensive manner.

6 Acknowledgments

We would like to thank Greg J. Stephens and André E. X. Brown for their valuable comments and discussions. Maurice Deserno and Katarzyna Bozek were supported by the North Rhine-Westphalia return program (311-8.03.03.02-147635), BMBF program Junior Group Consortia in Systems Medicine (01ZX1917B) and hosted by the Center for Molecular Medicine Cologne.

7 Author contributions statement

M.D.: methodology, software, experiments and analysis. K.B.: supervision, data and funding acquisition. M.D. and K.B. writing the article. All authors reviewed the manuscript.

8 Competing Interests Statement

The author(s) declare no competing interests.

References

1. Hu, B. *et al.* 3d mouse pose from single-view video and a new dataset. *Sci. Reports* **13**, 13554 (2023).
2. Karashchuk, P. *et al.* Anipose: A toolkit for robust markerless 3d pose estimation. *Cell Reports* **36**, 109730, DOI: https://doi.org/10.1016/j.celrep.2021.109730 (2021).
3. Mathis, A. *et al.* Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* (2018).
4. Stephens, G. J., Johnson-Kerner, B., Bialek, W. & Ryu, W. S. Dimensionality and dynamics in the behavior of c. elegans. *PLOS Comput. Biol.* **4**, 1–10, DOI: 10.1371/journal.pcbi.1000028 (2008).
5. Hebert, L., Ahamed, T., Costa, A. C., O'Shaughnessy, L. & Stephens, G. J. Wormpose: Image synthesis and convolutional networks for pose estimation in c. elegans. *PLOS Comput. Biol.* **17**, 1–20, DOI: 10.1371/journal.pcbi.1008914 (2021).
6. Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. & Schafer, W. R. A database of caenorhabditis elegans behavioral phenotypes. *Nat. methods* **10**, 877–879 (2013).
7. Nagy, S., Goessling, M., Amit, Y. & Biron, D. A generative statistical algorithm for automatic detection of complex postures. *PLOS Comput. Biol.* **11**, e1004517 (2015).
8. Baek, J.-H., Cosman, P., Feng, Z., Silver, J. & Schafer, W. R. Using machine vision to analyze and classify caenorhabditis elegans behavioral phenotypes quantitatively. *J. neuroscience methods* **118**, 9–21 (2002).

9. Javer, A. *et al.* An open-source platform for analyzing and sharing worm-behavior data. *Nat. methods* **15**, 645–646 (2018).

10. Restif, C. *et al.* Celest: computer vision software for quantitative analysis of c. elegans swim behavior reveals novel features of locomotion. *PLoS computational biology* **10**, e1003702 (2014).

11. Nagy, S., Goessling, M., Amit, Y. & Biron, D. A generative statistical algorithm for automatic detection of complex postures. *PLOS Comput. Biol.* **11**, 1–23, DOI: 10.1371/journal.pcbi.1004517 (2015).

12. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034, DOI: 10.1109/ICCV.2015.123 (2015).

13. Alonso, A. & Kirkegaard, J. B. Fast detection of slender bodies in high density microscopy data. *Commun. Biol.* **6**, 754, DOI: 10.1038/s42003-023-05098-1 (2023).

14. Weheliye, W. H., Rodriguez, J., Feriani, L., Javer, A. & Brown, A. E. An improved neural network model enables worm tracking in challenging conditions and increases signal-to-noise ratio in phenotypic screens. *bioRxiv* DOI: 10.1101/2024.12.20.629717 (2024). https://www.biorxiv.org/content/early/2024/12/21/2024.12.20.629717.full.pdf.

15. European Organization For Nuclear Research & OpenAIRE. Zenodo, DOI: 10.25495/7GXK-RD71 (2013).

16. Javer, A., Ripoll-Sánchez, L. & Brown, A. E. Powerful and interpretable behavioural features for quantitative phenotyping of caenorhabditis elegans. *Philos. Transactions Royal Soc. B: Biol. Sci.* **373**, 20170375 (2018).

17. Ansel, J. *et al.* PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, DOI: 10.1145/3620665.3640366 (ACM, 2024).

18. Bardes, A., Ponce, J. & LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations* (2022).

19. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2017).

20. Loshchilov, I. & Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations* (2017).

21. Vaswani, A. *et al.* Attention is all you need. In *Neural Information Processing Systems* (2017).

22. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics* (2019).

23. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations* (2021).

24. Du, W., Côté, D. & Liu, Y. Saits: Self-attention-based imputation for time series. *Expert. Syst. with Appl.* **219**, 119619 (2023).

25. Rose, F. *et al.* Deep imputation for skeleton data (disk) for behavioral science. *bioRxiv* 2024–05 (2024).

26. McInnes, L., Healy, J., Saul, N. & Großberger, L. Umap: Uniform manifold approximation and projection. *J. Open Source Softw.* **3**, 861 (2018).

27. Ding, S. S., Romenskyy, M., Sarkisyan, K. S. & Brown, A. E. X. Measuring caenorhabditis elegans spatial foraging and food intake using bioluminescent bacteria. *Genetics* **214**, 577–587, DOI: 10.1534/genetics.119.302804 (2020). https://academic.oup.com/genetics/article-pdf/214/3/577/42105699/genetics0577.pdf.

28. López-Puebla, A., Mayoral-Peña, Z., Gómez-Cepeda, K. & Arellano-Carbajal, F. Caenorhabditis elegans daf-7 mutants exhibit burrowing behavior, DOI: 10.17912/MICROPUB.BIOLOGY.000172 (2019).

29. Woo, T. *et al.* The dynamics of pattern matching in camouflaging cuttlefish. *Nature* **619**, 122–128, DOI: 10.1038/s41586-023-06259-2 (2023).

30. Wan, K. Y. *et al.* Reorganization of complex ciliary flows around regenerating stentor coeruleus. *Philos. Transactions Royal Soc. B: Biol. Sci.* **375**, 20190167, DOI: 10.1098/rstb.2019.0167 (2020). https://royalsocietypublishing.org/doi/pdf/10.1098/rstb.2019.0167.

31. Wan, K. Y. Synchrony and symmetry-breaking in active flagellar coordination. *Philos. Transactions Royal Soc. B: Biol. Sci.* **375**, 20190393, DOI: 10.1098/rstb.2019.0393 (2020). https://royalsocietypublishing.org/doi/pdf/10.1098/rstb.2019.0393.

Chapter 4

Conclusion

In this work, we presented our contributions to automate behavior quantification of C. elegans using state-of-the-art Deep Learning (DL) methods. First, we presented an instance segmentation approach to extract pixel-level pose information in dense scenes [1]. The synthetic training data contains hard samples like coiling and overlapping individuals for which previous methods struggled to estimate accurate pose information. Compared to skeletonization and keypoint-based approaches that struggle when it comes to tightly coiling individuals, our approach is able to segment them with high accuracy. We tested our method on different challenging datasets and showed its capabilities when combined with a tracking approach.

Second, we presented a behavior representation learning approach capable to embed previously inaccessible behavior of C. elegans [2]. Since we based our method on self-supervised learning, it does not require any hand-labeled pose or behavior annotations. Not requiring any labels allowed us to choose freely what data we like to use for training and testing our approach. We chose to train our method on videos of wild-type (N2) C. elegans and seven mutant strains. For testing, we used the hold-out data of the same dataset used for training. Additionally, we evaluated our approach on a dataset of aging C. elegans. For this dataset the nematodes were recorded every day during their adulthood until they die. This data enables us to analyze how behavior changes with age. Visualizing the learned embedding space and coloring the embeddings based on various features computed by Tierpsy [18, 19] revealed clusters and color gradients in the data. Especially when looking at the embedding data of aging worm behavior, it becomes visible that young worms express a much higher behavior repertoire than older worms. The amount of different body poses they take on declines over time and, latest at the 15^{th} day of adulthood, the speed at which this repertoire shrinks, speeds up drastically. The clusters and gradients that become visible when coloring the embeddings by features computed with Tierpsy [18, 19], demonstrate that our approach is able to capture these hand-engineered features. As our approach is not explicitly trained to learn these features, it is able to discover different properties including novel ones, not captured by previously hand-engineered features.

With our research, we further closed the gap between automated high-throughput data acquisition and incomplete behavior quantification due to previously inaccessible poses and behavior. Our contributions have shown the potential of DL methods in pose estimation and behavior quantification for ethological studies.

Additional to the future work mentioned in chapter 2 and 3, our method could be applied to more strains of C. elegans that express challenging behavior and compare the results to wild-type strain to get a map of shared, similar and unique behavior comparable to the work of Brown et al. [17]. Mapping phenotypic relations helps scientists to understand the influence of different genetic mutations.

Another area of future work is to use our methods for downstream tasks like strain classification, action recognition and forecasting, as well as generation of synthetic data. Strain classification can be an entry point, since the data we used for training comes with strain labels usable to train a classification head. Action recognition would likely require additional labeled data, making it more challenging to start with. Nevertheless action recognition has great potential as it could be used to generate comprehensible behavior descriptions. As described in chapter 3, the repertoire of behavior of C. elegans declines with age. Based on the assumption, that this is a general property of all genetic strains, a DL approach can be trained to predict a *fitness score* for an individual nematode. This score can be used to describe how mobile an individual, or even a strain, is compared to the wild-type or other individuals of the same strain. Using this mobility estimate, a live span prediction could be made [47].

As our behavior representation method (see chapter 3) is trained to impute the masked-out end of a behavior sequence, it can be used as a forecasting model for C. elegans behavior. This type of network can be combined with segmentation and tracking, as forecasting the next moves of an individual does help locating it in the next frames, especially in crowded scenes. By changing the network to give a probabilistic output, forecasting could be further improved. This idea is similar to the functionality of a Kalman filter [48] which is widely used in tracking approaches [49, 50]. Probabilistic approaches are also used in recent keypoint-based pose estimation methods, applied to human and animal image data [51, 52]. From forecasting we could move on to synthetic data generation. Inputting a pose or short behavior sequence as starting point, the network could generate synthetic behavior data to train other networks. Generating images for behavior sequences can be solved by training a decoder network together with our pose embedding network (see chapter 3). The decoder network can be trained to predict the image, fed into the pose embedding network, based on embedding vector output (see e.g. Variational autoencoders (VAEs) [53]). This way, generated behavior embeddings could be decoded to image sequences. Additionally, the behavior representation network could be trained with behavior sequences annotated with strain class or action labels. This would enable us to choose a specific *behavior style* e.g. coiling, mating, or even wild-type behavior, to be generated by the network. Behavior data generated by such a method would come with action/activity labels and could in turn be used as synthetic training dataset for other methods, to avoid time-consuming hand labeling.

With novel DL methods and researchers applying them to study model organisms like C. elegans, we can expect improving results in the coming years. Today, DL methods are already able to automate many processes that were previously carried out manually. This automatization saves time that can be used to conduct more experiments or spend more time on interpreting the results. Speeding up biomedical research helps scientists to understand biological effects in less time and find cures for some of the worst disease of today.

Bibliography

[1] Maurice Deserno and Katarzyna Bozek. WormSwin: Instance segmentation of C. elegans using vision transformer. *Scientific Reports*, 13(1):11021, July 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-38213-7. URL https://doi.org/10.1038/s41598-023-38213-7.

[2] Maurice Deserno and Katarzyna Bozek. Unsupervised representation learning of c. elegans poses and behavior sequences from microscope video recordings. *bioRxiv*, 2025. doi: 10.1101/2025.02.14.638285. URL https://www.biorxiv.org/content/early/2025/02/19/2025.02.14.638285.

[3] Maurice Deserno and Katarzyna Bozek. Wormswin: C. elegans video datasets, April 2023. URL https://doi.org/10.5281/zenodo.7456803.

[4] Rachel A. Ankeny and Sabina Leonelli. What's so special about model organisms? *Studies in History and Philosophy of Science Part A*, 42(2):313–323, 2011. ISSN 0039-3681. doi: https://doi.org/10.1016/j.shpsa.2010.11.039. URL https://www.sciencedirect.com/science/article/pii/S0039368110001184. Model-Based Representation in Scientific Practice.

[5] Siwen Zhang, Fei Li, Tong Zhou, Guixia Wang, and Zhuo Li. Caenorhabditis elegans as a useful model for studying aging mutations. *Frontiers in Endocrinology*, 11, 2020. ISSN 1664-2392. doi: 10.3389/fendo.2020.554994. URL https://www.frontiersin.org/journals/endocrinology/articles/10.3389/fendo.2020.554994.

[6] Xiao Xu and Stuart K. Kim. The early bird catches the worm: new technologies for the caenorhabditis elegans toolkit. *Nature Reviews Genetics*, 12(11):793–801, 2011. doi: 10.1038/nrg3050. URL https://doi.org/10.1038/nrg3050.

[7] Maxwell C. K. Leung, Phillip L. Williams, Alexandre Benedetto, Catherine Au, Kirsten J. Helmcke, Michael Aschner, and Joel N. Meyer. Caenorhabditis elegans: An emerging model in biomedical and environmental toxicology. *Toxicological Sciences*, 106(1):5–28, 06 2008. ISSN 1096-6080. doi: 10.1093/toxsci/kfn121. URL https://doi.org/10.1093/toxsci/kfn121.

[8] Piper Reid Hunt. The c. elegans model in toxicity testing. *Journal of Applied Toxicology*, 37(1):50–59, 2017. doi: https://doi.org/10.1002/jat.

3357. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/jat.3357.

[9] Javier Apfeld and Scott Alper. *What Can We Learn About Human Disease from the Nematode C. elegans?*, pages 53–75. Springer New York, New York, NY, 2018. ISBN 978-1-4939-7471-9. doi: 10.1007/978-1-4939-7471-9_4. URL https://doi.org/10.1007/978-1-4939-7471-9_4.

[10] Maria Markaki and Nektarios Tavernarakis. Caenorhabditis elegans as a model system for human diseases. *Current Opinion in Biotechnology*, 63:118–125, 2020. ISSN 0958-1669. doi: https://doi.org/10.1016/j.copbio.2019.12.011. URL https://www.sciencedirect.com/science/article/pii/S0958166919301491. Nanobiotechnology, Systems Biology.

[11] Thomas J O'Brien, Ida L Barlow, Luigi Feriani, and André EX Brown. Systematic creation and phenotyping of mendelian disease models in c. elegans: towards large-scale drug repurposing. *eLife Sciences Publications, Ltd*, December 2024. doi: 10.7554/elife.92491.3. URL http://dx.doi.org/10.7554/eLife.92491.3.

[12] Steven J. Cook, Travis A. Jarrell, Christopher A. Brittin, Yi Wang, Adam E. Bloniarz, Maksim A. Yakovlev, Ken C. Q. Nguyen, Leo T. H. Tang, Emily A. Bayer, Janet S. Duerr, Hannes E. Bülow, Oliver Hobert, David H. Hall, and Scott W. Emmons. Whole-animal connectomes of both caenorhabditis elegans sexes. *Nature*, 571(7763): 63–71, 2019. doi: 10.1038/s41586-019-1352-7. URL https://doi.org/10.1038/s41586-019-1352-7.

[13] The C. elegans Sequencing Consortium*. Genome sequence of the nematode c. elegans: A platform for investigating biology. *Science*, 282(5396):2012–2018, 1998. doi: 10.1126/science.282.5396.2012. URL https://www.science.org/doi/abs/10.1126/science.282.5396.2012.

[14] Theresa Stiernagle. Maintenance of C. elegans. In *WormBook: The Online Review of C. elegans Biology [Internet]*. WormBook, February 2006. URL https://www.ncbi.nlm.nih.gov/books/NBK19649/.

[15] Linda P. O'Reilly, Cliff J. Luke, David H. Perlmutter, Gary A. Silverman, and Stephen C. Pak. C. elegans in high-throughput drug discovery. *Advanced Drug Delivery Reviews*, 69-70:247–253, 2014. ISSN 0169-409X. doi: https://doi.org/10.1016/j.addr.2013.12.001. URL https://www.sciencedirect.com/science/article/pii/S0169409X13002871. Innovative tissue models for drug discovery and development.

[16] Eviatar Yemini, Tadas Jucikas, Laura J Grundy, AndréE X Brown, and William R Schafer. A database of caenorhabditis elegans behavioral phenotypes. *Nature Methods*, 10(9):877–879, 2013. doi: 10.1038/nmeth.2560. URL https://doi.org/10.1038/nmeth.2560.

[17] André E. X. Brown, Eviatar I. Yemini, Laura J. Grundy, Tadas Jucikas, and William R. Schafer. A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion. *Proceedings of the National Academy of Sciences*, 110(2):791–796, 2013. doi: 10.1073/pnas.1211447110. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1211447110`.

[18] Avelino Javer, Lidia Ripoll-Sánchez, and André E.X. Brown. Powerful and interpretable behavioural features for quantitative phenotyping of caenorhabditis elegans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170375, 2018. doi: 10.1098/rstb.2017.0375. URL `https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2017.0375`.

[19] Avelino Javer, Michael Currie, Chee Wai Lee, Jim Hokanson, Kezhi Li, Céline N. Martineau, Eviatar Yemini, Laura J. Grundy, Chris Li, QueeLim Ch'ng, William R. Schafer, Ellen A. A. Nollen, Rex Kerr, and AndréE. X. Brown. An open-source platform for analyzing and sharing worm-behavior data. *Nature Methods*, 15(9): 645–646, 2018. doi: 10.1038/s41592-018-0112-1. URL `https://doi.org/10.1038/s41592-018-0112-1`.

[20] Mario de Bono, David M. Tobin, M. Wayne Davis, Leon Avery, and Cornelia I. Bargmann. Social feeding in caenorhabditis elegans is induced by neurons that detect aversive stimuli. *Nature*, 419(6910):899–903, 2002. doi: 10.1038/nature01169. URL `https://doi.org/10.1038/nature01169`.

[21] Siyu Serena Ding, Linus J Schumacher, Avelino E Javer, Robert G Endres, and André EX Brown. Shared behavioral mechanisms underlie *C. elegans* aggregation and swarming. *eLife*, 8:e43318, apr 2019. ISSN 2050-084X. doi: 10.7554/eLife.43318. URL `https://doi.org/10.7554/eLife.43318`.

[22] Jonathan Lipton, Gunnar Kleemann, Rajarshi Ghosh, Robyn Lints, and Scott W. Emmons. Mate searching in caenorhabditis elegans: A genetic model for sex drive in a simple invertebrate. *Journal of Neuroscience*, 24(34):7427–7434, 2004. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1746-04.2004. URL `https://www.jneurosci.org/content/24/34/7427`.

[23] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Steffen Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, Venkatesh N. Murthy, George Lauder, Catherine Dulac, Mackenzie Weygandt Mathis, and Alexander Mathis. Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods*, 19(4):496–504, 2022. doi: 10.1038/s41592-022-01443-0. URL `https://doi.org/10.1038/s41592-022-01443-0`.

[24] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[25] Bruno Artacho and Andreas Savakis. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] Joong-Hwan Baek, Pamela Cosman, Zhaoyang Feng, Jay Silver, and William R Schafer. Using machine vision to analyze and classify caenorhabditis elegans behavioral phenotypes quantitatively. *Journal of Neuroscience Methods*, 118(1):9–21, 2002. ISSN 0165-0270. doi: https://doi.org/10.1016/S0165-0270(02)00117-6. URL https://www.sciencedirect.com/science/article/pii/S0165027002001176.

[27] Jean Paul Frédéric Serra. Image analysis and mathematical morphology. 1983. URL https://api.semanticscholar.org/CorpusID:62066269.

[28] L. Breiman, J. H. Freidman, Richard A. Olshen, and C. J. Stone. Cart: Classification and regression trees. 1984. URL https://api.semanticscholar.org/CorpusID:59814698.

[29] Nicholas A Swierczek, Andrew C Giles, Catharine H Rankin, and Rex A Kerr. High-throughput behavioral analysis in c. elegans. *Nature Methods*, 8(7):592–598, 2011. doi: 10.1038/nmeth.1625. URL https://doi.org/10.1038/nmeth.1625.

[30] Greg J. Stephens, Bethany Johnson-Kerner, William Bialek, and William S. Ryu. Dimensionality and dynamics in the behavior of c. elegans. *PLOS Computational Biology*, 4(4):1–10, 04 2008. doi: 10.1371/journal.pcbi.1000028. URL https://doi.org/10.1371/journal.pcbi.1000028.

[31] Ida L. Barlow, Luigi Feriani, Eleni Minga, Adam McDermott-Rouse, Thomas James O'Brien, Ziwei Liu, Maximilian Hofbauer, John R. Stowers, Erik C. Andersen, Siyu Serena Ding, and AndréE. X. Brown. Megapixel camera arrays enable high-resolution animal tracking in multiwell plates. *Communications Biology*, 5(1): 253, 2022. doi: 10.1038/s42003-022-03206-1. URL https://doi.org/10.1038/s42003-022-03206-1.

[32] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

[33] Shoubhik Chandan Banerjee, Khursheed Ahmad Khan, and Rati Sharma. Deep-wormtracker: Deep learning methods for accurate detection and tracking for behavioral studies in c. elegans. *Applied Animal Behaviour Science*, 266:106024, 2023. ISSN 0168-1591. doi: https://doi.org/10.1016/j.applanim.2023.106024. URL https://www.sciencedirect.com/science/article/pii/S016815912300196X.

[34] Glenn Jocher. YOLOv5 by Ultralytics, May 2020. URL https://github.com/ultralytics/yolov5.

[35] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 25:8725–8737, 2023. doi: 10.1109/TMM.2023.3240881.

[36] Laetitia Hebert, Tosif Ahamed, Antonio C. Costa, Liam O'Shaughnessy, and Greg J. Stephens. Wormpose: Image synthesis and convolutional networks for pose estimation in c. elegans. *PLOS Computational Biology*, 17(4):1–20, 04 2021. doi: 10.1371/journal.pcbi.1008914. URL https://doi.org/10.1371/journal.pcbi.1008914.

[37] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

[38] Albert Alonso and Julius B. Kirkegaard. Fast detection of slender bodies in high density microscopy data. *Communications Biology*, 6(1):754, July 2023. ISSN 2399-3642. doi: 10.1038/s42003-023-05098-1. URL https://doi.org/10.1038/s42003-023-05098-1.

[39] Weheliye H Weheliye, Javier Rodriguez, Luigi Feriani, Avelino Javer, and André EX Brown. An improved neural network model enables worm tracking in challenging conditions and increases signal-to-noise ratio in phenotypic screens. *bioRxiv*, 2024. doi: 10.1101/2024.12.20.629717. URL https://www.biorxiv.org/content/early/2024/12/21/2024.12.20.629717.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[41] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4969–4978, 2019. doi: 10.1109/CVPR.2019.00511.

[42] Maurice Deserno and Katarzyna Bozek. Wormswin - model weights, August 2023. URL https://doi.org/10.5281/zenodo.8254728.

[43] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[44] Adrien Bardes, Jean Ponce, and Yann Lecun. VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning. In *ICLR 2022 - International Conference on Learning Representations*, Online, United States, April 2022. URL https://inria.hal.science/hal-03541297.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

[46] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

[47] Céline N. Martineau, Bora Baskaner, Renée I. Seinstra, William R. Schafer, André E. X. Brown, Ellen A. A. Nollen, and Patrick Laurent. Deep behavioural phenotyping reveals divergent trajectories of ageing and quantifies health state in c. elegans. *bioRxiv*, 2019. doi: 10.1101/555847. URL https://www.biorxiv.org/content/early/2019/02/20/555847.

[48] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[49] Hyeonchul Jung, Seokjun Kang, Takgen Kim, and HyeongKi Kim. Conftrack: Kalman filter-based multi-person tracking by utilizing confidence score of detection box. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6583–6592, 2024.

[50] Jingxian Liu, Shuhong Yang, and Fan Yang. A cross-and-dot-product neural network based filtering for maneuvering-target tracking. *Neural Computing and Applications*, 34(17):14929–14944, 2022.

[51] Miroslav Purkrabek and Jiri Matas. Probpose: A probabilistic approach to 2d human pose estimation. *arXiv preprint arXiv:2412.02254*, 2024.

[52] France Rose, Monika Michaluk, Timon Blindauer, Bogna M. Ignatowska-Jankowska, Liam O'Shaughnessy, Greg J. Stephens, Talmo D. Pereira, Marylka Y. Uusisaari, and Katarzyna Bozek. Deep imputation for skeleton data (disk) for behavioral science. *bioRxiv*, 2024. doi: 10.1101/2024.05.03.592173. URL https://www.biorxiv.org/content/early/2024/05/05/2024.05.03.592173.

[53] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL https://api.semanticscholar.org/CorpusID:216078090.