

Spline refinement with differentiable rendering

Frans Zdyb¹[0009–0005–6062–7790], Albert Alonso²[0000–0002–0441–0395], and Julius B. Kirkegaard^{1,2,3}[0000–0003–0799–3829]

¹ Department of Computer Science, University of Copenhagen, 2100, Denmark

² Niels Bohr Institute, University of Copenhagen, 2100, Denmark

³ Corresponding author: juki@di.ku.dk

Abstract. Detecting slender, overlapping structures remains a challenge in computational microscopy. While recent coordinate-based approaches improve detection, they often produce less accurate splines than pixel-based methods. We introduce a training-free differentiable rendering approach to spline refinement, achieving both high reliability and sub-pixel accuracy. Our method improves spline quality, enhances robustness to distribution shifts, and shrinks the gap between synthetic and real-world data. Being fully unsupervised, the method is a drop-in replacement for the popular active contour model for spline refinement. Evaluated on *C. elegans* nematodes, a popular model organism for drug discovery and biomedical research, we demonstrate that our approach combines the strengths of both coordinate- and pixel-based methods.

Keywords: Centerline Detection · Differentiable Rendering · Microscopy.

1 Introduction

Automated detection and tracking methods form the basis for studying physiological behavior, drug effects, and motility in model organisms, enabling quantification of assays in biomedical research [10]. Large animal models, such as mice and rats, are widely used for preclinical drug testing, allowing the quantification of complex physiological responses [2,3]. Zebrafish provide a versatile system for high-throughput screening, as their transparent larvae enable real-time observation of drug-induced behavioral changes [25,19]. Fruit flies with their rapid generation time, serve as powerful genetic models for studying neural function and the effects of drugs on behavior [30,24]. The detection and tracking of the behavior of these higher-order organisms are typically facilitated by landmark-based deep learning methods [21,18,20], which enable precise monitoring of limb movements, wing beats, and overall locomotion patterns. At the other scale, small model organisms such as the nematode *Caenorhabditis elegans* offer a simple and cost-effective model for studying neuroactive compounds and locomotion-based toxicity [23,11,26,15]. Due to their visually homogeneous appearance, landmark-based deep learning methods [18,20] are not ideal for high-throughput analysis of *C. elegans* assays. Instead, pixel-based segmentation methods can often estimate nematode poses extremely accurately [27,13,22,5]. However, these methods

struggle when the images are at low resolution and worms appear thin, or when worms are occluded either due to self-overlap or due to imaging at high densities. These challenges are overcome by spline-predicting methods [9,4,31], which can track worms during self-coiling and at extreme densities. These methods operate under a trade-off, however: Their accuracy is not pixel-level-accurate and are highly adversely affected when applied for out-of-distribution inference.

Here, we propose an unsupervised model that uses differentiable rendering [16] to refine spline predictions for pixel-level-accuracy. Our method optimizes a reconstructed microscopy image, adjusting the background, textures, colors, spline shapes, and widths via differentiable rendering. We compare this to the active contour model [14], an unsupervised refinement approach, which in contrast relies solely on feedback from the spline centerline, and therefore requires careful image preprocessing to work. This model works per-spline and therefore does not generalize well to overlapping splines. In contrast, in our approach, all splines are fitted simultaneously to an image.

We note that supervised deep learning refinement methods [17,29] are always faster and have greater detection accuracy. However, these rely on pixel-perfect labels, which often are unavailable. Such labels can be generated synthetically [4], where ground truth is fully controlled. In that case, the challenge is that synthetic data introduces a gap between simulation and reality, as model assumptions and textures differ from real-world conditions. Thus supervised refinement is possible for synthetic data, but bridging the reality-gap introduces the need for unsupervised refinement methods, such as the active contour model or the approach taken here.

2 Method

The input to our model is one or more (potentially overlapping) splines that need to be refined. These could, for instance, be the output of a deep learning detection model. Our refinement approach, illustrated in Fig. 1, reconstructs the input image through differentiable rendering by optimizing parameters to minimize the difference between the input and reconstructed images. Our primary concern are the parameters that define the splines, but the optimization is performed jointly with the background colors and texture of the image. Further, as our approach is fully differentiable, it is simple to regularize, ensuring e.g. smoothly curved spline (curvature regularization) or minor changes to total spline length.

We model the spline placement, spline fitting and rendering, along with background and texture generation, as an end-to-end differentiable program. A set of parameters define the location, shape, and width of each spline. Specifically, for each set of K knots $\{\mathbf{x}\}^K = \{(x_1, y_1, w_1), \dots, (x_K, y_K, w_K)\}$ we fit three natural cubic splines describing parametric curves of varying width,

$$\mathbf{r}^n : [0, 1] \rightarrow [0, \text{RES}]^2 \times [0, 1] = (r_x(s), r_y(s), r_w(s)), \quad (1)$$

where (r_x, r_y) are coordinates in a $\text{RES} \times \text{RES}$ image. r_w scales a global width parameter W which is shared for all splines. Additional parameters define background properties such as brightness, contrast and color gradient, and the texture

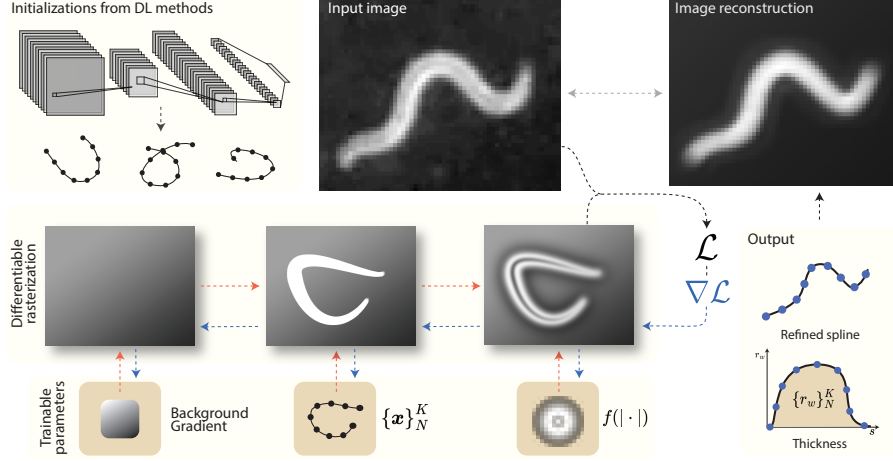


Fig. 1: Overview of the spline refinement method using differentiable rendering. The process starts with initial spline guesses, potentially from deep learning models. Through differentiable rasterization (orange arrows), an image is rendered from spline control points and, background and texture adjustments. The loss $\mathcal{L}$ is computed by comparing the rendered and input images, and gradients (blue arrows) update trainable parameters (bottom). At convergence, the refined spline (bottom right) accurately reconstructs the input shape (top right), as well as provides auxiliary data e.g. spline thickness.

of the image (Fig. 1). We have developed the method towards identifying splines whose texture is approximately constant along the spline, and we therefore learn only one (shared) blob texture.

2.1 Differentiable rendering

To render the reconstruction image, we sample a large set of interpolating (x, y, w) points uniformly along each spline. Each (x, y) point is rendered onto a separate image by differentiable resampling of a learned blob image, which is scaled by $W \cdot w$. The radially-symmetric blob image is produced by evaluating $f(z)$ in a small grid $\{(x_{ij}, y_{ij})\}$, where $z = \exp(-\sqrt{x^2 + y^2})$ and f is a learned 1D spline function. Summation over these scaled single-blob images produces a rendered spline. The images generated for each spline are combined using summation and max operations, with a parameter controlling the interpolation between the two. This allows for matching how overlapping splines interact visually. A background is generated and added, and the result is convolved with a learned 3×3 convolutional filter, which allows for brightness/contrast matching and minor blurring.

2.2 Optimization

Our loss function is the reconstruction error between the input image $\mathbf{Y}$ and the generated image $\hat{\mathbf{Y}}$,

$$\mathcal{L} = \|\mathbf{Y} - \hat{\mathbf{Y}}\|^2. \quad (2)$$

Akin to the active contour model [14], we regularize the optimization with terms that exploit prior information (barred parameters):

$$\mathcal{L}_{\text{reg}} = \lambda_1 \sum_i^N (\ell_i - \bar{\ell}_i)^2 + \lambda_2 \sum_i^N C_i + \lambda_3 \sum_n (\min_k w_k - \bar{w})^2 + \lambda_4 (W - \bar{W})^2. \quad (3)$$

where $C_i = \int \sqrt{(r''_x)^2 + (r''_y)^2} ds$.

Respectively, these terms regularize the spline length ℓ_i using the initial length $\bar{\ell}_i$ from the input knots, the curvature C_i (smoothness) of the spline, and the minimum ($\min w_k W$) and maximum (W) point width. The constants are per-dataset values.

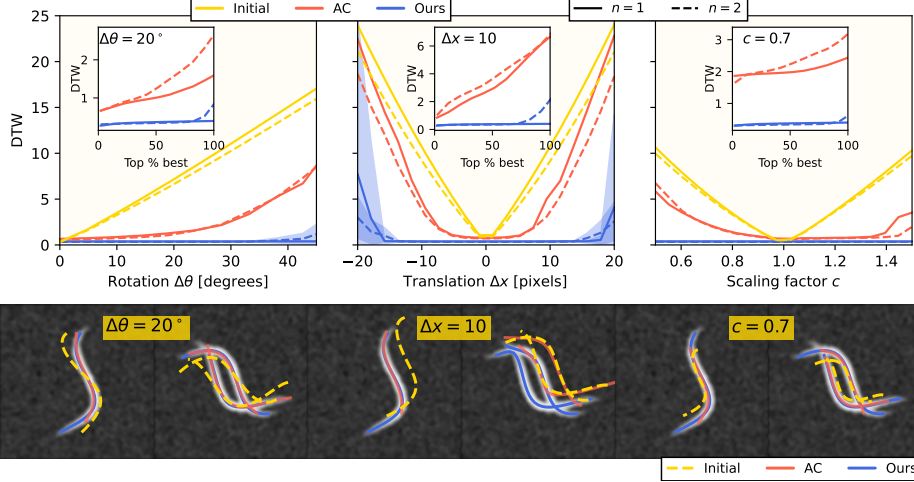


Fig. 2: Robustness analysis of the proposed method on SOSB (64 frames). Average DTW after transformations: rotation (left), translation (middle), and scaling (right). The initial guess (yellow) is refined using active contour (red) or our method (blue). Shaded areas indicate worsened performance. Main plots show the top 50% performance, with insets displaying percentile variations. Some reconstruction examples are shown below.

As our loss function is extremely non-convex, optimization is performed in three phases. First, we freeze the spline parameters, and optimize the parameters

of the background using Adam(1e-2). Next, we optimize all parameters except the convolutional filter and the blob parameters, using Adam(1e-2) with cosine annealing. Here we add Gaussian noise to the gradients, with a decaying variance, sampled with different random seeds in parallel, and select the parameters that achieve the lowest reconstruction loss. Lastly, we finetune all parameters with Adam(1e-4). The runtime is on the order of half a minute per batch of 3 images, and scales well with increasing batch size. While the choice of hyperparameters can be varied, the ordering of the optimization is crucial. For instance, spline locations must be optimized prior to the blob image, as otherwise the blob risks converging towards the background color.

3 Results

3.1 Evaluation Metrics

Centerline accuracy is often measured using mean squared error (MSE) [31], but it penalizes minor shifts that do not reflect significant spline differences. Pixel-based metrics, such as the Jaccard Index [28], quantify overlap but are sensitive to image discretization, making them unsuitable for subpixel structures [6,4]. To avoid such issues, we here use average Dynamic Time Warping (DTW) [1]. Before evaluation, both labels and refined splines are resampled to $N=100$ points using cubic interpolation.

3.2 *In-silico* Robustness Evaluation

Overlapping bodies dataset Access to high-quality labels for overlapping slender bodies is limited, obstructing quantitative evaluation [9,4,31]. Similarly, the majority of existing centerline prediction methods focus on single structures and struggle with overlaps [9,5]. Particularly for refinement studies, such as the present, labels need to be perfect to accurately reflect the improvements offered by refinement. For that reason, we introduce a dataset of synthetic overlapping slender bodies (SOSB) [8], resembling microscopy scenarios with overlapping slender bodies [4,22]. This enables controlled comparisons and systematic evaluation as it provides perfect labels.

Experiments Classic spline refinement methods, such as the active contour model [14], adjust centerline positions using pixel gradients. They struggle when the gradients are weak or ambiguous, especially in microscopy images with occlusions and noise [4]. Despite deep learning advancements, these methods remain popular due to their generality and ease of use, a trait shared by our proposed unsupervised differentiable approach. We benchmark against active contour (AC) and assess robustness under perturbations, here exemplified by affine transformations of the centerline label: translation, rotation, and scaling (Fig. 2). Similarly, we showcase the model’s ability to refine PCA-compressed labels, which represent the prediction space of some deep learning models [4,31]. The results shown

Number of Bodies n	Rotation $\Delta\theta = 20^\circ$			Displacement $\Delta x = 10$			Scaling $c = 0.7$			PCA $k = 2$		
	Initial	Ours	AC	Initial	Ours	AC	Initial	Ours	AC	Initial	Ours	AC
1	7.14	0.48	0.85	9.38	0.49	2.27	5.33	0.53	2.39	5.88	0.52	1.44
2	7.14	0.43	1.51	9.33	0.40	4.37	5.31	0.45	2.65	5.71	0.40	1.79
3	7.18	1.63	2.49	9.36	0.78	5.62	5.30	1.39	3.16	5.69	1.62	2.37

Table 1: Comparison of average DTW distance between original and transformed label (initial), and after refinement using either our method or active contour. Best-performing in **bold**. Values show the top 50% performance out of 256 frames for each n .

in Fig. 2 and in Table 1 demonstrate improvement over classical refinement techniques. For non-overlapping examples ($n = 1$), the method robustly recovers the labels even for extreme perturbations. For overlapping splines ($n \geq 2$), failure examples occur at extreme perturbation levels, but the method remains robust even far outside the range of perturbations expected for spline initializations.

3.3 Correcting *in-vivo* Distribution Shifts

Recent coordinate-based methods such as **de(ep)tangle** [4] and its newer extension **DeepTangleCrawl** (DTC) [31] have demonstrated well-performant spline detection and tracking of *C. elegans* in challenging scenarios, such as during overlapping and occlusions, and in noisy environments. Despite their capabilities, these methods struggle in out-of-distribution situations where e.g. nematodes exhibit complex morphologies. For example, tightly coiled or heavily occluded worms are difficult to label and are often underrepresented in training sets. Such failures affect the robustness of the methods and can discourage their application in quantitative studies [7].

Here, we address such distribution shifts. For DTC, distribution shifts arise because it relies on **Tierspy** [12] to generate most of the labels, which performs well in most situations, but struggles when the boundaries of the nematodes are not well-defined. Fig. 3A demonstrates how our method corrects predictions for coiled worms, where initial predictions underperform, but refinement significantly improves accuracy. In contrast, distribution shifts in **de(ep)tangle** stem from training fully *in-silico*. This leaves a synthetic-to-real gap between the physics-based simulations and frame rendering and real worm recordings, which results in stiffer (straighter) spline predictions than those observed [4]. Fig. 3B shows these examples and how our method can help resolve the issue even at this high density. Note that while only one prediction is shown, we refine all visible splines simultaneously — this is what allows refinements despite overlap. These demonstrate the potential of our approach to not only refine predictions for coordinate-based deep learning models but also provide more accurate labels for future dataset curation.

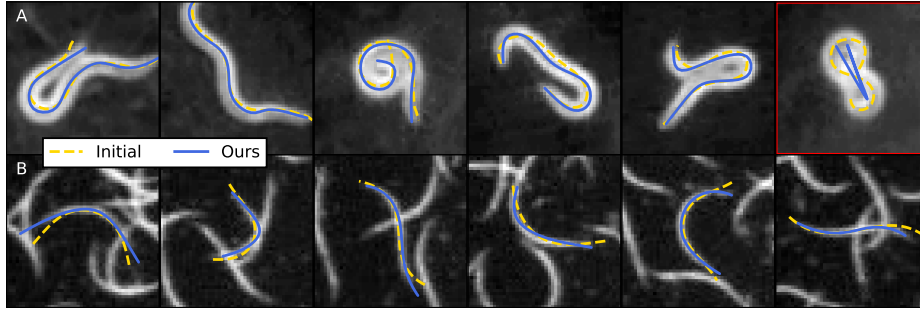


Fig. 3: Examples of refinement (blue) of predictions (yellow) from (A) DeepTangleCrawl [31] and from (B) de(ep)tangle [4] on low-density *C. elegans* crawling experiments and high density swimming experiments, respectively. Example failure case is framed red.

3.4 Assisted Differentiable Labeling

Labeling data for training supervised detection models is cumbersome and labor-intensive, particularly so for slender objects at high densities. As our method can converge from extreme perturbations, it can enable an assisted labeling workflow in which quick first-pass labels, such as straight lines, are refined to full labels. We show examples of this in Fig. 4, and compare to DTC labels in Table 2, demonstrating the potential of this idea. While we mainly care about the splines, our method further gives auxiliary output. For instance, the reconstruction loss can be used to assess whether a refinement was successful. Likewise, our method outputs spline widths (Fig. 4B), which potentially could produce labels for pixel-based segmentation. Similarly, being able to recreate the body of each spline independently can ease the process of edge detection even in situations where the bodies are obfuscated.

Percentile	5 %			10 %			20 %			50 %		
	Initial	Ours	AC	Initial	Ours	AC	Initial	Ours	AC	Initial	Ours	AC
	2.06	0.28	1.02	2.32	0.30	1.12	2.89	0.34	1.43	3.98	0.42	1.84

Table 2: Refining straight lines on DTC dataset, DTW percentiles. $n = 1,000$.

4 Perspective

In this paper, we have presented a method that refines centerline predictions to achieve pixel-level accuracy without requiring training. Our approach relies on global reconstruction of the input image, leading to improved convergence

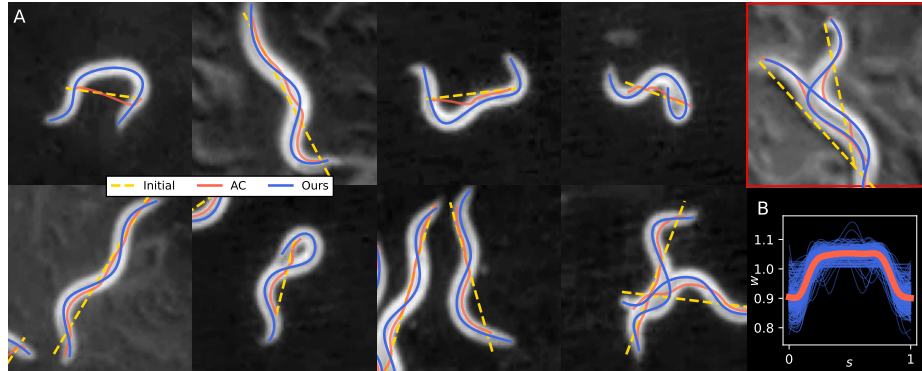


Fig. 4: (A) Refining low-information labels (straight lines). Example framed in red shows a failure case. (B) Extracted scale parameter as a function of arc length, average in red.

compared to local centerline methods such as active contours. More broadly, our results demonstrate the advantages of differentiable rendering-based matching, even when the reconstructed output only approximately resembles the input image. Despite these benefits, our method has clear limitations. For instance, failure cases highlight its sensitivity to the choice of regularization hyperparameters. Concretely, for the used *C. elegans* dataset, the method is generally successful in correcting both overlapping and coiling configurations, but may still fail due to regularization parameters being incorrectly tuned. Parallel worms that appear as a single thick spline remain challenging. Future work in the direction of 2D-splines with a temporal dimension could help resolve this. Our approach is quite data-agnostic, but can be extended in a more data-specific direction, for instance by including prior knowledge relating to nematodes and the imaging process. In particular, reformulating the differentiable program as a probabilistic program would allow for multi-modal priors, covering a much wider range of worm phenotypes without relying on tuning regularization hyperparameters. In addition, this could allow for quantifying uncertainty in the refinements. Finally, refinement can enable self-supervised learning: Any supervised model used to provide the initial spline can be fine-tuned on refined output, yielding better initial splines on future images. Such a training loop, potentially guided by the image reconstruction loss, could be used to train deep supervised models with initially few but diverse training labels.

Code availability. All code is available at <https://github.com/kirkegaardlab/splender>

Acknowledgments. We thank André EX Brown, Javier Rodriguez and Weheliye H Weheliye for providing us with the *C. elegans* crawling data for the qualita-

tive analysis. This work has received funding from the Novo Nordisk Foundation, Grant Agreement No. NNF20OC0062047.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Dynamic Time Warping. In: Müller, M. (ed.) Information Retrieval for Music and Motion, pp. 69–84. Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74048-3_4
2. Alexandrov, V., Brunner, D., Hanania, T., Leahy, E.: High-throughput analysis of behavior for drug discovery. *European journal of pharmacology* **750**, 82–89 (2015)
3. Aljovic, A., Zhao, S., Chahin, M., de la Rosa, C., Van Steenberghe, V., Kerschensteiner, M., Bareyre, F.M.: A deep learning-based toolbox for automated limb motion analysis (alma) in murine models of neurological disorders. *Communications Biology* **5**(1), 131 (2022)
4. Alonso, A., Kirkegaard, J.B.: Fast detection of slender bodies in high density microscopy data. *Communications Biology* **6**(1), 1–12 (Jul 2023). <https://doi.org/10.1038/s42003-023-05098-1>
5. Cutler, K.J., Stringer, C., Lo, T.W., Rappez, L., Stroustrup, N., Brook Peterson, S., Wiggins, P.A., Mougous, J.D.: Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature methods* **19**(11), 1438–1448 (2022)
6. Cutler, K.J., Stringer, C., Lo, T.W., Rappez, L., Stroustrup, N., Brook Peterson, S., Wiggins, P.A., Mougous, J.D.: Omnipose: A high-precision morphology-independent solution for bacterial cell segmentation. *Nature Methods* **19**(11), 1438–1448 (Nov 2022). <https://doi.org/10.1038/s41592-022-01639-4>
7. Deserno, M., Bozek, K.: Unsupervised representation learning of c. elegans poses and behavior sequences from microscope video recordings. *bioRxiv* pp. 2025–02 (2025)
8. The dataset is available at <https://zenodo.org/records/15519588> and contains images and labels for $n = 1, 2, 3$ overlapping slender bodies and each dataset contains 256 independent frames of 128×128 size.
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 computational biology* **17**(4), e1008914 (2021)
10. Hoffman, K.L.: Artificial intelligence pushes the boundaries of behavioral analysis in drug discovery: a revolution from the deep. *Expert Opinion on Drug Discovery* **19**(1), 1–3 (2024)
11. Ikenaka, K., Tsukada, Y., Giles, A.C., Arai, T., Nakadera, Y., Nakano, S., Kawai, K., Mochizuki, H., Katsuno, M., Sobue, G., et al.: A behavior-based drug screening system using a caenorhabditis elegans model of motor neuron disease. *Scientific reports* **9**(1), 10104 (2019)
12. Javer, A., Currie, M., Lee, C.W., Hokanson, J., Li, K., Martineau, C.N., Yemini, E., Grundy, L.J., Li, C., Ch’ng, Q., Schafer, W.R., Nollen, E.A.A., Kerr, R., Brown, A.E.X.: An open-source platform for analyzing and sharing worm-behavior data. *Nature Methods* **15**(9), 645–646 (Sep 2018). <https://doi.org/10.1038/s41592-018-0112-1>

13. Javer, A., Currie, M., Lee, C.W., Hokanson, J., Li, K., Martineau, C.N., Yemini, E., Grundy, L.J., Li, C., Ch'ng, Q., et al.: An open-source platform for analyzing and sharing worm-behavior data. *Nature methods* **15**(9), 645–646 (2018)
14. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* **1**(4), 321–331 (Jan 1988). <https://doi.org/10.1007/BF00133570>
15. Kong, F., Jin, H., Xu, Y., Shen, J.: Behavioral toxicological tracking analysis of drosophila larvae exposed to polystyrene microplastics based on machine learning. *Journal of Environmental Management* **359**, 120975 (2024)
16. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. *Science* **350**(6266), 1332–1338 (2015)
17. Laube, P., Franz, M.O., Umlauf, G.: Deep learning parametrization for b-spline curve approximation. In: 2018 International Conference on 3D Vision (3DV). pp. 691–699. IEEE (2018)
18. Lauer, J., Zhou, M., Ye, S., Menegas, W., Schneider, S., Nath, T., Rahman, M.M., Di Santo, V., Soberanes, D., Feng, G., et al.: Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods* **19**(4), 496–504 (2022)
19. Miyawaki, I.: Application of zebrafish to safety evaluation in drug discovery. *Journal of toxicologic pathology* **33**(4), 197–210 (2020)
20. Pereira, T.D., Tabris, N., Matsliah, A., Turner, D.M., Li, J., Ravindranath, S., Papadopyannis, E.S., Normand, E., Deutsch, D.S., Wang, Z.Y., et al.: Sleep: A deep learning system for multi-animal pose tracking. *Nature methods* **19**(4), 486–495 (2022)
21. Pérez-Escudero, A., Vicente-Page, J., Hinz, R.C., Arganda, S., De Polavieja, G.G.: idtracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods* **11**(7), 743–748 (2014)
22. Perni, M., Challa, P.K., Kirkegaard, J.B., Limbocker, R., Koopman, M., Hardenberg, M.C., Sormanni, P., Müller, T., Saar, K.L., Roode, L.W., et al.: Massively parallel *c. elegans* tracking provides multi-dimensional fingerprints for phenotypic discovery. *Journal of neuroscience methods* **306**, 57–67 (2018)
23. Perni, M., Galvagnion, C., Maltsev, A., Meisl, G., Müller, M.B., Challa, P.K., Kirkegaard, J.B., Flagmeier, P., Cohen, S.I., Cascella, R., et al.: A natural product inhibits the initiation of α -synuclein aggregation and suppresses its toxicity. *Proceedings of the National Academy of Sciences* **114**(6), E1009–E1017 (2017)
24. Qu, S., Zhu, Q., Zhou, H., Gao, Y., Wei, Y., Ma, Y., Wang, Z., Sun, X., Zhang, L., Yang, Q., et al.: Easyflytracker: a simple video tracking python package for analyzing adult drosophila locomotor and sleep activity to facilitate revealing the effect of psychiatric drugs. *Frontiers in Behavioral Neuroscience* **15**, 809665 (2022)
25. Rihel, J., Schier, A.F.: Behavioral screening for neuroactive drugs in zebrafish. *Developmental neurobiology* **72**(3), 373–385 (2012)
26. Sohrabi, S., Mor, D.E., Kaletsky, R., Keyes, W., Murphy, C.T.: High-throughput behavioral screen in *c. elegans* reveals parkinson's disease drug candidates. *Communications Biology* **4**(1), 203 (2021)
27. Swierczek, N.A., Giles, A.C., Rankin, C.H., Kerr, R.A.: High-throughput behavioral analysis in *c. elegans*. *Nature methods* **8**(7), 592–598 (2011)
28. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging* **15**(1), 29 (Aug 2015). <https://doi.org/10.1186/s12880-015-0068-x>
29. Wang, H., Song, Q., Yin, R., Ma, R.: B-spine: Learning b-spline curve representation for robust and interpretable spinal curvature estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5381–5389 (2024)

30. Wang, Y.Y., Ma, W.W., Peng, I.F.: Screening of sleep assisting drug candidates with a drosophila model. *PloS one* **15**(7), e0236318 (2020)
31. Weheliye, W.H., Rodriguez, J., Feriani, L., Javer, A., Uhlmann, V., Brown, A.E.: An improved neural network model enables worm tracking in challenging conditions and increases signal-to-noise ratio in phenotypic screens. *bioRxiv* pp. 2024–12 (2024)