Annotation-free whole-brain neuron tracking via transformer-based self-supervised learning

Hang Lu

hang.lu@gatech.edu

Georgia Institute of Technology https://orcid.org/0000-0002-6881-660X

Haejun Han

Georgia Institute of Technology

Robert Pritchard

Georgia Institute of Technology

Article

Keywords:

Posted Date: January 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8080516/v1

Additional Declarations: There is **NO** Competing Interest.

Annotation-free whole-brain neuron tracking via transformer-based self-supervised learning

Haejun Han[1], Robert Pritchard[1], Hang Lu[1,2*]

[1]Interdisciplinary Bioengineering Graduate Program, Georgia Institute of Technology, Atlanta, Georgia, USA.

[2]School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA.

*Corresponding author(s). E-mail(s): hang.lu@gatech.edu;

Abstract

Tracking individual neurons in dense 3D microscopy recordings of fast-moving, deforming organisms is a critically important in neuroscience. This task is often hindered by complex motion and the need for laborious manual annotation. Here, we introduce Annotation-free Self-supervised Contrastive Embeddings for Neuron Tracking (ASCENT), a computational framework that learns robust representations of neuronal identity that are invariant to tissue deformation. ASCENT trains a Neuron Embedding Transformer (NETr) through a self-supervised contrastive scheme on augmented data that mimics animal movement and tissue deformations. NETr generates highly discriminative embeddings for each neuron by combining visual features with positional information, refined by the context of all other neurons in the frame. On a benchmark freely-moving *C. elegans* dataset, ASCENT achieves best tracking accuracy yet, surpassing even supervised methods. We further demonstrate ASCENT's robust performance across several additional datasets in different anatomical regions and under different imaging conditions. We show that the method is generally robust to image noises, is highly data-efficient, and generalizes across different imaging conditions. By eliminating the annotation bottleneck, ASCENT provides a highly scalable and practical solution for the analysis of whole-brain neural dynamics in behaving animals.

Introduction

Understanding how the activity of large neural populations gives rise to complex behaviors is one of the central goals in neuroscience. Functional imaging, particularly in behaving animals, provides a way to directly associate the dynamics of neural activities and circuits to relevant behaviors. Advances in volumetric microscopy, such as light-sheet or light field microscopy and light-sculpting techniques, now allow one to record three-dimensional (3D) time-lapse images of nearly entire brains in small model organisms such as the nematode *Caenorhabditis elegans* [1–3], the larval zebrafish *Danio rerio* [4–7], and the fruit fly *Drosophila melanogaster* [8–10] at single-cell resolution. Extracting biological insight from these rich datasets, however, first requires solving the essential task of multi-neuron tracking: accurately following the position and identity of each neuron through time to extract its individual activity traces [11–14].

This tracking problem is exceptionally challenging in the context of 3D fluorescence microscopy. Neurons are often densely packed and, when viewed in isolation, lack uniquely

distinguishing image features such as texture or shape, making them appear visually similar [12, 13]. They are also subject to complex, non-rigid deformations as the animal moves and its brain bends, twists, and stretches [15–17]. To address this, early computational methods relied on registering point clouds representing neuron coordinates between frames [12, 18] or aligning them to a reference atlas [19–21]. Subsequent deep learning techniques often focused primarily on positional features [22, 23], sometimes overlooking valuable image content from the local cellular neighborhood that can provide context for disambiguation. More recent methods incorporating image features have improved accuracy by using techniques such as graph matching [24], voxel classification [25], or local and global image registration [16, 26]. However, despite these advances, many of these state-of-the-art methods still do not yield high enough accuracy to avoid extensive manual corrections for tracking, in part because of noises in fluorescence images, imperfections in object detection, and large motions. Additionally, creating the ground-truth data for supervised learning (i.e. manually identifying and linking hundreds of neurons across thousands of volumes) is an extremely labor-intensive and expertise-dependent process that limits the scalability and widespread adoption of these powerful whole-brain imaging methods [27–29].

Here, we present Annotation-free Self-supervised Contrastive Embeddings for Neuron Tracking (ASCENT), a computational framework designed to overcome these challenges by learning embeddings directly from unlabeled imaging data. We designed a Transformer-based network that integrates local image features along with positional information to track dense, visually similar cells. This network uses self-attention to refine each embedding with contextual cues from all co-detected neurons, which provides robustness against the complex, non-rigid deformations inherent to imaging behaving animals. Importantly, we train this network using a self-supervised contrastive learning scheme with a tailored data augmentation strategy, which completely eliminates the need for manual track annotations. We demonstrate that the synergistic use of both image and positional features is crucial for robust performance, particularly in noisy conditions, which is a common feature in 3D video microscopy for moving/deforming samples. We demonstrate the power of ASCENT on four highly heterogeneous datasets, including publicly available benchmark datasets and new recordings, covering both head and tail ganglia of *C. elegans* under varying conditions. We show that ASCENT surpasses state-of-the-art accuracy and generalizes effectively across different imaging conditions. Furthermore, ASCENT is highly data-efficient, achieving high degree of performance with as few as eight unlabeled training volumes.

Results

The ASCENT framework for annotation-free neuron tracking

Tracking individual neurons in dense, deforming 3D volumes is a great challenge in fluorescence video microscopy, particularly in freely moving animals. Neurons often appear visually similar, and the sample is subject to large, complex movements that include non-rigid deformations, where different parts of the tissue can shift and warp independently (Figure 1A,B, Supplementary Video 1,2). In addition, detection errors (e.g. missed neurons or positional noise) contribute to performance degradations of tracking algorithms. To overcome this challenge, we developed ASCENT (Figure 1C-E), a self-supervised contrastive learning framework for tracking. The core principle of ASCENT is to train a deep embedding network that generates a unique and discriminative feature vector for each neuron at every timepoint. This transforms the complex task of spatiotemporal linking into a more straightforward feature-matching problem between different frames.

The framework is a modular three-stage process (Figure 1C). In the first stage, initial neuron coordinates are provided by an off-the-shelf detection algorithm (see Methods for

details). This design choice makes ASCENT agnostic to the specific detection method and allows it to readily incorporate future advances in object detection or segmentation. In the second and most critical stage, our Neuron Embedding Transformer (NETr) computes a robust, context-aware representation for each detected neuron. Finally, neurons are linked across frames using a simple bipartite matching algorithm. This straightforward approach is made possible by the highly discriminative nature of the learned embeddings and avoids the complex post-processing and case-by-case parameter tuning required by many other tracking methods.

Specifically, the NETr architecture is designed to produce context-aware embeddings by combining local visual appearance and the relative position of neurons (Figure 1D, Supplementary Figure 1A; see Methods for details). While individual neurons often appear visually similar, we hypothesized that the local cellular neighborhood provides a rich, identity-preserving visual signature. By extracting features from a 3D image patch around each neuron, a powerful visual backbone can learn these unique contextual patterns. However, visual features alone are not sufficient for robust tracking under all conditions; they are inherently sensitive to image noise and can fail when spatially distant neurons have visually indistinguishable surroundings. As a complement to image features, NETr separately computes positional features from the coordinates of detected neurons. This provides a global spatial frame of reference that disambiguates visually confusing cases. These two feature modalities are then combined and processed by a Transformer encoder. The choice of this architecture was inspired by the success of its self-attention mechanism in the feature-matching literature [30–33]. This mechanism refines each neuron's embedding by integrating information from all other neurons present in the same frame.

To overcome the manual annotation bottleneck inherent to supervised neuron tracking, we adopted a self-supervised contrastive learning scheme (Figure 1E, Supplementary Figure 1B; see Methods for details) based on the SimCLR framework [34]. It draws on the success of contrastive methods in other contexts [35–39]. The network learns by being shown two different, randomly augmented "views" of the same set of neurons from a single frame. A contrastive loss function then trains the network to produce consistent embeddings for the same neuron across the two augmented views, while simultaneously pushing apart the embeddings of different neurons. By doing so, NETr learns to generate feature embeddings that are robust to deformations and imaging variability, yet highly specific to individual neuron identity.

An experimental condition-aware data augmentation strategy creates a discriminative embedding space

To learn features that are invariant to imaging noise and sample deformation yet specific to each neuron, we implemented a data augmentation strategy that creates diverse views of the same neurons while preserving their identity (Figure 2A, Supplementary Figure 2). For instance, transformations such as rotation and elastic deformation are meant to mimic moving samples during imaging, while position jitter and color (intensity) jitter mimic perturbations due to the nature of the reagents and noise from the microscopy system.

To understand how this process enables learning, we visualized the neuron embedding space before and after training for a whole-brain recording of a freely-moving *C. elegans* from a previous study [3] using uniform manifold approximation and projection (UMAP) [40] (see Methods for details). Before training, the embedding space is largely unstructured. While the pre-trained visual backbone provides some initial feature separation, the embeddings corresponding to most neuron identities are intermingled and indistinct (Supplementary Figure 3, left). Consequently, the representations of the same neuron from an original and an augmented view are scattered randomly across the space (Figure 2B, left). This demonstrates

that the untrained network lacks the capacity to distinguish neurons, and thus will not allow reliable feature-based tracking. After training with the contrastive objective, the embedding space becomes highly organized (Figure 2B, right; Supplementary Figure 3, right): embeddings of the same neuron, including those from augmented views, converge into tight, identity-specific clusters that are well-separated across the dataset. The emergence of this structure is a direct visualization of the learning process; it demonstrates that the network has successfully learned to generate a unique and robust feature signature for each neuron, creating a space where simple distance-based matching can be used for reliable tracking.

To determine the contribution of each transformation to this outcome, we systematically dissected the augmentation pipeline using the same dataset. We first trained the embedding network using only one or a combination of two transformations from our suite of six transformations to assess their individual and pairwise contributions (Figure 2C). While any single augmentation dramatically improved performance over the no-transformation baseline (44.5% accuracy), we found that applying random jitter to the neuron positions provided the single most effective gain, boosting tracking accuracy to 95.1%. This result is particularly significant; it demonstrates that the most critical factor in learning a useful embedding space is to force the network to be robust to small positional perturbations, which are inevitable in real data (variations from worm to worm, from frame to frame) and realistic detection pipelines producing non-negligible segmentation/detection errors. Furthermore, a combination of just two transformations – position jitter and random rotation – was sufficient to achieve the peak accuracy of 95.7%, matching the performance of the full suite of six augmentations.

To assess the relative importance and potential redundancy of each augmentation, we also performed the converse experiment by systematically removing one or two transformations from the full suite (Figure 2D). As expected, removing position jitter alone caused the most substantial performance drop, reducing accuracy from 95.7% to 81.4% (Figure 2D). In contrast, the removal of other augmentations (e.g. elastic deformation or color jitter) from the full set resulted in negligible changes to accuracy. This suggests a degree of functional overlap of these augmentations with each other (Figure 2D). This apparent redundancy likely arises from the inherent frame-to-frame variability in this rather challenging dataset that provides a learning signal similar to that simulated by these specific transformations. It is worth noting that with other datasets, the contributions of these features may be quantitatively different and less redundant. Since no combination of augmentations meaningfully hampered performance, our results suggest that while a minimal set of augmentations is highly effective for this dataset, the full suite could provide a robust default strategy that may confer additional benefits on more complex or varied datasets. Here, by transforming raw image data and neuron positions into a rich set of augmented views, the self-supervised training strategy successfully guides the network to learn an embedding space structured by neuron identity, which is required for accurate tracking.

Synergistic integration of image and positional features ensures tracking robustness

A common assumption in tracking visually similar cells is that local appearance is insufficient for reliable identification, which is why most methods primarily rely on positional information [12, 22, 23]. Our approach challenges this position-centric view. While individual neurons may appear very similar morphologically, the 3D spatial arrangement of their neighbors within a local patch creates a unique geometric signature. We reasoned that even under non-rigid deformation, this local constellation of neurons provides a robust, identity-preserving feature that a powerful visual backbone can learn. However, visual context alone

can be ambiguous when heavy image noises are present or when spatially distinct neurons share similar local neighborhoods. Thus, we designed the architecture of ASCENT to include both image features and positional features. To dissect contributions of image and positional features to ASCENT's performance, we evaluated ablated models that used only image features ('Image-only') or only positional embeddings ('Positional-only') against the complete 'Standard' NETr model.

Image features are highly susceptible to image noises, which are common and microscope-dependent. We thus, simulated various noise levels by adding zero-mean pixel-wise Gaussian noise with standard deviation σ applied to the normalized whole-brain recording of the freely moving *C. elegans* (Figure 3A,B; see Methods). On the original, low-noise imaging data, the standard model achieved the highest accuracy. Surprisingly, the model using exclusively local image features performed nearly as well as the complete model. This demonstrates that our self-supervised learning strategy can extract powerfully discriminative information from local appearance alone (Figure 3B, orange dots). However, this reliance on image information is fragile; as noise levels increased, the accuracy of the image-only model degraded severely. In contrast, the model using only positional information proved highly robust to this image degradation, maintaining its performance as image quality decreased (Figure 3B, blue dots). These results highlight the complementary nature of the two feature modalities. While image information provides highly discriminative features under ideal conditions, positional information provides an essential scaffold of robustness against image noise. Therefore, the integration of both is necessary to achieve consistently high tracking accuracy across the variable conditions encountered in live-animal microscopy.

Tracking errors visualized in Figure 3C further demonstrate the complementary nature of these components and their distinct failure modes: the image-only model fails by swapping two visually similar but spatially separated neurons (#34 and #36) that are indistinguishable within their local image patch. In contrast, the position-only model fails by swapping two spatially proximal but visually distinct neurons (#4 and #61) whose relative positions become ambiguous in the presence of tissue deformation. This latter failure mode is a key limitation of trackers that largely ignore image content. By integrating both modalities, ASCENT's full architecture overcomes the specific limitations of each, using positional encoding to disambiguate visually similar neurons and local appearance to resolve identities when relative positions are distorted.

ASCENT sets a new state-of-the-art performance and generalizes across diverse datasets

We validated ASCENT's performance on two datasets with different characteristics: a publicly available recording of a freely moving *C. elegans* (NeRVE) and an in-house recording of a worm confined in a microfluidic channel subjected to odorant stimuli (Figure 4A). Supplementary Videos 1 and 2 show the final tracking results, with neuron identities colored and maintained over time, for the in-house and NeRVE datasets, respectively. We first benchmarked ASCENT against state-of-the-art methods on the NeRVE dataset (Supplementary Table 1, Figure 4B). ASCENT achieves a tracking accuracy of (95.90%), surpassing the previous best-performing supervised method, ZephIR (94.48%) [16] (Figure 4B). ZephIR shows a clear trade-off between accuracy and annotation labor (Figure 4B). Very importantly, we note that ASCENT's superior performance is achieved with zero manual annotation time, in contrast to the hours of expert annotation *per video* analyzed for supervised methods such as ZephIR [16] and CeNDeR [23]. This result demonstrates that ASCENT can circumvent the traditional trade-offs between accuracy and annotation effort. Furthermore, ASCENT substantially outperforms other annotation-free methods such as the original NeRVE tracker

(82.9%) [12] and fDNC (79.1%) [22], highlighting the power of ASCENT's synergistic feature integration and contextual refinement.

A key to robustness of computational method is its ability to generalize across different datasets and imaging conditions. We assessed this in ASCENT by training a model on in-house dataset and testing it on NeRVE and vice versa. Interestingly, the results show an asymmetric generalization performance (Figure 4C). A model trained on the more challenging, diverse NeRVE dataset generalized effectively to the in-house data (with an accuracy of 98.81%), which is nearly identical to the 98.87% achieved by a model trained specifically on the in-house data. In contrast, applying the model trained on the simpler in-house data on constrained worm to the NeRVE dataset resulted in a performance drop from 95.90% for the NeRVE-specific model to 89.79%. Notably however, this reduced accuracy, achieved without any NeRVE-specific training, still substantially outperforms other annotation-free methods such as the original NeRVE tracker [12] and fDNC [22]. Furthermore, it even surpasses supervised methods like the CeNDeR model (trained with 130 labeled frames) [23] and the baseline ZephIR model (trained with 10 labeled frames) [16]. It was only outperformed by the ZephIR model trained with 10 labeled frames and 10 labeled neurons across all 1,536 frames, which required extensive manual work (estimated 536 minutes [12, 16]). ASCENT's generalization performance shown here indicates that it learns fundamental, transferable features of neuron appearance and spatial configuration.

We next examine how the high-fidelity tracks produced by ASCENT can enable reliable downstream analysis. Using the tracks from the in-house dataset, we extracted neural activity traces that demonstrate odor-entrained responses as well as spontaneous activities in the animal (Figure 4D; see Methods). To evaluate the fidelity of these tracks and identify sources of error, we quantified the rates of missing and mismatched neurons for each frame. We found that both missing and mismatch errors are sparse, with most frames having zero or one error in each category. To isolate the source of error, whether from the core embedding and linking module, or from the upstream detection task, we evaluated ASCENT in a 'track-only' mode, where it processed ground-truth neuron coordinates directly (see Methods for details) (Supplementary Table 1). The near-perfect accuracy in this mode (99.99%) compared to the standard mode (98.87%) on the in-house data confirms that the vast majority of errors in the end-to-end pipeline are attributable to the initial detection step and not the embedding or the tracking step. A qualitative analysis of the most error-prone frames supports this conclusion: errors often arise from segmentation failures, such as under-segmentation of neurons or ambiguities between ground-truth annotations and segmentation masks, rather than from the core embedding-and-linking methodology of ASCENT (Supplementary Figure 4). Improving the detection accuracy would further improve the end-to-end performance.

Data efficiency enables rapid training on minimal data

Beyond its accuracy, a major practical advantage of ASCENT is its data efficiency. This is important for applications such as rapidly training models for new experimental conditions, or building large, foundational models from diverse recordings. To quantify this feature, we examined the impact of training set composition on performance using the NeRVE dataset (Figure 5). A visualization of the image space across the entire recording reveals a circular and continuous manifold, suggesting considerable informational redundancy between temporally adjacent frames (Figure 5A; see Methods for visualization details), perhaps a result of the freely-moving animal's undulatory locomotion. To test this, we created training sets of increasing size (1, 2, 4, ..., 64 frames) by sampling frames at regular intervals across the recording. For comparison, we created an additional training set using only the first 8 consecutive frames (Figure 5B). We found that ASCENT achieves over 95% accuracy using as few as 8 regularly sampled, unlabeled frames, reaching near-peak performance

with only a fraction of the full 1,532-frames dataset (Figure 5C). Notably, a model trained on the first 8 consecutive frames, which cover a less diverse range of postures, still achieved a high accuracy of 95.42%, only slightly lower than the 95.66% from 8 regularly sampled frames, suggesting that our data augmentation strategy can largely compensate for a lack of diversity in the training frames. This efficiency extends to training time, with convergence to plateau accuracy in under 15 minutes with 1 H200 GPU for all tested subset sizes. This performance underscores the feasibility of quickly generating dataset-specific models without a significant time investment (Figure 5D). Although this data efficiency is advantageous, the self-supervised nature of ASCENT means that using all available frames to maximize exposure to diverse postures is always an option (provided that there is sufficient computing power), as it incurs no additional manual labeling cost.

Joint model generalization across imaging regimes and enables rapid adaptation

To test ASCENT's ability to generalize to new experiments, we next applied it to diverse datasets and examined its capacity for rapid fine-tuning to novel contexts (Figure 6). First, we tested generalization by training models on three distinct *C. elegans* recordings of the head ganglia: device-restricted worms (Dataset A; in-house dataset), freely moving worms (Dataset B; NeRVE dataset [12]), and body-axis–aligned freely moving worms (Dataset C; Flavell lab dataset [41]) (Figure 6A). We compared dataset-specific models against a single joint model trained on all three (A+B+C) (Figure 6B; see Methods for training details). The joint model's performance was statistically indistinguishable from or superior to the dataset-specific models, with high tracking accuracies on Dataset B (97.94% vs. 97.86%, p=0.09) and C (98.41% vs. 98.34%, p=0.29). While the dataset-specific model was technically higher on Dataset A (99.99% vs 99.98%, p = 0.03), this difference is negligible as both achieved near-perfect tracking. These results demonstrate that ASCENT can be trained jointly on heterogeneous datasets to produce a single, robust model that generalizes well across all constituent imaging conditions.

We next evaluated if this joint model could serve as a pre-trained basis for rapid adaptation to a new biological context, the *C. elegans* male tail (Datasets D) (Figure 6D). Males when mating exhibit highly complex locomotion, producing among the most challenging tracking tasks. We fine-tuned the A+B+C joint model on four independent recordings (D_1–D_4) and compared its performance to models trained from scratch on the same data (see Methods for training details). While differences on D_1 (p = 0.151) and D_2 (p = 0.820) were not significant, fine-tuned models significantly outperformed from-scratch models on D_3 (p = 0.0011) and D_4 (p = 0.0110) (Figure 6E). Early-epoch analysis further confirmed that fine-tuning achieves high accuracy rapidly, reaching benchmark fractions of each model's peak accuracy (for example, 98% and 99%) in less training time than models trained from scratch (Supplementary Figure 5). These results suggest that ASCENT can learn transferable representations from diverse data, possibly creating a generalist "foundation" model. This model can then be rapidly fine-tuned to new imaging contexts. Our test case here points to two advantages of this approach: fine-tuning is not only efficient, but also often results in better performance because the "foundation" model anticipates the data diversity that may not be sampled by the data-specific training.

Discussion

In this work, we present ASCENT, a self-supervised framework that sets a new standard for annotation-free 3D neuron tracking. By training the Neuron Embedding Transformer, NETr, with a contrastive learning objective, ASCENT generates highly discriminative embeddings

directly from unlabeled volumetric imaging data. This eliminates the most significant bottleneck in conventional tracking pipelines: the need for laborious manual track annotation. Our results demonstrate that the accuracy of this approach exceeds that of state-of-the-art methods that rely on supervised training. A key insight from our architectural analysis is the effectiveness of learned local visual features. Contrary to the long-held assumption that the visual similarity of neurons necessitates a primary reliance on positional information, our results show that a model using only visual features can achieve remarkable accuracy on standard benchmark data, as long as noise is not too severe. This underscores the power of combining a robust pretrained visual backbone with a self-supervised objective tailored for instance discrimination. However, our experiments also confirm that this approach is not perfect; incorporating explicit positional encoding is important for providing robustness against image noise often encountered in fluorescence imaging. Therefore, the integration of both visual and positional information within the NETr architecture offers the most reliable and broadly generalizable solution.

Overall, the principles demonstrated in ASCENT open several promising avenues for advancing the analysis of neural dynamics. The high data efficiency and strong generalization capabilities position ASCENT as a candidate for the development of a 'foundation model' for neuron tracking. We showed that such a model, pretrained on diverse data, can be rapidly fine-tuned to achieve superior tracking performance on entirely new anatomical contexts, often exceeding models trained from scratch. This approach drastically reduces the annotation and training burden for new experiments, greatly increasing accessibility and standardization across the field. By removing the barrier of manual annotation and demonstrating high efficiency in both model training and inference, ASCENT makes high-performance 3D neuron tracking more accessible. We envision that with ASCENT, the quantitative analysis of whole-brain data will be more scalable, robust, and accessible, for larger and more complex problems in neuroscience.

While optimizing neuron segmentation and detection models was beyond the scope of this work, ASCENT's modularity means it can readily benefit from future advances in faster and more accurate keypoint detection or segmentation algorithms. Furthermore, while ASCENT's track linking is highly efficient, scaling to datasets with tens of thousands of neurons, such as those from larger vertebrate brains, would require integrating more advanced, scalable assignment algorithms to overcome the computational complexity of exhaustive pairwise matching.

Methods

Datasets and ground-truth

Experiments were conducted on four distinct *C. elegans* imaging datasets, representing varied anatomical regions and experimental conditions. Three datasets feature whole-brain (head ganglion) imaging, while the fourth focuses on the male tail ganglia. For all experiments, the stable red fluorescent channel, which tags neuron nuclei, was used for neuron detection and for generating the embeddings used in tracking.

1. **In-house Dataset:** This dataset was acquired to test performance under different imaging and behavioral conditions. It comprises a recording of neuronal activity across the head ganglion of *C. elegans* (strain ZM9624, expressing pan-neuronal GCaMP6 and mNeptune) confined within a microfluidic device based on the 'olfactory chip' design [42] to permit controlled chemical stimulation. Worms were subjected to repeated cycles of 60s buffer followed by a 30s OP50 supernatant stimulus. Imaging was performed on a Bruker Opterra II swept-field confocal microscope using a 40x, 0.75 NA air objective and

a Photometrics Evolve 512 Delta EM-CCD camera. The recording consists of 1,100 volumetric frames ($2 \times 9 \times 512 \times 512$; channels $\times$ z $\times$ y $\times$ x) acquired at 3.3 Hz with a spatial resolution of $1.5\mu m$ in z and $0.243\mu m$ in xy. Ground-truth tracks for 96 neurons were generated by manually proofreading and correcting an initial set of tracks generated by ZephIR [16].

2. **NeRVE Dataset [3]:** This publicly available dataset features a freely moving animal (*C. elegans* strain AML14, expressing pan-neuronal GCaMP6s and tagRFP) and serves as a challenging benchmark due to significant inter-frame movements and non-rigid deformations. The dataset was acquired using a spinning disk confocal microscope equipped with a tracking system to keep the worm centered in the field of view [3]. It consists of 1,532 volumetric frames ($2 \times 34 \times 600 \times 632$; channels $\times$ z $\times$ y $\times$ x) acquired at 6 Hz with a spatial resolution of $1.5\mu m$ in z and $0.3226\mu m$ in xy. Evaluation was performed against the manual track annotations provided by the original authors [3], which cover 79 neurons.
3. **Flavell Lab Dataset:** This dataset, originally published in Atanas et al. [41], features whole-brain imaging of a freely moving *C. elegans* (strain SWF702, expressing nuclear-localized GCaMP7f, TagRFP, and mNeptune2.5). It was included to test ASCENT's generalization across different imaging platforms and parameters, notably its near-isotropic spatial resolution. Imaging was performed on a custom spinning disk confocal microscope [41]. We analyzed one recording consisting of 1,600 volumetric frames ($2 \times 64 \times 120 \times 284$; channels $\times$ z $\times$ y $\times$ x) acquired at 1.7 Hz with a spatial resolution of approximately $0.3\mu m$ in x, y, and z [41]. Reference tracks were generated using the ANTSUN 2.0 algorithm [26] and obtained via personal communication; these represent high-quality algorithmic annotations rather than manual ground truth. The image volumes used in our analysis were obtained in their pre-processed state, including shear correction and body-axis alignment as described in Atanas et al. [41].
4. **Male Tail Dataset:** This dataset, originally from Susoy et al. [43], was used to test ASCENT's adaptability to a novel anatomical context, the *C. elegans* male tail ganglia. It comprises recordings of neuronal activity during mating behavior in males of strain ZM9624, expressing pan-neuronal nuclear-localized GCaMP6s and mNeptune [43]. Imaging was performed on a custom spinning-disk confocal microscope using a 40x, 1.3 NA oil objective [43]. Volumetric frames were acquired at 10 Hz with a spatial resolution of $1.75\mu m$ in z and $0.45\mu m$ in xy [43]. We analyzed four recordings with dimensions $1331 \times 320 \times 192 \times 20$, $3428 \times 320 \times 192 \times 20$, $3002 \times 256 \times 160 \times 16$, and $1326 \times 320 \times 192 \times 20$ (t $\times$ x $\times$ y $\times$ z). Ground-truth tracks covering approximately 76 neurons were generated via manual annotation using MaMuT [44]. The image volumes used in our analysis had undergone prior preprocessing described in [25], including coarse global alignment and difference-of-Gaussian filtering.

Data pre-processing

A key advantage of ASCENT is its minimal pre-processing pipeline. Unlike methods that require computationally expensive image registration, animal straightening, or graph construction, ASCENT operates directly on normalized volumetric data. We apply percentile normalization on a frame-by-frame basis, where intensity values are linearly rescaled such that the 1st and 99.99th percentiles of each frame's voxel intensities map to a range of [0, 1]. This approach effectively mitigates the impact of outlier pixels (e.g., dead or hot pixels) without distorting the intensity distribution of the relevant neuronal signals.

Problem formulation

ASCENT employs a tracking-by-detection strategy. Given a volumetric time-series recording, $I \in \mathbb{R}^{T \times C \times D \times H \times W}$, where T is the number of frames, C the number of channels, and D,

H, and W are the spatial dimensions, the framework proceeds as follows. For each frame I_t, a detection algorithm provides a set of n_t neuron candidates $\mathcal{O}_t = \{o_i : o_i = (t, x_i, y_i, z_i)\}_{i=1}^{n_t}$, where (x_i, y_i, z_i) are the spatial coordinates. The core of the framework is the Neuron Embedding Transformer (NETr), denoted as $f_\theta(\cdot)$, which maps the image data I_t and the set of detections $\mathcal{O}_t$ to a set of discriminative feature vectors $\{e_i\}_{i=1}^{n_t}$, where each embedding $e_i \in \mathbb{R}^d$:

$$\{e_i\}_{i=1}^{n_t} = f_\theta(I_t, \mathcal{O}_t) \tag{1}$$

These embeddings e_i serve as the basis for linking detections across time frames to build a set of tracks, $TR = \{tr_1, tr_2, ..., tr_K\}$, where K is the total number of unique neuron trajectories. Each track tr_k represents the trajectory of a single neuron, defined as a sequence that associates a persistent identity k with at most one neuron detection from $\mathcal{O}_t$ for each frame $t \in \{1, ..., T\}$.

Neuron detection

Neuron candidate coordinates for each frame were obtained using a custom-trained 3D StarDist [45–47] model specific to each dataset. To generate the necessary training data, ground-truth instance masks were created for two representative frames from each dataset. This involved first applying a pre-trained nnUNet [48] model to produce an initial binary segmentation mask that distinguishes neuron pixels from the background. Connected component analysis was then performed on this output to generate preliminary instance labels. These labels were subsequently manually proofread and corrected to ensure accurate, individual neuron identities suitable for training. Each dataset-specific StarDist model was then trained for 500 epochs using these curated instance masks. The 3D coordinates of the final detected neuron centroids served as the positional input, $\{o_i\}$, to the NETr embedding network.

NETr architecture

The Neuron Embedding Transformer (NETr) is designed to generate context-aware embeddings by combining local visual appearance and positional information, refined through a self-attention mechanism. The network processes batches of image frames and their corresponding neuron detections to produce a set of unique embeddings.

The process begins with local visual feature extraction. For each detected neuron o_i in a batch of frames, a 3D local patch of size $(p_x, p_y, p_z) = (64, 64, 3)$ pixels is extracted around its centroid coordinates. To achieve high efficiency, this operation is vectorized across all neurons in the batch using a pre-computed sampling grid. The three z-slices of each patch are treated as the input channels for a 2D ChannelViT backbone, a Vision Transformer variant adapted for multi-channel images [49]. This backbone, pre-trained on the ImageNet-1k dataset using the DINO self-supervised learning method [39], outputs a feature vector, vis_i, which is then projected to the target embedding dimension, $d = 256$. Concurrently, positional features are generated. The 3D coordinates of each neuron are normalized and processed by a 3-layer multilayer perceptron (MLP), which is implemented as a 1D convolutional network for computational efficiency, with Batch Normalization applied between layers. This module produces a positional embedding, $pos_i \in \mathbb{R}^d$. These two feature vectors are then combined via element-wise addition to yield an initial representation for each neuron: $h_i = vis_i + pos_i$. Finally, the sequence of initial representations, $\{h_i\}_{i=1}^{n_t}$, for each frame undergoes contextual refinement using a 4-layer Transformer Encoder [50]. The encoder employs self-attention with 4 attention heads, a feedforward dimension of 512, and a dropout rate of 0.1. A key padding mask is generated from the input neuron identifiers to ensure that padded, non-existent neurons are ignored during the self-attention calculation. The output from the

encoder provides the final, contextually refined embeddings $\{e_i\}$, with a final embedding dimension d of 256.

Self-supervised contrastive training of NETr

To learn discriminative embeddings without manual track annotations, we employ a self-supervised contrastive learning strategy based on the SimCLR framework [34]. The objective is to train the NETr network, f_θ, to produce embeddings that are invariant to data augmentations while remaining specific to each neuron's identity.

The general training proceeds in mini-batches. Each batch contains $B = 4$ frames randomly selected from the dataset. For each frame t in the batch, consisting of the image I_t and its n_t detections O_t, we apply two independent random augmentation pipelines, ϕ_1 and ϕ_2. This process generates two distinct augmented "views" of the same set of neurons, $(I_{t,1}, O_{t,1})$ and $(I_{t,2}, O_{t,2})$. These augmentations include transformations relevant to 3D microscopy, such as rotation, elastic deformation, resized cropping, position jitter, intensity variations, and object dropout. Both augmented views for the entire batch are passed through the NETr network to produce two sets of embeddings, $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$. The network parameters θ are then optimized by minimizing the NT-Xent (Normalized Temperature-scaled Cross-Entropy) loss [34], which is computed for each frame in the batch and then averaged. For a single frame with N' valid neurons present in both augmented views, the loss for a positive pair of embeddings (e_i, e_j), corresponding to the same neuron i under two different augmentations, is defined as:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(e_i, e_j)/\tau)}{\sum_{k=1}^{2N'} 1_{[k \neq i]} \exp(\text{sim}(e_i, e_k)/\tau)} \tag{2}$$

where $sim(u, v)$ is the cosine similarity between vectors u and v, τ is a temperature hyperparameter set to 0.05, and the indicator function $1_{[k \neq i]}$ ensures the denominator sums over all embeddings in the augmented set of $2N'$ embeddings except for the anchor e_i itself. This objective function attracts embeddings of the same neuron (positive pairs) while repelling embeddings of different neurons (negative pairs). The final loss for the frame is averaged over all positive pairs, computed symmetrically for both views.

Models were trained using the AdamW optimizer with an initial learning rate of 5×10^{-5}, a weight decay of 1×10^{-4}, and betas of (0.9, 0.999). All training was performed on a single NVIDIA H100 GPU. The total number of training epochs varied: 20 epochs for initial benchmarks, 300 for augmentation ablations, and adjusted counts for the data-efficiency study to normalize total steps. Specific protocols were adapted for the joint training and fine-tuning experiments. For joint models, one epoch processed all frames from all included datasets. Because image dimensions varied, mini-batches still contained frames from only a single recording at a time, cycling through recordings until the epoch was complete. Joint models were trained for 12 hours. Dataset-specific models for comparison were trained for 6 hours. For fine-tuning, the weights from a 12-hour joint model checkpoint were loaded, and the model was trained on the new target dataset for an additional 6 hours with a reduced learning rate of 2×10^{-5}.

Track-linking algorithm

The discriminative embeddings produced by NETr enable a simplified track-linking stage that does not require complex post-processing heuristics. After generating embeddings $\{e_i\}_t$ for all detections in each frame t, tracks were formed using serial bipartite matching. For each subsequent frame, a cost matrix $C_{ij} = -\text{sim}(e_i^{track}, e_j^t)$, where e_i^{track} is the current embedding of an active track and sim is the cosine similarity, was computed between active tracks and

new detections. A dynamic cost threshold θ_t was estimated as $0.5 \times$ (mean(top-k costs) + mean(all costs)), where k is the number of active tracks. The Hungarian algorithm was used to find the minimum cost assignment, and any potential matches with a cost $C_{ij} > \theta_t$ were rejected. Detections that remained unmatched initiated new tracks. The embeddings of successfully matched tracks were updated using an exponential moving average: $e_i^{track} \leftarrow \alpha e_i^{track} + (1 - \alpha) e_j^t$, with a momentum parameter $\alpha = 0.5$.

Track-only mode

To isolate the performance of the core embedding and linking module from errors introduced by the upstream neuron detection step, we evaluated ASCENT in a 'track-only' mode. In this mode, instead of using the output of a detection algorithm, the ground-truth neuron coordinates from the manually annotated track labels were used as the input detections, $\mathcal{O}_t$, for the NETr embedding network. This approach provides a direct measure of the embedding and linking performance by removing the confounding variable of detection inaccuracies, such as missed neurons or imprecise localizations. The subsequent tracking and evaluation procedures remained identical to the standard pipeline.

Ablation studies

1. **Architecture ablation:** To dissect the NETr architecture, we created two ablated models: a 'Image-only' model that omitted the positional encoding MLP, and a 'Positional-only' model that omitted the ChannelViT visual backbone. These were compared against the 'Standard' full model. To test for robustness, models were trained and evaluated on modified versions of the NeRVE dataset where Gaussian noise with a mean of 0 and a standard deviation (σ) of 0.1, 0.2, and 0.3 was added to the normalized image data.
2. **Augmentation ablation:** To evaluate the contribution of different data augmentations, the standard NETr model was trained on a reduced dataset of 16 regularly sampled NeRVE frames for 300 epochs. Performance was systematically tested by either including only one or two augmentations at a time, or by starting with the full suite of six augmentations and removing one or two at a time.

Tracking-evaluation metrics

Tracking accuracy was evaluated following the methodology of previous works [12, 16, 23]. Correspondences between ground-truth points and pipeline-tracked points were established by finding mutually closest pairs within a maximum Euclidean distance of $5\mu m$. Based on these matches, we defined two primary error types: Missing errors, where a ground-truth point has no matched point in the tracker output, and Mismatch errors (ID switches), where a tracked point is assigned a track ID that differs from its matched ground-truth point. We use a modified Multiple Object Tracking Accuracy (MOTA [51]) metric that excludes false positives, as ground-truth annotations typically cover only a known subset of all neurons present:

$$\text{Modified MOTA} = 1 - \frac{\text{mismatches} + \text{misses}}{\text{total ground-truth detections}} \tag{3}$$

This metric reflects the tracker's ability to correctly maintain neuron identities over time relative to the provided ground-truth.

UMAP visualization of the neuron embedding space

To visualize the structure of the learned feature space, we used the uniform manifold approximation and projection (UMAP) algorithm [40]. We fitted two separate 2D UMAP models to

project the high-dimensional neuron embeddings into a 2D space. The first model was fitted using embeddings generated by the randomly initialized NETr network (before training), and the second was fitted using embeddings from the fully trained network. To train each UMAP model, we first generated feature embeddings for all ground-truth neurons across all frames of the NeRVE dataset using the corresponding (untrained or trained) NETr network. Cosine similarity was specified as the distance metric for the UMAP algorithm, consistent with the similarity metric used in the contrastive training objective. For the visualizations shown in the main text, embeddings from an example frame and its randomly augmented counterpart were then projected into these pre-fitted 2D UMAP spaces.

t-SNE visualization of the frame space

To visualize the global image space throughout the NeRVE recording, we projected each frame into a low-dimensional space using a two-stage process. First, we trained a convolutional autoencoder to learn a compressed 32-dimensional representation for each frame. The input data for this network was generated by computing a maximum intensity projection along the z-axis for each of the 1,532 volumetric frames, with each resulting 2D image resized to 64×64 pixels. The autoencoder, consisting of a convolutional encoder and a corresponding decoder, was trained for 1,000 epochs using the Adam optimizer (learning rate = 1×10^{-3}) to minimize the mean squared error (MSE) between the input and reconstructed images. Next, the trained encoder was used to extract a 32-dimensional latent vector for every frame in the dataset. We then applied t-Distributed Stochastic Neighbor Embedding (t-SNE) to these latent vectors to generate a 2D embedding suitable for visualization.

Extraction of calcium activity

To generate ratiometric neural activity traces, we used the two-channel fluorescence data from the GCaMP (green) and structural marker (red) channels. For each tracked neuron at each time point, fluorescence intensity was calculated independently for each channel. A $7 \times 7 \times 3$ pixel (x, y, z) patch was defined around the neuron's centroid, and the raw fluorescence signal (F_G for green, F_R for red) was determined by averaging the values of the 10 brightest pixels within this patch. A channel-specific background value, calculated as the modal intensity of each frame, was then subtracted from the raw signal. The background-corrected intensities were used to compute a ratiometric signal, $R = \frac{F_G}{F_R}$, for each time point. For each neuron, a baseline fluorescence ratio, R_{50}, was established by taking the median of its ratiometric signal R over the full duration of the video recording. Finally, $\frac{\Delta R}{R_{50}} = \frac{R - R_{50}}{R_{50}}$ was calculated and plotted.

Data availability

The datasets generated and/or analyzed in the current study is available at https://zenodo.org/records/10008744. The NeRVE dataset is publicly available at https://ieee-dataport.org/open-access/tracking-neurons-moving-and-deforming-brain-dataset. The Flavell lab dataset is publicly available at https://dandiarchive.org/dandiset/000776?pos=3. The male tail dataset is publicly available at https://zenodo.org/records/10008744.

Code availability

The codes implemented for this study are available at https://github.com/lu-lab/ascent.

Supplementary information

Supplementary Figure 1, 2, 3, 4. Supplementary Table 1. Supplementary Video 1, 2.

References

[1] Schrödel, T., Prevedel, R., Aumayr, K., Zimmer, M. & Vaziri, A. Brain-wide 3D imaging of neuronal activity in Caenorhabditis elegans with sculpted light. *Nature Methods* **10**, 1013–1020 (2013).

[2] Venkatachalam, V. *et al.* Pan-neuronal imaging in roaming Caenorhabditis elegans. *Proceedings of the National Academy of Sciences* **113**, E1082–E1088 (2016).

[3] Nguyen, J. P. *et al.* Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans. *Proceedings of the National Academy of Sciences* **113**, E1074–E1081 (2016).

[4] Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M. & Keller, P. J. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods* **10**, 413–420 (2013).

[5] Kim, D. H. *et al.* Pan-neuronal calcium imaging with cellular resolution in freely swimming zebrafish. *Nature Methods* **14**, 1107–1114 (2017).

[6] Cong, L. *et al.* Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (Danio rerio). *eLife* **6**, e28158 (2017).

[7] Hasani, H. *et al.* Whole-brain imaging of freely-moving zebrafish. *Frontiers in Neuroscience* **17**, 1127574 (2023).

[8] Lemon, W. C. *et al.* Whole-central nervous system functional imaging in larval drosophila. *Nature communications* **6**, 7924 (2015).

[9] Mann, K., Gallen, C. L. & Clandinin, T. R. Whole-brain calcium imaging reveals an intrinsic functional network in drosophila. *Current Biology* **27**, 2389–2396 (2017).

[10] Aimon, S. *et al.* Fast near-whole–brain imaging in adult drosophila during responses to stimuli and behavior. *PLoS biology* **17**, e2006732 (2019).

[11] Lin, A. *et al.* Imaging whole-brain activity to understand behaviour. *Nature Reviews Physics* **4**, 292–305 (2022).

[12] Nguyen, J. P., Linder, A. N., Plummer, G. S., Shaevitz, J. W. & Leifer, A. M. Automatically tracking neurons in a moving and deforming brain. *PLoS Computational Biology* **13**, e1005517 (2017).

[13] Hanson, A. *et al.* Automatic monitoring of neural activity with single-cell resolution in behaving Hydra. *Scientific Reports* **14**, 5083 (2024).

[14] Adhinarta, J. *et al.* WormID-Bench: A benchmark for whole-brain activity extraction in C. elegans. *bioRxiv* (2025).

[15] Parag, T., Butler, V. & Chklovskii, D. Ourselin, S. & Haynor, D. R. (eds) *Tracking multiple neurons on worm images.* (eds Ourselin, S. & Haynor, D. R.) *Medical Imaging 2013: Image Processing*, Vol. 8669, 86692P. International Society for Optics and Photonics (SPIE, 2013). URL https://doi.org/10.1117/12.2000087.

[16] Ryu, J. *et al.* Versatile multiple object tracking in sparse 2D/3D videos via deformable image registration. *PLoS Computational Biology* **20**, e1012075 (2024).

[17] Gallusser, B. & Weigert, M. Leonardis, A. *et al.* (eds) *Trackastra: Transformer-based cell tracking for live-cell microscopy.* (eds Leonardis, A. *et al.*) *Computer Vision – ECCV 2024*, 467–484 (Springer Nature Switzerland, Cham, 2025).

[18] Lagache, T., Lansdell, B., Tang, J., Yuste, R. & Fairhall, A. Tracking activity in a deformable nervous system with motion correction and point-set registration. *bioRxiv* 373035 (2018).

[19] Chaudhary, S., Lee, S. A., Li, Y., Patel, D. S. & Lu, H. Graphical-model framework for automated annotation of cell identities in dense cellular images. *eLife* **10**, e60321 (2021).

[20] Lee, H. J. *et al.* Automated cell annotation in multi-cell images using an improved CRF_ID algorithm. *eLife* **12**, RP89050 (2025).

[21] Toyoshima, Y. *et al.* Neuron ID dataset facilitates neuronal annotation for whole-brain activity imaging of C. elegans. *BMC Biology* **18**, 1–20 (2020).

[22] Yu, X. *et al.* Fast deep neural correspondence for tracking and identifying neurons in C. elegans using semi-synthetic training. *eLife* **10**, e66410 (2021).

[23] Wu, Y. *et al.* Rapid detection and recognition of whole brain activity in a freely behaving Caenorhabditis elegans. *PLoS Computational Biology* **18**, e1010594 (2022).

[24] Jones, C., Barzegar-Keshteli, M., Gross, A., Obozinski, G. & Rahi, S. J. A graph matching approach to tracking neurons in freely-moving C. elegans. *bioRxiv* 2023–11 (2023).

[25] Park, C. F. *et al.* Automated neuron tracking inside moving and deforming C. elegans using deep learning and targeted augmentation. *Nature Methods* **21**, 142–149 (2024).

[26] Atanas, A. A. *et al.* Deep neural networks to register and annotate cells in moving and deforming nervous systems. *bioRxiv* 2024–07 (2025).

[27] Moen, E. *et al.* Deep learning for cellular image analysis. *Nature Methods* **16**, 1233–1246 (2019).

[28] Yazdi, R. & Khotanlou, H. A survey on automated cell tracking: challenges and solutions. *Multimedia Tools and Applications* **83**, 81511–81547 (2024).

[29] Volpe, G. *et al.* Roadmap on deep learning for microscopy. *Journal of Physics: Photonics* (2023).

[30] Wang, Y. *et al.* Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* **38**, 1–12 (2019).

[31] Sarlin, P.-E., DeTone, D., Malisiewicz, T. & Rabinovich, A. *Superglue: Learning feature matching with graph neural networks* (2020).

[32] Lu, X., Yan, Y., Kang, B. & Du, S. Paraformer: Parallel attention transformer for efficient feature matching. *Proceedings of the AAAI Conference on Artificial Intelligence* **37**, 1853–1860 (2023).

[33] Deng, Y., Zhang, K., Zhang, S., Li, Y. & Ma, J. ResMatch: Residual attention learning for feature matching. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**, 1501–1509 (2024). URL https://ojs.aaai.org/index.php/AAAI/article/view/27915.

[34] Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. III, H. D. & Singh, A. (eds) *A simple framework for contrastive learning of visual representations.* (eds III, H. D. & Singh, A.) *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, 1597–1607 (PMLR, 2020). URL https://proceedings.mlr.press/v119/chen20j.html.

[35] He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. *Momentum contrast for unsupervised visual representation learning* (2020).

[36] Chen, X., Fan, H., Girshick, R. & He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[37] Grill, J.-B. *et al.* Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds) *Bootstrap your own latent - a new approach to self-supervised learning.* (eds Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H.) *Advances in Neural Information Processing Systems*, Vol. 33, 21271–21284 (Curran Associates, Inc., 2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[38] Caron, M. *et al.* Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds) *Unsupervised learning of visual features by contrasting cluster assignments.* (eds Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H.) *Advances in Neural Information Processing Systems*, Vol. 33, 9912–9924 (Curran Associates, Inc., 2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

[39] Caron, M. *et al. Emerging properties in self-supervised vision transformers*, 9650–9660 (2021).

[40] McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[41] Atanas, A. A. *et al.* Brain-wide representations of behavior spanning multiple timescales and states in C. elegans. *Cell* **186**, 4134–4151 (2023).

[42] Chronis, N., Zimmer, M. & Bargmann, C. I. Microfluidics for in vivo imaging of neuronal and behavioral activity in Caenorhabditis elegans. *Nature Methods* **4**, 727–731 (2007).

[43] Susoy, V. *et al.* Natural sensory context drives diverse brain-wide activity during C. elegans mating. *Cell* **184**, 5122–5137 (2021).

[44] Wolff, C. *et al.* Multi-view light-sheet imaging and tracking with the MaMuT software reveals the cell lineage of a direct developing arthropod limb. *eLife* **7**, e34410 (2018).

[45] Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C. & Fichtinger, G. (eds) *Cell detection with star-convex polygons*. (eds Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C. & Fichtinger, G.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, 265–273 (Springer International Publishing, Cham, 2018).

[46] Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. *Star-convex polyhedra for 3d object detection and segmentation in microscopy* (2020).

[47] Weigert, M. & Schmidt, U. *Nuclei instance segmentation and classification in histopathology images with stardist*, 1–4 (2022).

[48] Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J. & Maier-Hein, K. H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods* **18**, 203–211 (2021).

[49] Bao, Y., Sivanandan, S. & Karaletsos, T. Channel vision transformers: an image is worth 1 x 16 x 16 words. *arXiv preprint arXiv:2309.16108* (2023).

[50] Vaswani, A. *et al.* Guyon, I. *et al.* (eds) *Attention is all you need*. (eds Guyon, I. *et al.*) *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017). URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[51] Bernardin, K. & Stiefelhagen, R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing* **2008**, 246309 (2008). URL https://doi.org/10.1155/2008/246309.

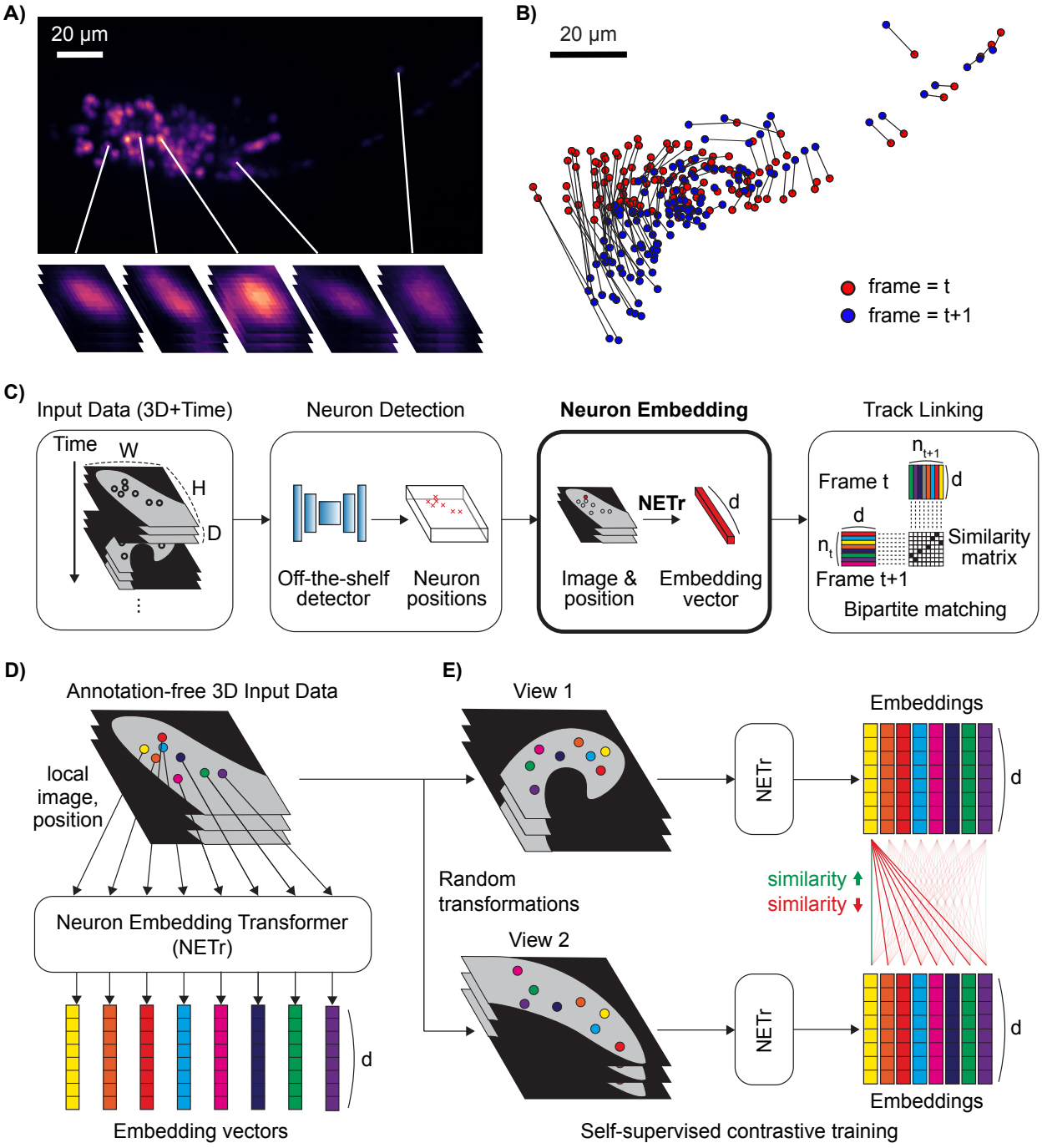


Fig. 1: **The ASCENT framework for annotation-free neuron tracking. A**,**B**) Challenges in 3D neuron tracking, including visually indistinguishable neurons in a dense cluster (**A**) and complex, non-rigid frame-to-frame movements (**B**). **C**) The ASCENT pipeline proceeds from neuron detection in input 4D data to neuron embedding via NETr, and finally to track linking. **D**) Conceptual diagram of the Neuron Embedding Transformer (NETr) in the neuron embedding step. NETr takes a 3D image and detected neuron positions as input to produce a unique, d-dimensional embedding vector for each neuron. **E**) Conceptual diagram of the self-supervised training scheme. For a given frame, two different, randomly augmented "views" are generated and passed through the embedding network. A contrastive loss then ensures that the embeddings for the same neuron are consistent across views, while embeddings for different neurons are pushed apart.

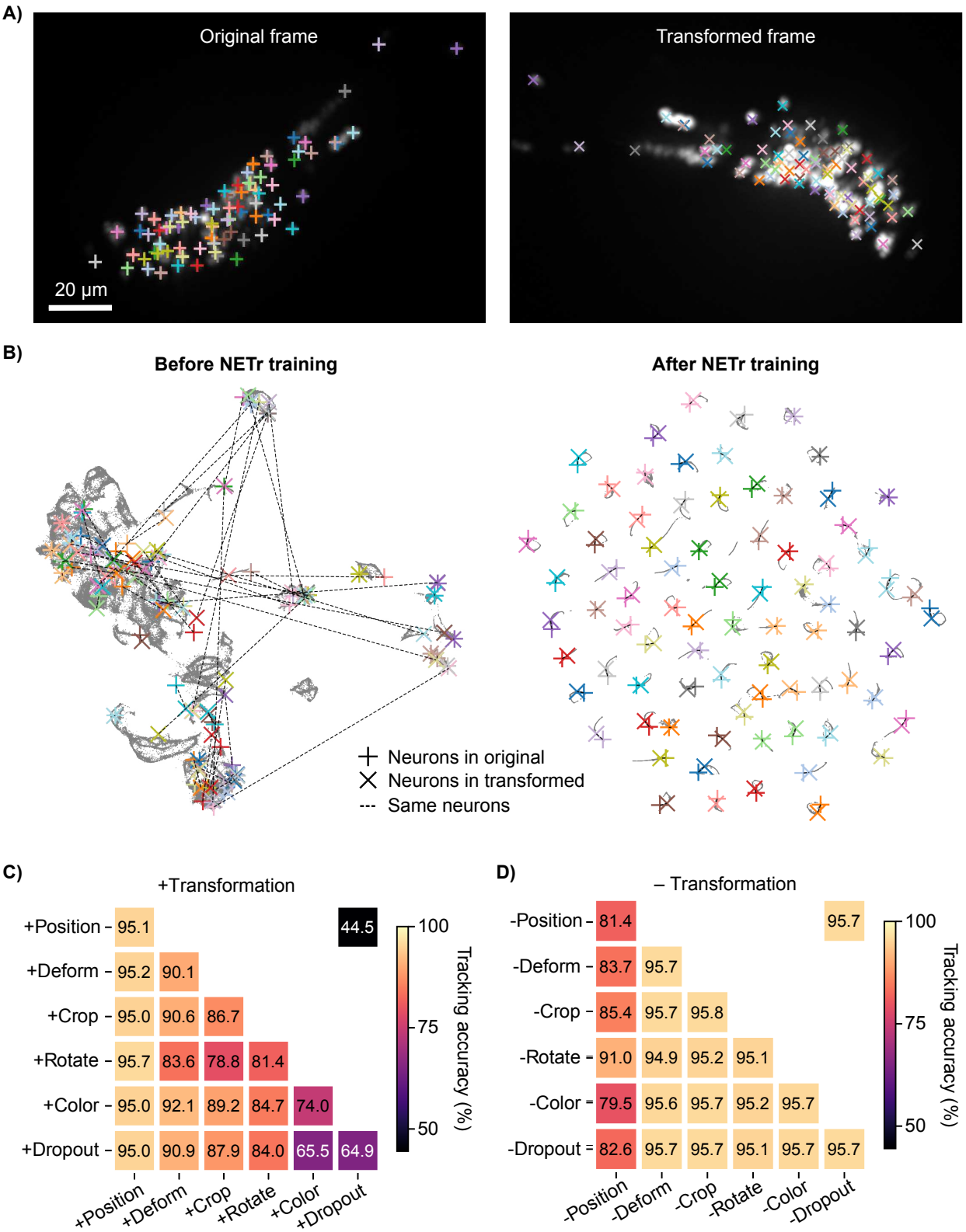


Fig. 2: A tailored data augmentation strategy creates a discriminative embedding space. Caption on next page.

Fig. 2: **An experimental condition-aware data augmentation strategy creates a discriminative embedding space. A**) Example of a random data augmentation applied to a frame from the NeRVE dataset. The original neuron positions are marked with '+' and the transformed positions with '×'. **B**) UMAP visualization of the neuron embedding space of all neurons in the dataset before (left) and after (right) self-supervised training. Colored markers represent neurons from the original (+) and transformed (×) views shown in **A**; gray points represent all other neurons. Dotted lines connect corresponding neuron pairs, which are scattered randomly across the space. **C**,**D**) Quantitative analysis of augmentation contributions on the NeRVE dataset. Heatmaps show the tracking accuracy gain when adding one or two transformations to a baseline (**C**), and the accuracy drop when removing one or two transformations from the full set (**D**), confirming the positive contribution of each component.

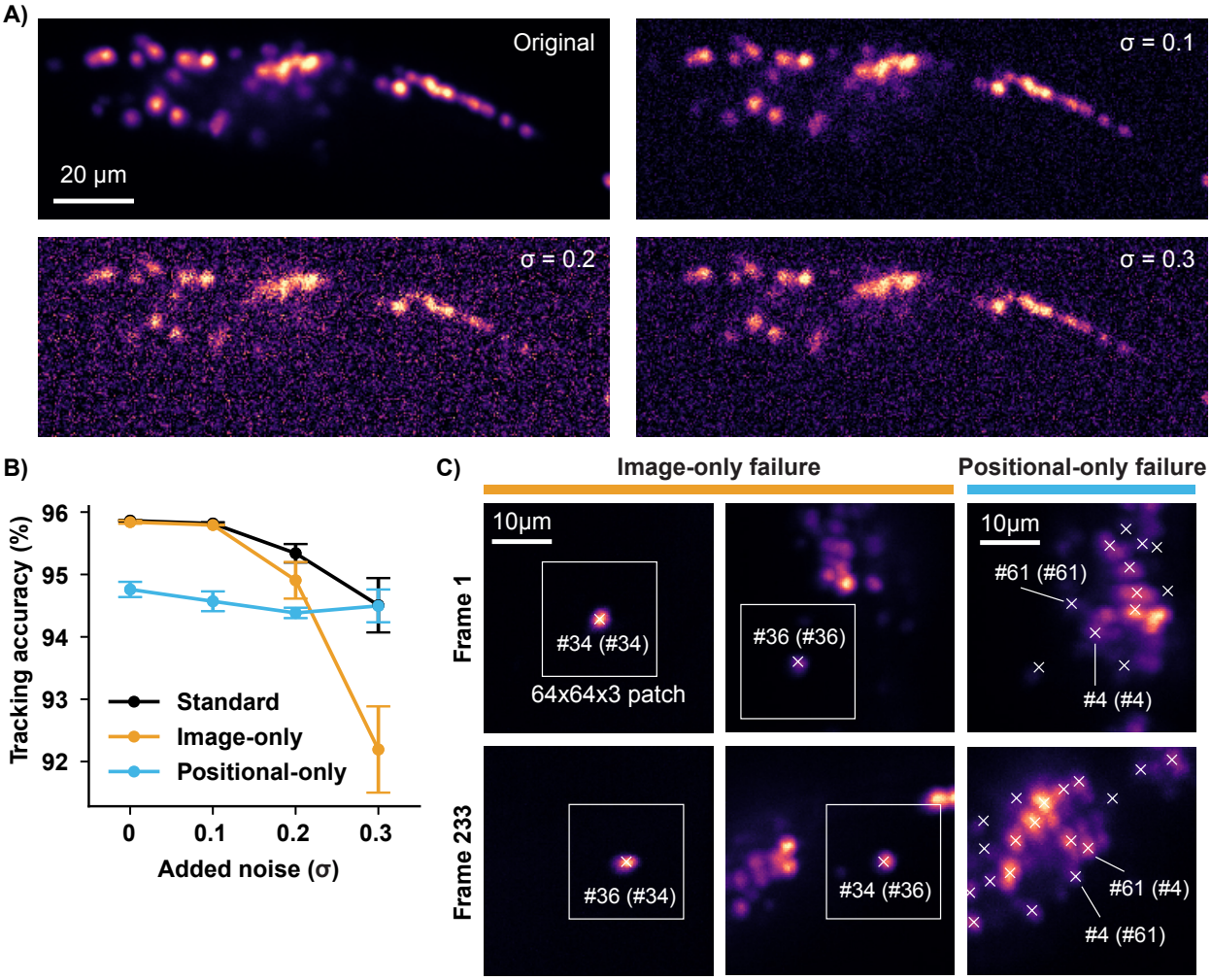


Fig. 3: **Synergistic use of image and positional features enables robust tracking.** **A)** Example of a frame from the NeRVE dataset with varying levels of synthetic Gaussian noise applied, from $\sigma = 0.1$ to $\sigma = 0.3$. **B)** Tracking accuracy of NETr architecture variants as a function of image noise. The 'Standard' model (black), using both feature types, achieves the highest accuracy in low-to-moderate noise conditions. The 'Image-only' model (red) performs well on clean data but its accuracy degrades severely with noise. The 'Positional-only' model (blue) is robust to noise even at extreme noise levels. Data are presented as mean ± standard deviation over three independently trained models. **C)** Qualitative examples of the distinct failure modes of ablated models. Labels are formatted as 'predicted neuron # (ground-truth neuron #)'. Left & Center: The 'Image-only' model fails by swapping two visually similar neurons (#34 and #36) between frame 1 and frame 233. The white box indicates the 64x64 pixel local patch from which image features are extracted. Right: The 'Positional only' model fails by swapping two spatially proximal but visually distinct neurons (#4 and #61) between frame 1 and frame 233.

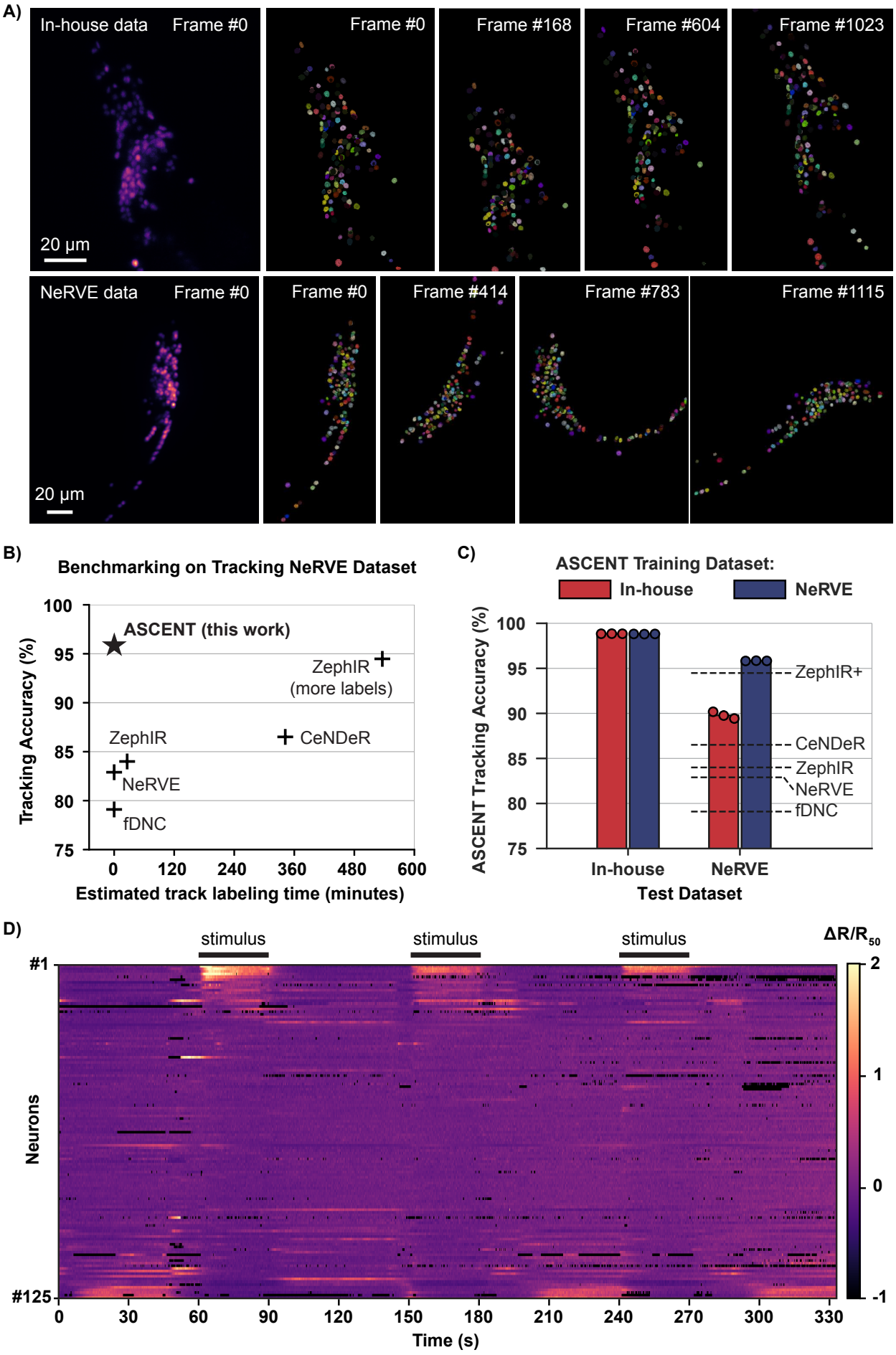


Fig. 4: ASCENT achieves robust, state-of-the-art performance and generalizes across diverse datasets. Caption on next page.

Fig. 4: **ASCENT achieves state-of-the-art performance on the NeRVE benchmark and generalizes across diverse datasets. A)** Example frames from the in-house and NeRVE datasets, illustrating differences in imaging conditions and animal confinement. **B)** Tracking accuracy versus manual labeling time required for various methods on the NeRVE benchmark. ASCENT achieves the highest accuracy with zero labeling time. Labeling time for other methods is estimated based on the original NeRVE publication [12] and the ZephIR publication [16]. The 'CeNDeR' model used annotations from 130 frames (342 minutes labeling time) [23], the 'ZephIR' model used annotations from 10 frames (26 minutes labeling time), and the 'ZephIR (more labels)' used anntations from 10 frames and full track annotations for additional 10 neurons across all 1,536 frames (536 minutes labeling time) [16]. **C)** Cross-dataset generalization performance. Bars show the tracking accuracy of different training and test dataset combinations. Dotted lines indicate the performance of competitor methods on the NeRVE dataset for reference. 'ZephIR+' referes to the 'ZephIR (more labels)' model in **B**. **D)** Ratiometric neural activity traces ($\Delta R/R_{50}$) extracted from the in-house dataset using ASCENT's tracking results. Attractive food odor stimuli were applied during 60–90s, 150–180s, and 240–270s.

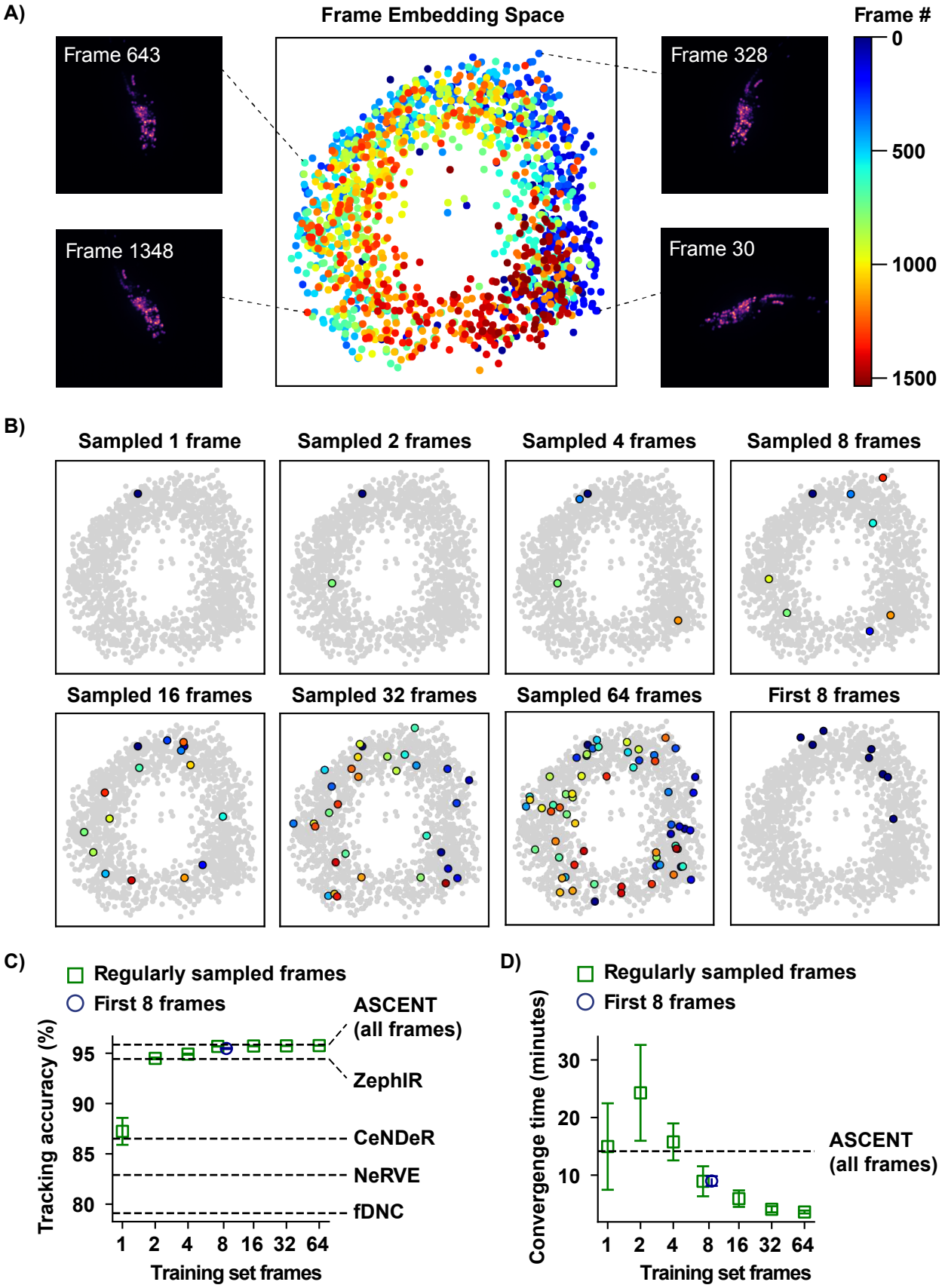


Fig. 5: High data efficiency enables rapid model training on minimal data. Caption on next page.

Fig. 5: **High data efficiency enables rapid model training on minimal data. A)** t-SNE visualization of the image space of all frames from the NeRVE dataset. Each point represents a single frame, colored according to its point in time based on the color bar (right). Inset images show representative worm postures from different regions of the space. **B)** Visualization of the subsets of frames used for training. Sampled frames for each condition are highlighted in color against the full set of frames (gray). **C)** Tracking accuracy as a function of the number of training frames used. Accuracy plateaus quickly, achieving >95% with as few as 8–16 frames. The performance of competitor methods is shown as horizontal dashed lines for reference. **D)** Wall-clock convergence time as a function of training set size. Training is rapid across all conditions, typically requiring less than 15 minutes to reach convergence. Data are from the NeRVE dataset; error bars in **C** and **D** represent mean ± standard deviation. over three independent runs.

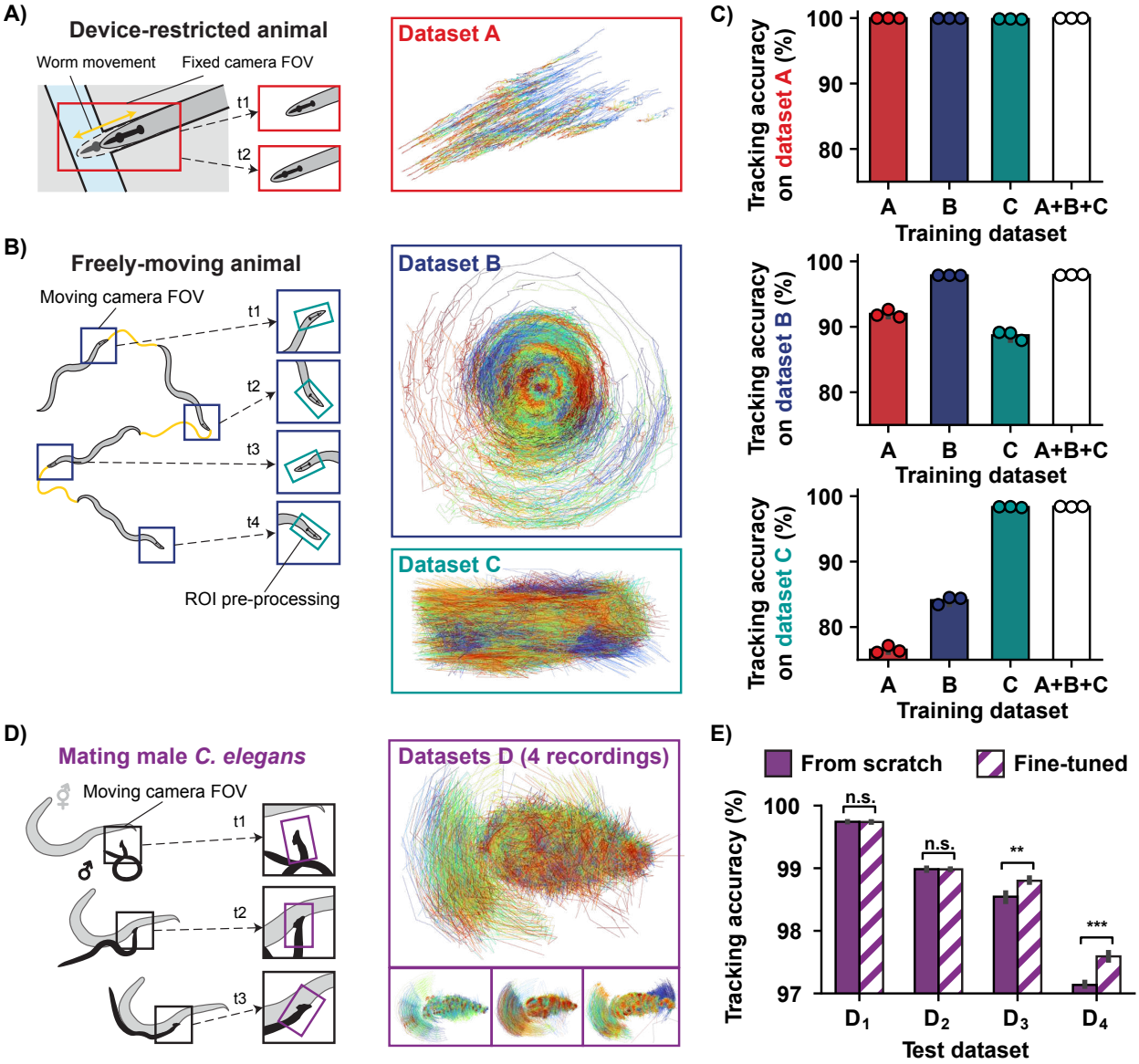
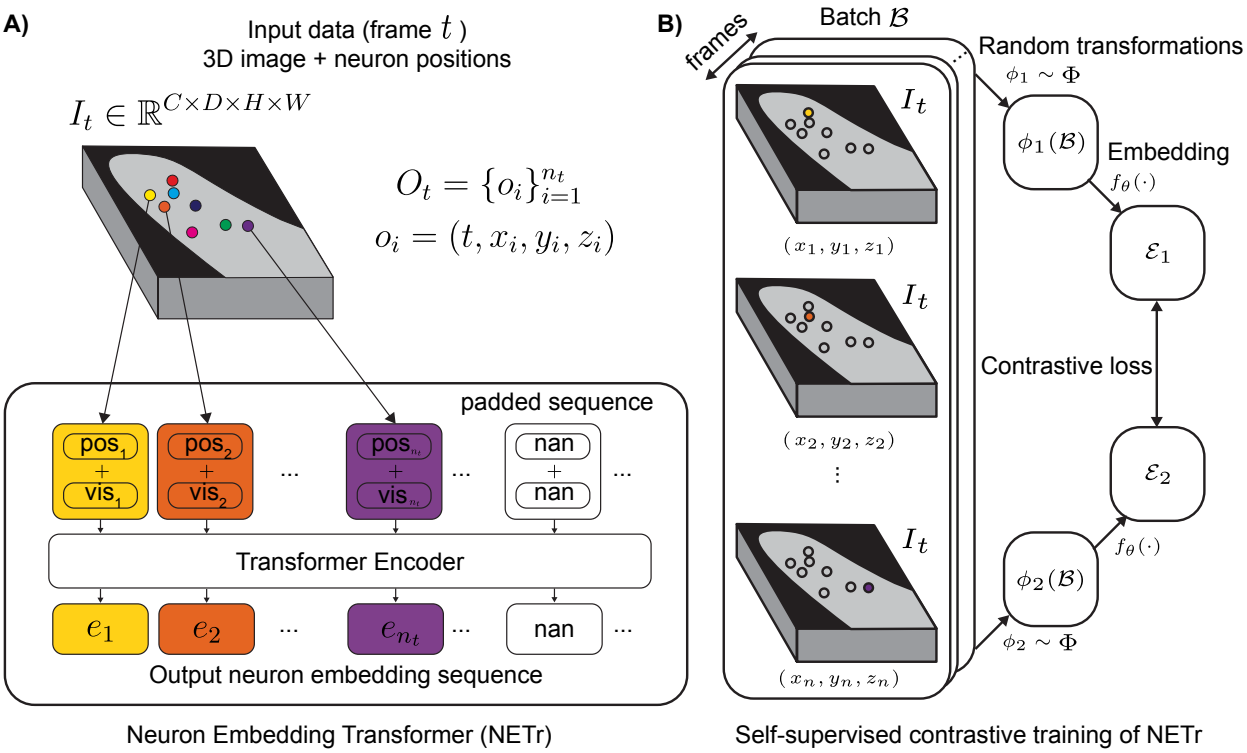
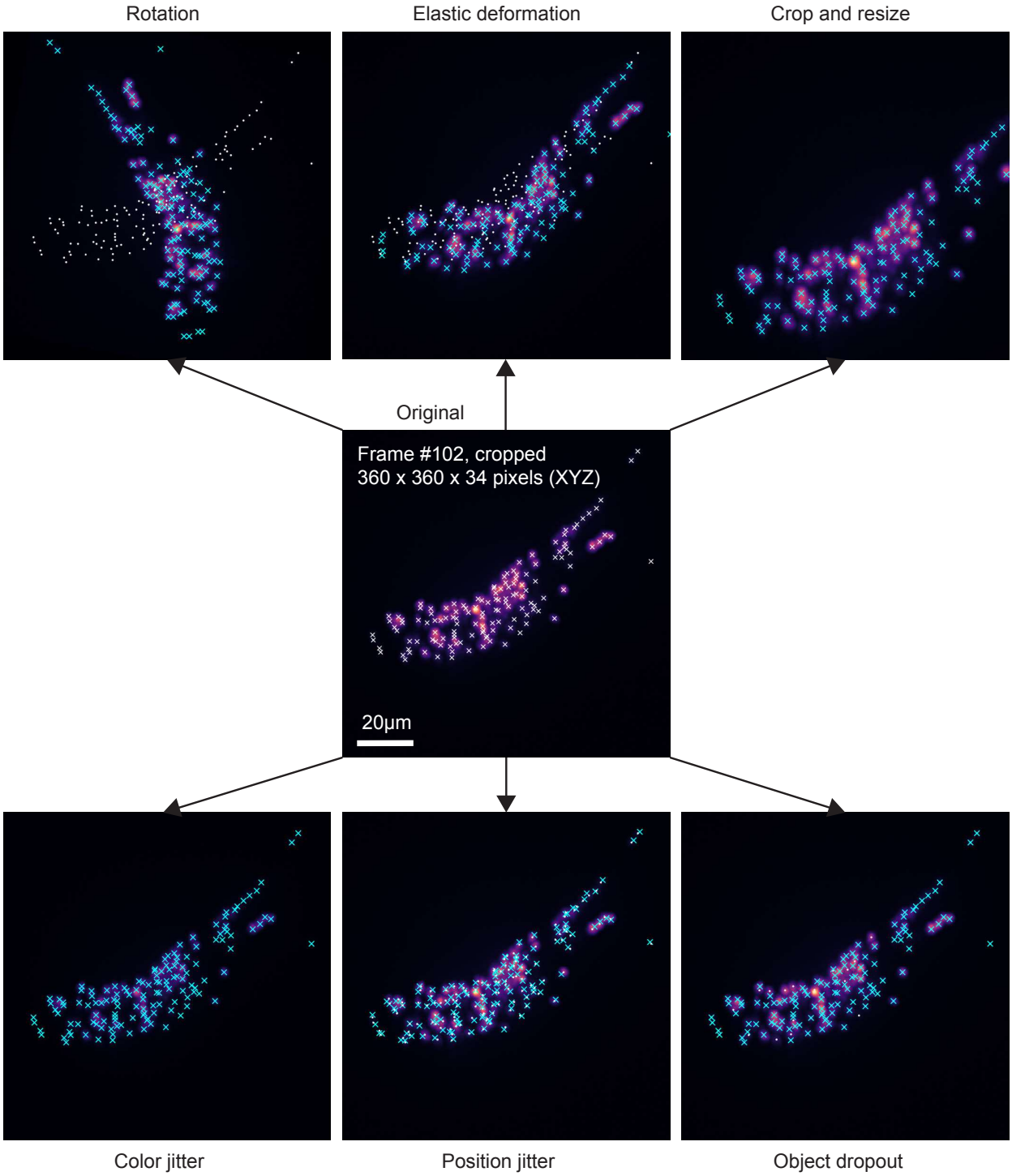


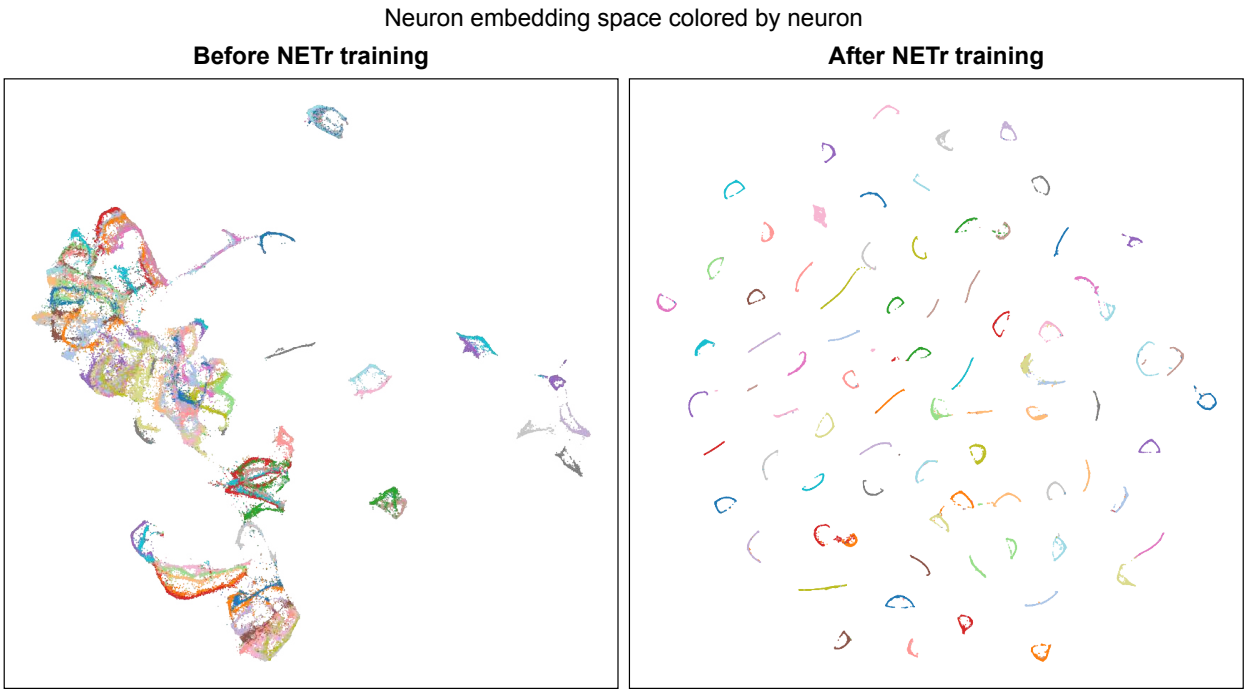
Fig. 6: Joint model generalization and rapid fine-tuning of ASCENT. A) Representative imaging from Dataset A (device-restricted worm) showing a constrained motion regime under a fixed field of view. **B**) Example recordings from Datasets B and C, both freely moving animals. Dataset B shows unaligned motion, whereas Dataset C has been body-axis aligned by preprocessing. **C**) Cross-dataset generalization. Models trained individually on A, B, or C and a joint A + B + C model were evaluated on each test domain. The joint model achieved tracking accuracies statistically indistinguishable from the dataset-specific models on B and C, and only marginally lower on A (99.98% vs. 99.99%, $p < 0.05$). Both models reached near-perfect accuracy, confirming that joint training supports robust generalization across imaging domains. **D**) Representative imaging of the male-tail Datasets D, capturing a distinct anatomical region and movement pattern of a male *C. elegans* during mating. **E**) Fine-tuning versus from-scratch training on four independent male-tail recordings (D_1–D_4). Fine-tuned models significantly outperformed from-scratch models on D_3 ($p = 0.011$) and D_4 ($p = 0.001$), while differences on D_1 ($p = 0.820$) and D_2 ($p = 0.151$) were not significant.



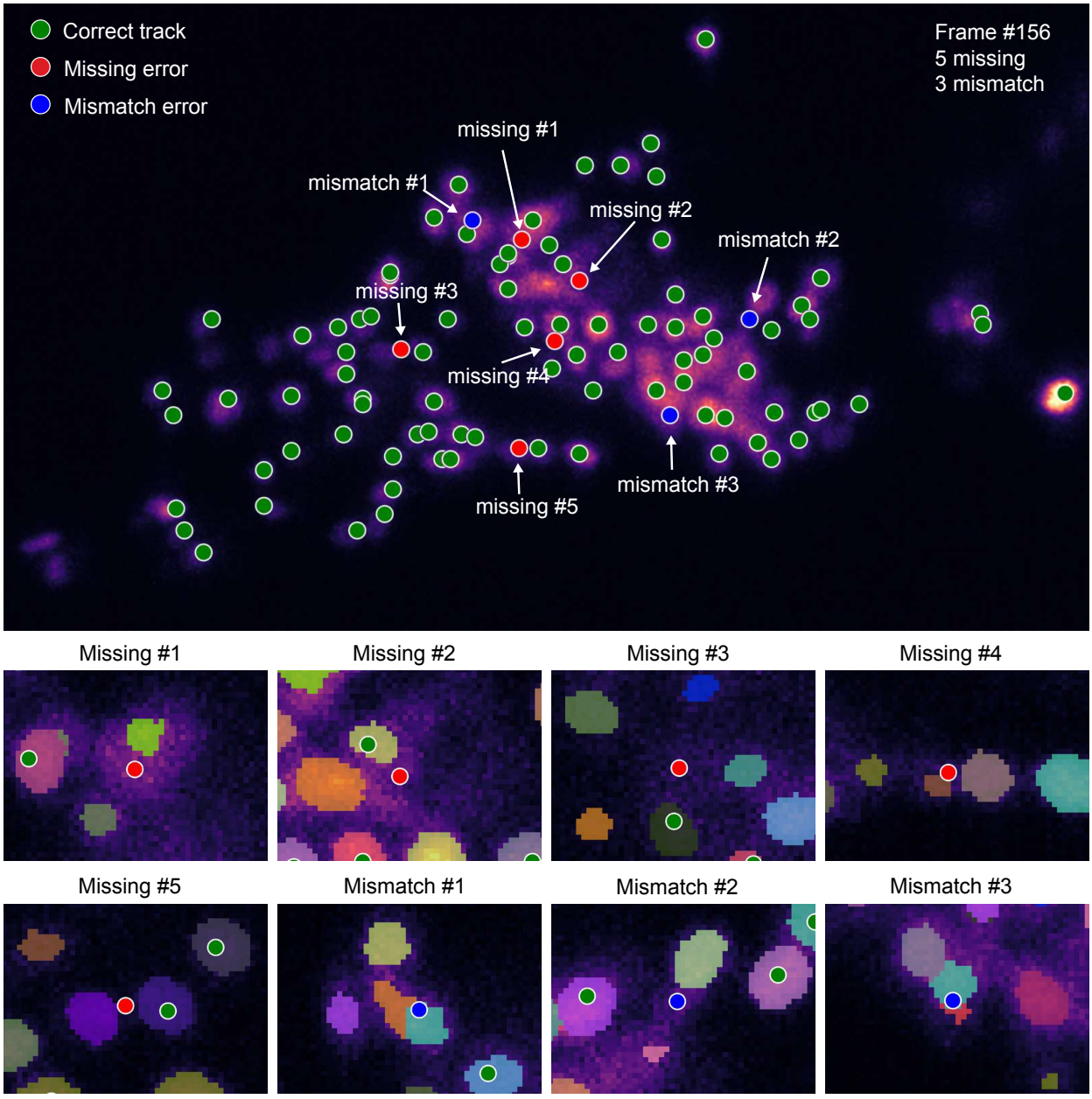
Supplementary Fig. 1: **Detailed architecture and training scheme of the Neuron Embedding Transformer (NETr). A)** The NETr architecture. For each detected neuron, NETr combines local visual features (vis_i) with positional features (pos_i). Visual features are extracted from a local 3D patch using a DINO-pretrained ChannelViT backbone. Positional features are generated from the neuron's 3D coordinates using a multi-layer perceptron (MLP). A Transformer Encoder then uses self-attention to integrate contextual information from all neurons in the frame ($o_1 ... o_{n_t}$) to produce final, refined embeddings (e_i). **B)** The self-supervised training scheme. The network is trained by applying two independent random transformations (ϕ_1, ϕ_2) to batches of frames. A contrastive loss function maximizes the similarity between embeddings of the same neuron ($e_i^{(1)}, e_i^{(2)}$) from the two augmented views, while minimizing similarity between different neurons.



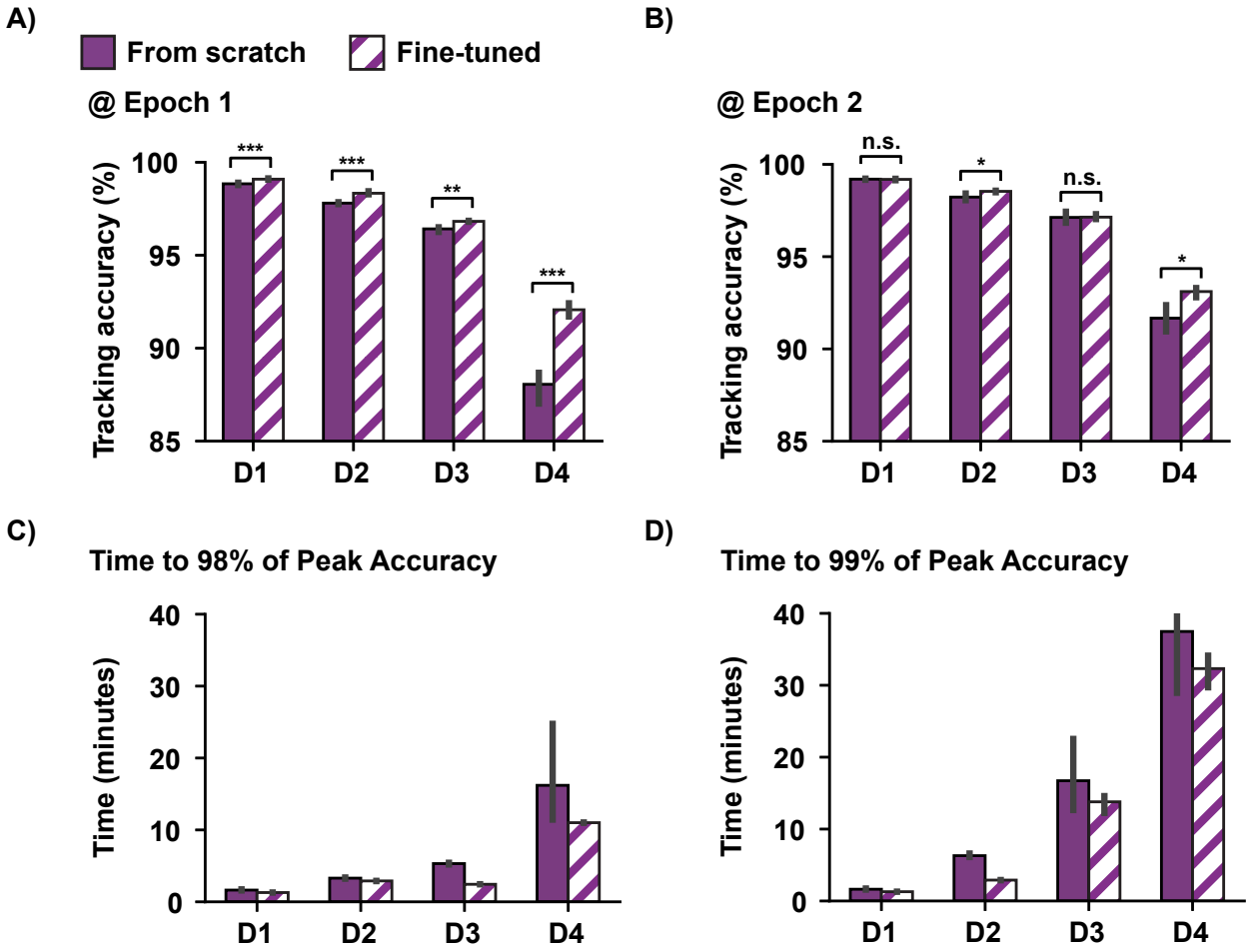
Supplementary Fig. 2: **Data augmentations for self-supervised learning of NETr.** Examples of the six individual transformations applied to an original frame from the NeRVE dataset (center), shown as maximum intensity projections of a 360×360 pixel cropped region. In the original image, detected neurons are marked with white '×'s. In the transformed images, the new neuron positions are marked with cyan '×'s, while the original positions are shown with white dots where applicable to illustrate the transformation's effect. Geometric transformations such as rotation, elastic deformation, and crop/resize alter both image and coordinate data. Color jitter modifies only the image data, while position jitter and object dropout alter only the coordinate data.



Supplementary Fig. 3: **UMAP visualization of the neuron embedding space.** High-dimensional feature embeddings for all ground-truth neurons across all frames of the NeRVE dataset were projected into a 2D space using UMAP. Each point represents a single neuron at a single timepoint, and points are colored according to their unique neuron identity. **(left)** Before training, the embeddings are unstructured and intermingled, indicating the network cannot distinguish between different neurons. **(right)** After self-supervised contrastive training, the embeddings form tight, identity-specific clusters, demonstrating that the network has learned a discriminative feature space where each neuron identity occupies a unique region of the space.



Supplementary Fig. 4: **Qualitative analysis of tracking errors.** Detailed breakdown of tracking errors in an error-prone frame from the in-house dataset (frame #156). The main panel shows a maximum intensity projection of the frame, with ground-truth neurons colored by their tracking status: correctly tracked (green), missing (red), or mismatched (blue). The eight bottom panels show zoomed-in views of individual errors at the relevant z-plane, with the StarDist-generated segmentation mask overlaid. The analysis reveals that the majority of errors are attributable to the upstream detection step. Missing #2, #3 are due to StarDist failing to detect the neurons. Missing #1, #4, #5 and mismatch #2 are caused by incomplete segmentation masks that do not sufficiently cover the ground-truth neuron position. Mismatch #1, #3 occur where StarDist detects two distinct objects near a single ground-truth neuron, suggesting potential ambiguities in the ground-truth annotation itself.



Supplementary Fig. 5: **Fine-tuning enables rapid performance gains on male-tail datasets. A-B)** Tracking accuracy of ASCENT models trained from scratch or fine-tuned on four independent male-tail recordings (D_1–D_4) after the first and second training epochs. In contrast to from-scratch models, fine-tuned models rapidly reach high performance, exceeding 92% accuracy on all recordings after a single epoch. **C-D)** Training time required to reach 98% and 99% of each model's peak accuracy (average of from-scratch and fine-tuned models). Fine-tuned models consistently require less training time to achieve these benchmark performances, confirming that initializing from a pre-trained joint model substantially accelerates convergence without compromising final accuracy.

Supplementary Table 1: **Benchmark evaluation of ASCENT compared to existing methods on the in-house and NeRVE datasets**. "t.o." (tracking only) refers to a processing mode where only ground-truth points are embedded and tracked. Labeling time is estimated based on the original NeRVE publication [12], which reported an annotation rate of approximately 2 seconds per neuron. Accuracy is measured using Modified MOTA (Equation 3).

Method	Dataset	**Detections /vol**	**Tracks w/ GT**	**Track label used**	**Labeling time**	**Tracking accuracy**	**Inference time**	**Tracking only time**
ASCENT	This paper	129.4	96	None	0	98.87%	0.93 s/vol	0.02 s/vol
ASCENT (t.o.)	This paper	96	96	None	0	99.99%	—	0.02 s/vol
ASCENT	NeRVE	133.0	79	None	0	95.90%	2.39 s/vol	0.02 s/vol
ASCENT (t.o.)	NeRVE	79	79	None	0	98.07%	—	0.02 s/vol
ZephIR [16]	NeRVE	79	79	10 frames + 10 neurons	536m	94.48%	1.08 s/vol	—
ZephIR [16]	NeRVE	79	79	10 frames	26m	84.0%	1.08 s/vol	—
CeNDeR [23]	NeRVE	118	69	130 frames	342m	86.51%	—	0.01 s/vol
NeRVE [12]	NeRVE	119	69	None	0	82.9%	$>$ 40 s/vol	—
fDNC [22]	NeRVE	118.4	69	None	0	79.1%	—	0.01 s/vol

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryVideo1InHouse.mp4
- SupplementaryVideo2NeRVE.mp4