Higher-Order Cell Tracking Transformer

Jordão Bragantini **Ilan Theodoro** **Loïc A. Royer**
Biohub
San Francisco, CA
{jordao.bragantini, ilan.silva, loic.royer}@biohub.org

Abstract

Reconstructing lineages from live-imaging microscopy requires linking cell detections across time, including through cell divisions. A common approach is to construct a candidate graph and associate cell segmentations (nodes) across frames. However, these and other existing methods overlook two structural obstacles in candidate tracking graphs: (i) cell divisions entangle distinct lineage paths in the node embedding space, and (ii) edges sharing a node have near-random label agreement, so the candidate-graph topology offers little label signal for message passing to aggregate. We propose the **Higher-Order Cell Tracking Transformer** (HOCT), an edge-centric architecture in which candidate cell links attend to one another under a 3D geometric prior, resolving both issues. Evaluated on the Cell Tracking Challenge and a bacteria division benchmark, HOCT achieves state-of-the-art results without deep pre-trained image encoders. Moreover, the proposed approach is easier to fine-tune, quickly reducing tracking errors by 59% with 400 annotations in a human-in-the-loop setting, outperforming LoRA fine-tuning of competing transformer baselines (6.3% error reduction). The implementation is publicly available at `https://github.com/royerlab/hoct`.

1 Introduction

Since the earliest microscopes [22], advances in imaging have been inseparable from advances in biology. Modern live-imaging platforms [27, 12, 6, 61, 51] routinely produce terabytes of volumetric time-lapse data, enabling high-content screening [15], the study of embryonic development [32], tissue regeneration [11], and more. A fundamental computational challenge in these applications is *cell tracking*: the reconstruction of cell lineages, capturing trajectories and division events over time.

The current dominant paradigm is *tracking by detection* [24, 37], in which cells are first segmented in each frame and subsequently linked into lineages. Unlike multi-object tracking in natural images [13], cells are visually near-identical and lack persistent appearance cues, impairing re-identification methods [60, 42] that succeed in pedestrian or vehicle tracking [4, 3, 40]. Worse, cells can *divide*, changing their morphology and producing a variable number of objects. Cell tracking has therefore relied on specialized solutions that exploit spatial information, either explicitly via optical-flow [20, 36, 48] or implicitly by feeding appearance and positional features to learned models [2, 17].

Recent learning-based approaches employ neural networks on spatiotemporal graphs. Trackastra [17] uses an encoder-decoder Transformer with multi-dimensional Rotary Position Embeddings (RoPE) [52, 21] to embed each cell detection (node) into a latent space and compute association probabilities from cosine similarity between all nodes in adjacent frames. Ben-Haim and Raviv [2] and Braso and Leal-Taixe [9] formulate tracking as edge classification on spatiotemporal graphs, for cell and pedestrian tracking respectively, the former using GNN message passing on the candidate graph [49].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

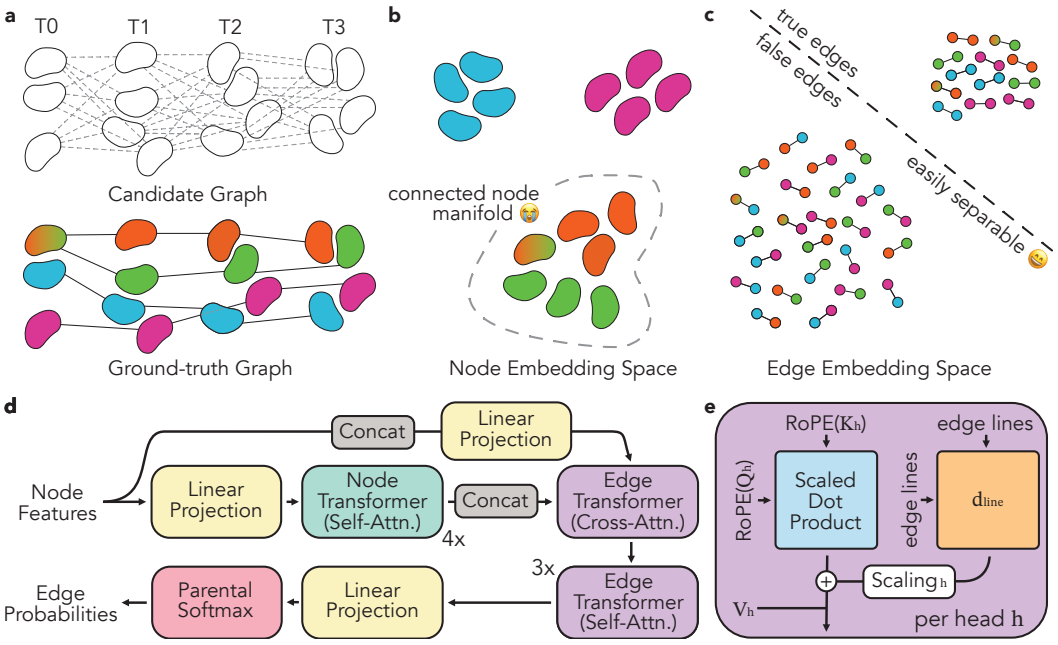


Figure 1: **Motivation and architecture.** (a) A candidate tracking graph and its ground-truth solution, where each color denotes a distinct cell trajectory (simple path in the lineage tree). The graph is non-homophilic: edges sharing a node have near-random label agreement, so graph topology carries little first-order label signal. (b) In node embedding approaches, divisions create connected manifolds that merge distinct lineage paths, confounding similarity-based association because distinct paths are connected in the embedding space by the division junction. (c) In our edge-centric approach, each candidate link is an independent token; attention is biased by inter-link geometry rather than the candidate-graph topology, removing both the connected-manifold and the non-homophilic topology issues. (d) HOCT architecture: node features are projected and refined through $L_n{=}4$ self-attention layers with 3D RoPE; edge tokens are constructed from node pairs (Eq. 1), then refined through $L_e{=}4$ attention layers with RoPE and line-to-line distance biases; a classification head produces per-edge logits, normalized via the parental softmax. (e) Detail of the edge attention layer with geometric distance bias from edge line segments (Eq. 3).

Node-embedding approaches face two structural problems on this graph (Fig. 1a–b). First, divisions create a *connected-manifold* effect: when a cell p divides into daughters d_1, d_2, the embedding of p must be close to both, merging lineage paths that should form separate clusters and leaving them prone to identity swapping. Second, the candidate graph is non-homophilic [64, 44]: only one (or two, at divisions) of the many edges incident to a target node is correct, so edges sharing a node have near-random label agreement (Sec. 4, $\mathcal{H}_{\text{adj}} \approx 0$). Message passing therefore aggregates noise, which explains why edge-classifying GNNs [2] have not consistently surpassed baselines.

Both issues are resolved by working in an *edge* embedding space, with attention biased by inter-edge geometry rather than candidate-graph adjacency (Fig. 1c). We call this approach **Higher-Order Cell Tracking Transformer** (HOCT), a two-stage architecture where: (1) node self-attention contextualizes cell detections, and (2) edge self-attention refines candidate-link representations through edge-to-edge comparisons with geometric distance biases. The term *higher-order* reflects that the model compares edges (second-order structures) to each other. Our main contributions are:

- An **edge-centric Transformer architecture** for cell tracking that compares candidate links through geometry-driven self-attention, bypassing the weak topological label signal and resolving the connected-manifold problem at divisions (Sec. 3.1).
- A **geometric distance encoding** based on the line-to-line distance between edge pairs, injected as a learnable, per-head attention bias (Sec. 3.2).
- A **multi-frame parental softmax** that extends the single-frame formulation of [17] to multiple temporal gaps with an implicit no-parent option, variable appearance costs, and a two-pass tracklet ILP solver (Sec. 3.3).

- An **incremental correction** experiment showing that a simple logistic regression head on frozen HOCT features outperforms LoRA fine-tuning of a node transformer [31], demonstrating that edge-level representation quality dominates more complex approaches (Sec. 4.5).

2 Related work

Cell tracking. Classical pipelines formulate tracking as an optimal matching problem on a candidate graph through: linear assignment [24, 28], network flow [62], or general integer linear programming (ILP) [50, 54, 7], optimizing association costs globally under biological constraints on division, appearance, and disappearance. Learned matching costs are increasingly popular: optical-flow advection of detections across frames [20, 53], sparse-annotation association costs with ILP-based lineage reconstruction [36], and contrastive cell representations for direct association on centroids [65] or cell patches [19].

Transformers for tracking. The Transformer architecture [56] has been applied to MOT [40, 63] and object detection [10]. Trackastra [17] adapts the Transformer for cell tracking with RoPE-modulated attention over node tokens, achieving strong results on the Cell Tracking Challenge [37].

Positional encoding. Rotary Position Embeddings (RoPE) [52] encode positions by rotating query and key vectors to modulate the attention between spatially distant tokens. Heo et al. [21] extended RoPE to higher dimensions for vision, and Liu et al. [34] refined it with learnable reflections. We apply 3D RoPE with learnable per-head frequencies, giving each head an independent spatial encoding.

Non-homophilic graphs. Most graph neural networks assume *homophily*, the property that connected elements share similar labels, so neighborhood aggregation reinforces correct predictions [64]. When this assumption fails, aggregation mixes uninformative or conflicting signals and degrades representations [44]; proposed remedies remain topology-driven [64]. Although Transformers can in principle attend to all tokens, spatial positional encodings, such as RoPE, reintroduce a topology-like bias. As in tracking, candidate graphs are constructed by connecting detections within a distance threshold τ; RoPE decays attention with spatial distance, so the highest-weighted tokens coincide with the graph neighbors. When attention is further restricted by a spatial mask (as in both Trackastra [17] and our node stage), the effective receptive field collapses to the graph neighborhood. Since candidate tracking graphs are non-homophilic (Sec. 4), this amounts to aggregating topologically adjacent but label-uninformative neighbors. HOCT's edge attention sidesteps this: attention is biased by the line-to-line distance between candidate links, a geometric relationship that captures spatial compatibility beyond the original graph adjacency.

3 Higher-order cell tracking

Let $G_C = (V_C, E_C)$ be a candidate graph where V_C is the set of cell detections across all frames and E_C contains candidate links between detections in different frames within a spatial distance threshold τ. Each node $i \in V_C$ carries features $\mathbf{x}_i \in \mathbb{R}^d$, a time dimension t_i, and a 3D position $\mathbf{p}_i \in \mathbb{R}^3$. Each edge $e = (i, j) \in E_C$ connects a source node i at time t_i to a target node j at time $t_j > t_i$, with temporal gap $\Delta t_e = t_j - t_i$.

The goal is to find a solution subgraph $G = (V, E) \subseteq G_C$ that forms a valid cell lineage, i.e. a binary forest such that each node has at most one incoming edge (parent) and at most two outgoing edges (allowing for division). HOCT learns to predict a probability $p_e \in [0, 1]$ for each candidate edge $e \in E_C$, and an ILP solver (Sec. 3.3) recovers the optimal biologically valid solution.

3.1 Architecture overview

HOCT is a two-stage Transformer (Fig. 1d–e) whose attention layers encode spatial coordinates into queries and keys via 3D RoPE [52] with learnable per-head frequencies and a learnable reflection [34].

Node self-attention. Input features are projected to hidden dimension C via $\mathbf{z}_i^{(0)} = \mathbf{W}_{\text{in}}\mathbf{x}_i$, then refined through L_n self-attention layers. Each layer applies RoPE to queries and keys using the node position $\mathbf{p}_i$, and masks out node pairs farther than a distance threshold τ to preserve sparsity.

Edge attention. For each candidate edge $e = (i, j)$, we construct an edge token by concatenating the contextualized node features and projecting through an MLP:

$$\mathbf{h}_e^{(0)} = \mathrm{MLP}_{\mathrm{gather}}\big([\mathbf{z}_i^{(L_n)} \| \mathbf{z}_j^{(L_n)}]\big) \tag{1}$$

where $[.\|.]$ is the feature concatenation operator. The edge position for RoPE is the midpoint $\mathbf{p}_e = (\mathbf{p}_i + \mathbf{p}_j)/2$. Edge tokens, $\mathbf{h}_e^{(l)}$, are then refined through L_e attention layers, with the first layer performing cross-attention with $\mathbf{f}_e$ as queries and $\mathbf{h}_e^{(0)}$ as keys/values, grounding the edge representation in the raw feature relationships; here $\mathbf{f}_e = \mathbf{W}_{\mathrm{edge}}([\mathbf{x}_i \| \mathbf{x}_j])$ are edge-feature tokens computed from the raw input features. Subsequent layers perform self-attention among edge tokens. Every edge attention layer incorporates a geometric distance bias (Sec. 3.2) based on the line-to-line distance between edge pairs in addition to RoPE.

Classification head. The final edge embeddings are passed through LayerNorm and a linear projection to produce scalar logits $\ell_e \in \mathbb{R}$, which are normalized via the parental softmax (Sec. 3.3).

3.2 Line-to-line distance encoding

RoPE encodes the position of each token (node or edge midpoint) but ignores the *geometric relationship* between edge pairs. For instance, two edges can have the same midpoint distance while passing close to each other or running parallel at a distance, encoding distinct motion patterns such as coherent displacement, collision, or position swapping.[1] To provide an explicit geometric signal, we introduce a distance bias based on the 3D line-to-line distance.

Each edge $e = (i, j)$ defines a directed line segment from $\mathbf{p}_i$ to $\mathbf{p}_j$. For any pair of edges e_m and e_n, we compute the minimum Euclidean distance between their line segments:

$$d_{\mathrm{line}}(e_m, e_n) = \min_{s,u \in [0,1]} \big\| \big(\mathbf{p}_{i_m} + s \cdot \mathbf{d}_m\big) - \big(\mathbf{p}_{i_n} + u \cdot \mathbf{d}_n\big) \big\| \tag{2}$$

where $\mathbf{d}_m = \mathbf{p}_{j_m} - \mathbf{p}_{i_m}$ is the direction vector of edge $e_m = (i_m, j_m)$. This distance is computed analytically via the parametric closest-point formula for 3D line segments, with clamping for boundary cases (Appendix A) and injected into the edge attention logits as a *per-head learnable bias*:

$$A_h(m, n) = \frac{\hat{\mathbf{q}}_{h,m}^\top \hat{\mathbf{k}}_{h,n}}{\sqrt{D}} + \sigma_h \cdot \mathrm{softplus}(\alpha_h) \cdot d_{\mathrm{line}}(e_m, e_n) \tag{3}$$

where $\hat{\mathbf{q}}, \hat{\mathbf{k}}$ are the RoPE-encoded queries and keys, d_{line} is in dataset spatial units (positions standardized per dataset), α_h is a learnable scalar per head (initialized near zero), and $\sigma_h \in \{+1, -1\}$ is a fixed sign alternating across heads. The `softplus` prevents the sign from flipping, which we observed to be unstable even with transformations that vary slowly near zero. With $\sigma_h = -1$, nearby edges receive higher attention (*attractive* heads), encouraging comparison of near-intersecting edges; with $\sigma_h = +1$, distant edges receive higher attention (*repulsive* heads), providing global context about competing trajectories. Combining both signs yields the best results.

3.3 Parental softmax and ILP inference

A biological constraint in cell tracking is that each cell has at most one parent. Trackastra [17] introduced the *parental softmax* to normalize association scores among candidate parents of the same target node, but applied it only to consecutive frames ($\Delta t = 1$). We extend it to enable long-range connections, normalizing per target node *and* temporal gap:

$$p_e = \frac{\exp(\ell_e)}{1 + \sum_{e' \in \mathcal{C}(j_e, \Delta t_e)} \exp(\ell_{e'})} \tag{4}$$

where $\mathcal{C}(j, \Delta t) = \{e' \in E_C : j_{e'} = j,\ \Delta t_{e'} = \Delta t\}$ is the set of candidate edges targeting node j at temporal gap Δt. Normalizing within each Δt group ensures that short- and long-range candidates compete only with edges of the same temporal gap, preventing competition between multiple probabilities along the same path. The constant 1 in the denominator acts as an implicit "no-parent" option: when no good match exists, all edge probabilities for that node are driven toward

[1] In 3D the line segments rarely strictly intersect; the line-to-line distance still smoothly captures their proximity.

zero. We train with focal loss [33] applied to the parental-softmax probabilities, with increased weight on division edges because they are rare and challenging (Appendix A).

During inference, an ILP solver converts predicted edge probabilities into globally consistent tracks that respect the biological constraints on cell division, appearance, and disappearance, see Appendix B for full ILP formulation. The parental softmax also enables an unexplored improvement over standard ILP formulations [62, 54], which we describe below.

Variable appearance probability. Most ILP formulations use a fixed appearance cost [17, 8, 7, 36], treating all cells as equally likely to start a new track, an unrealistic simplification: new tracks are more likely near the field-of-view boundary. We instead estimate the appearance probability from the predicted edge logits.

The constant 1 in the denominator of Eq. (4) yields a per-Δt "no-match" probability $q_{j,\Delta t}$, which we aggregate with a λ-decayed weighted average over temporal gaps to obtain a per-node appearance cost:

$$q_{j,\Delta t} = \frac{1}{1 + \sum_{e' \in \mathcal{C}(j,\Delta t)} \exp(\ell_{e'})} \tag{5}$$

Note that when $\Delta t = 1$, $q_{j,1}$ is the probability that node j has no parent in the immediately preceding frame (orphan probability). The per-temporal gap probabilities $q_{j,\Delta t}$ are then aggregated with a weighted average:

$$p_\alpha(j) = \sum_{\Delta t} w_{\Delta t} \cdot q_{j,\Delta t}, \quad w_{\Delta t} = \frac{\exp\big(-(\Delta t - 1)\lambda\big)}{\sum_{\Delta t'} \exp\big(-(\Delta t' - 1)\lambda\big)} \tag{6}$$

where $w_{\Delta t}$ gives more weight to shorter temporal gaps. This reduces the penalty for starting new tracks based on the network's predictions, without any additional loss term.

Two-pass tracklet solver. Long-range edges ($\Delta t > 1$) bridge missing detections but compete with short-range alternatives along the same path. We resolve this in two passes: (1) solve the ILP using only $\Delta t = 1$ edges to produce high-confidence tracklets; (2) aggregate each tracklet into a single meta-node and re-solve over the tracklet graph using all edges. This prevents long-range edges from overriding confident short-range associations.

4 Experiments

4.1 CTC benchmark

Architecture and training. HOCT uses C=288, H=4 heads, L_n=L_e=4 layers, and distance threshold τ=300 pixels. The model is trained on 16 datasets from the Cell Tracking Challenge (CTC) [37], spanning brightfield, fluorescence, and phase-contrast microscopy in 2D and 3D. Each dataset contains two annotated time-lapse sequences (32 sequences total); we form two cross-validation splits by assigning one sequence per dataset to each split, training on one and evaluating on the other. Data augmentation includes spatial crops, affine transformations, and feature dropout that randomly zeroes entire feature groups to encourage robustness to missing features. Full training details (optimizer, schedule, augmentation, and ILP configuration) are in Appendix A.

Metrics. We report standard CTC metrics [38, 55]: **LNK** (Linking accuracy); **BIO** (Biological metric); and **CLB** (Cell Linking Benchmark), the main ranking metric for the challenge and the average between LNK and BIO; all in $[0, 1]$ with higher being better. For the bacteria benchmark, we extract scores from [1] and report: **AOGM** (Acyclic Oriented Graph Metric, lower is better) with its decomposition into addition (AOGM-A) and deletion (AOGM-D) errors, and **Division F1**.

Topological label information. To quantify the non-homophilic claim from Sec. 1, we build the line graph $L(G_C)$ of each candidate tracking graph (where original edges become nodes that are connected when they share an endpoint) and measure label agreement on it; per-sequence statistics are in Appendix C. Across the CTC sequences, the adjusted homophily [44] on $L(G_C)$ is $\mathcal{H}_{\mathrm{adj}} = 0.01 \pm 0.04$, indistinguishable from random; for each true (GT) edge, only 29% of its co-incident edges are also true, so 71% of a correct link's line-graph neighbors carry a conflicting label. For reference, standard homophilic benchmarks have $\mathcal{H}_{\mathrm{adj}} > 0.5$ [44], where graph neural networks succeed because topology carries the labels. This explains why GNN-based edge classification [2] has

not surpassed simpler baselines, since message passing aggregates neighbors whose labels are near-random. Note that this statistic only measures first-order label agreement; directional, relation-aware, or higher-order models could still exploit structure it does not capture.

4.2 Ablations

ILP solver components. Table 1 ablates the two ILP contributions on the CTC benchmark across two cross-validation splits, both with and without long-range candidate edges ($\Delta t > 1$). The variable appearance probability is consistently beneficial, improving CLB in every paired comparison (rows 1→2, 3→4, 5→6). Long links alone degrade performance (-3.2% CLB, -5.5% BIO at row 1→3) due to competing edges along the same path but become beneficial when paired with the two-pass tracklet solver, with the full configuration reaching the best CLB (0.920) and BIO (0.858).

Table 1: Ablation of ILP solver components on the CTC benchmark. Long links require the two-pass tracklet solver: without it, candidate edges of different lengths along the same path compete in the ILP. Best results in bold, near-best underlined.

Setup			Average			Split 1			Split 2		
Long links	2-pass	Appear. Prob.	CLB	LNK	BIO	CLB	LNK	BIO	CLB	LNK	BIO
			0.909	**0.982**	0.835	0.930	**0.988**	0.871	0.888	<u>0.976</u>	0.799
		✓	0.914	**0.982**	0.845	0.932	**0.988**	0.876	0.896	**0.977**	0.815
✓			0.877	0.974	0.780	0.904	0.982	0.827	0.849	0.965	0.733
✓		✓	0.897	0.977	0.816	0.920	0.984	0.857	0.873	0.971	0.775
✓	✓		0.916	<u>0.981</u>	0.851	<u>0.938</u>	<u>0.987</u>	0.888	0.894	0.975	0.813
✓	✓	✓	**0.920**	<u>0.981</u>	**0.858**	**0.939**	<u>0.987</u>	**0.890**	**0.900**	<u>0.976</u>	**0.825**

Edge-stage aggregation mechanism. Given the line-graph heterophily measured above, any aggregator restricted to candidate-graph topology (GNN message passing or topology-restricted attention) should combine each edge with label-uncorrelated neighbors. We test this by replacing HOCT's edge-attention stage with three alternatives, holding everything else fixed (input features, Stage 1, edge gatherer, candidate graph, focal loss, training schedule, and ILP post-processing of Table 1, last row); GNN variants use eight edge-stage layers to match the transformer in parameter count.

Table 2: Edge-stage aggregation mechanism ablation: HOCT's edge-attention stage is replaced with three alternatives (Edge-Transformer: global self-attention without RoPE or distance bias; GAT [57]: graph attention on $L(G_C)$; FAGCN [5]: heterophily-aware GNN), holding Stage 1, edge gatherer, candidate graph, loss, training schedule, and ILP post-processing fixed. Each replicate round averages the two cross-validation splits; we report the mean and standard deviation across rounds (three rounds).

Method	CLB	LNK	BIO
FAGCN	0.909 ± 0.0025	0.981 ± 0.0008	0.837 ± 0.0043
Edge-Transformer	0.914 ± 0.0019	0.981 ± 0.0004	0.847 ± 0.0033
GAT	0.916 ± 0.0004	0.982 ± 0.0002	0.851 ± 0.0009
HOCT	$\mathbf{0.921 \pm 0.0011}$	$\mathbf{0.983 \pm 0.0001}$	$\mathbf{0.859 \pm 0.0022}$

Table 2 compares[2]: (i) Edge-Transformer, a geometry-free analogue of HOCT: global self-attention on edge tokens without the line-to-line distance bias; (ii) GAT [57], vanilla graph attention on the line graph $L(G_C)$, the closest analogue to topology-driven GNN tracking [2]; (iii) FAGCN [5], frequency-adaptive sign-aware aggregation as a heterophily-aware GNN baseline.

All three alternatives underperform HOCT on every metric, consistent with the heterophily analysis. The two GNN variants are limited by their input topology: aggregating over $L(G_C)$ pulls each true edge's representation toward mostly incorrect labels, even when the layer is sign-aware (FAGCN). The Edge-Transformer keeps global attention but removes the geometric biases of Sec. 3.2. Without them, CLB drops to 0.914, below GAT (0.916), while adding the line-to-line distance encoding raises

[2]The implementation of [2] did not allow training on all CTC datasets and therefore was not included.

Table 3: **Cell Tracking Challenge leaderboard (hidden test set).** Official scores from the CTC organizers on the held-out test sequences for the three primary metrics (CLB, LNK, BIO; higher is better). We list the top three submissions per dataset and metric; **HOCT (ours)** is bolded when in the top three, see text for the full list of competing methods. The *Average* row reports the official CTC overall-performance (Generalizability) score, the headline cross-dataset measure.

	CLB			LNK			BIO		
Dataset	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
Average	**0.930**	0.923 [5]	0.886 [1]	**0.985**	0.984 [5]	0.977 [1]	**0.875**	0.862 [5]	0.794 [1]
HSC	0.873 [5]	**0.845**	0.807 [3]	**0.992**	0.991 [5]	0.986 [4]	0.756 [5]	**0.699**	0.635 [3]
MuSC	**0.902**	0.894 [3]	0.889 [5]	**0.986**	0.985 [4]	0.985 [5]	**0.819**	0.806 [3]	0.794 [5]
HeLa_{DIC}	0.960 [7]	**0.943**	0.936 [1]	**0.991**	0.989 [7]	0.989 [6]	0.931 [7]	**0.895**	0.885 [1]
MSC	0.837 [5]	**0.825**	0.789 [1]	0.947 [5]	**0.936**	0.924 [4]	0.727 [5]	**0.713**	0.657 [1]
A549	**1.000**	1.000	1.000	**1.000**	1.000	1.000	**1.000**	1.000	1.000
H157	**1.000**	0.993 [5]	0.972 [1]	**1.000**	0.985 [5]	0.983 [7]	**1.000**	1.000 [5]	0.962 [1]
MDA231	**0.929**	0.921 [1]	0.916 [5]	0.958 [1]	**0.957**	0.957 [5]	**0.902**	0.883 [1]	0.875 [5]
GOWT1	0.887 [5]	**0.882**	0.849 [2]	0.986 [5]	0.982 [6]	**0.982**	0.788 [5]	**0.783**	0.723 [2]
$\text{HeLa}_{\text{Fluo}}$	**0.978**	0.974 [5]	0.960 [2]	**0.997**	0.996 [5]	0.995 [1]	**0.959**	0.951 [5]	0.926 [2]
CE	**0.950**	0.922 [5]	0.877 [1]	**0.986**	0.982 [5]	0.971 [1]	**0.913**	0.862 [5]	0.782 [1]
CHO	0.974 [1]	**0.962**	0.953 [6]	**0.993**	0.993 [1]	0.990 [6]	0.955 [1]	**0.931**	0.915 [6]
U373	0.979 [4]	0.973 [8]	0.972	0.996 [8]	0.995	0.995	0.964 [4]	0.949	0.949
PSC	**0.948**	0.927 [5]	0.901 [1]	**0.994**	0.992 [5]	0.991 [1]	**0.902**	0.862 [5]	0.812 [1]
A549_{SIM}	**1.000**	1.000	1.000	**1.000**	1.000	1.000	**1.000**	1.000	1.000
$\text{Fluo2D}_{\text{SIM+}}$	0.989 [7]	0.988 [1]	0.983 [5]	0.999 [1]	0.999 [7]	0.999 [8]	0.980 [7]	0.977 [1]	0.968 [5]
$\text{Fluo3D}_{\text{SIM+}}$	0.997 [7]	**0.993**	0.992 [5]	1.000 [7]	**0.999**	0.999 [7]	0.994 [7]	**0.988**	0.984 [7]

it to 0.921. Global attention alone is not sufficient, and the geometric bias is what places edge-centric attention above the GNN baselines. Appendices D–G give the per-dataset breakdown and probe robustness to feature removal and segmentation degradation.

4.3 Official CTC submission

We submitted HOCT to the Cell Tracking Challenge [37]; Table 3 reports the official scores computed by the organizers on the *hidden* test sequences,[3] providing an unbiased comparison against all currently submitted methods (full per-dataset scores and ranks, including the datasets where HOCT falls outside the top three, are listed in Appendix J). All entries come from a *single* HOCT model trained jointly on all 2D and 3D CTC datasets, testing generalization across diverse cell types and imaging modalities. This single model ranks first in CLB, LNK, and BIO by overall generalizability and reaches the top three on 14 of 16 datasets in at least one metric. Competing methods referenced in the table are 1) Trackastra [17], 2) Baxter Algorithm [35], 3) MAMHT [26], 4) TrackTour [14], 5) ByoTrack [48], 6) Medical SAM2 and SAM 3D [66, 59], 7) zTrack [67], and 8) MAGIK [43].

4.4 Bacteria tracking benchmark

We evaluate on the bacteria tracking benchmark from [17], which features frequent cell divisions in a dense monolayer of rod-shaped bacteria. Table 4 compares HOCT against Trackastra with various feature extractors, including powerful pre-trained visual encoders (SAM2.1, DinoV2, microSAM), in both *general* (trained on diverse data) and *specialized* (trained on bacteria) settings.

HOCT achieves the lowest AOGM (6.36 ± 1.35) using only 19-dimensional hand-crafted input features, outperforming the best Trackastra variant (general, SAM2.1: 6.53 ± 1.68) which requires a large vision foundation model. With equivalent hand-crafted features, the specialized Trackastra scores 11.25 ± 3.67. Appendix H visualizes where the remaining link errors fall, comparing HOCT against the publicly released Trackastra checkpoints on identical detections.

[3]The CTC organizers have not yet refreshed the public leaderboard as of 2026-11-02; the scores reported here were provided to us directly.

Table 4: **Bacteria tracking benchmark.** HOCT achieves the best AOGM with zero deletion errors using only geometric features, even against Trackastra variants that rely on pre-trained visual encoders (SAM2.1, DinoV2, microSAM). Baselines from [1].

		Model	AOGM↓	AOGM-A↓	AOGM-D↓	Division F1↑
General	Trackastra	SAM2.1	6.53 ± 1.68	6.53 ± 1.68	0.00	$\mathbf{0.9981 \pm 0.0005}$
		CoTracker3	17.67 ± 7.08	16.56 ± 6.33	1.11 ± 1.92	0.9914 ± 0.0054
		Public weights [18]	49.58	37.92	11.67	0.9752
		features	12.31 ± 5.43	11.19 ± 4.46	1.11 ± 1.92	0.9957 ± 0.0019
		features-DO	13.67 ± 4.86	13.67 ± 4.86	0.00	0.9945 ± 0.0022
Specialized	Trackastra	CoTracker3	20.81 ± 5.41	15.81 ± 2.93	5.00 ± 3.59	0.9899 ± 0.0026
		microSAM	11.92 ± 4.06	10.81 ± 3.21	1.11 ± 0.96	0.9960 ± 0.0013
		DinoV2	14.31 ± 8.02	12.64 ± 8.02	1.67	0.9938 ± 0.0047
		SAM	14.81 ± 5.69	12.03 ± 2.41	2.78 ± 3.85	0.9958 ± 0.0014
		SAM2.1	7.00 ± 2.14	6.44 ± 1.78	0.56 ± 0.96	0.9977 ± 0.0010
		Original (greedy) [17]	29.0	–	–	–
		Original (ILP) [17]	19.0	–	–	–
		features	21.97 ± 8.53	18.64 ± 6.84	3.33 ± 4.55	0.9913 ± 0.0057
		features-DO	11.25 ± 3.67	9.03 ± 2.75	2.22 ± 1.92	0.9963 ± 0.0019
		HOCT	$\mathbf{6.36 \pm 1.35}$	$\mathbf{6.36 \pm 1.35}$	**0.00**	0.9962 ± 0.0012

4.5 Incremental correction

As a final experiment, to assess the argument that edge-embedding is a better paradigm, we evaluate how efficiently each system can be improved with limited user feedback [45] in an *incremental correction* setting on the three bacteria validation sequences from Sec. 4.4. We compare against Attrackt [31], which builds upon Trackastra, proposing an unsupervised pre-training for tracking and human-in-the-loop fine-tuning. Both HOCT and Trackastra + Attrackt [17, 31] start from their respective CTC-pretrained weights (never trained on bacteria training data) and run an independent correction loop on each sequence: 20 rounds of 20 annotations each (400 per sequence). For both systems, annotations act only through the updated model; they are never imposed as hard constraints in the ILP. HOCT annotates candidate *edges* and refits a logistic regression head on frozen edge embeddings; Trackastra annotates *nodes* and, in its strongest variants developed exclusively for human-in-the-loop training [29], it fine-tunes the network via LoRA [23]. A linear classifier head cannot be applied to Trackastra, because it encodes associations as dot-product similarity between node embeddings; the closest analogue is a *linear probe* that learns a single map L applied to both embeddings before the dot product, replacing $\langle \mathbf{n}_i, \mathbf{n}_j \rangle$ with $\langle L\mathbf{n}_i, L\mathbf{n}_j \rangle$, which we also include in the comparison. HOCT is evaluated with three sampling strategies and Trackastra with three model-update variants, all using Attrackt's confidence-based node sampling (Table 5); full details are in Appendix I.

Results. Table 5 and Fig. 2 compare both systems across all strategies. HOCT starts from a better baseline (AOGM 1418 vs. 4951) and improves much more rapidly, even though Trackastra has more room for improvement. The best HOCT strategy (*uncertainty*) reduces AOGM by 59% after 400 annotations, while the best Trackastra+Attrackt strategy (*LoRA + unsup*) achieves only a 6.3% reduction despite using LoRA fine-tuning with 2000 gradient steps per round. The full HOCT round (logistic refit + inference + ILP + metrics on all three sequences) takes $\sim$2 min on an RTX 3090 and is ILP-dominated. This is well within the latency budget for active learning, whereas LoRA fine-tuning requires $\sim$40–55 min per round.

Because the two systems start from very different baselines (HOCT: 1418; Trackastra: 4951), Fig. 2 normalizes each curve to its round-0 value. HOCT's gains concentrate in the early rounds: the first 80 annotations account for roughly half the total reduction, and active strategies outperform random sampling. Trackastra + Attrackt curves remain near-flat: even LoRA + unsup, which updates $\sim 5\%$ of model weights per round, reduces AOGM by only 6.3%. This suggests that explicit per-edge features enable cheaper and more effective correction than implicit node-similarity representations.

Table 5: Incremental correction on the bacteria validation sequences. Both systems annotate 20 elements per round for 20 rounds in an independent loop per sequence (400 per sequence). AOGM is averaged over three sequences (lower is better). Δ denotes relative improvement from round 0. HOCT uses a frozen backbone with a refitted linear head; Trackastra strategies use LoRA fine-tuning or a learned linear map of the embeddings, with confidence-based node sampling. *LoRA + unsup* requires pretrained per-detection autoencoder embeddings and `MLP_E`/`MLP_D` weights (Appendix I); we pretrain both directly on the validation sequences before round 0.

System	Strategy	AOGM_0	AOGM_{20}	Δ (%)
HOCT	Random	1418	796	-43.8
	Uncertainty	**1418**	**587**	$\mathbf{-58.6}$
	Solution distance	1418	619	-56.3
Trackastra + [31]	Linear probe	4951	4926	-0.5
	LoRA (sup)	4951	4670	-5.7
	LoRA + unsup	**4951**	**4639**	$\mathbf{-6.3}$

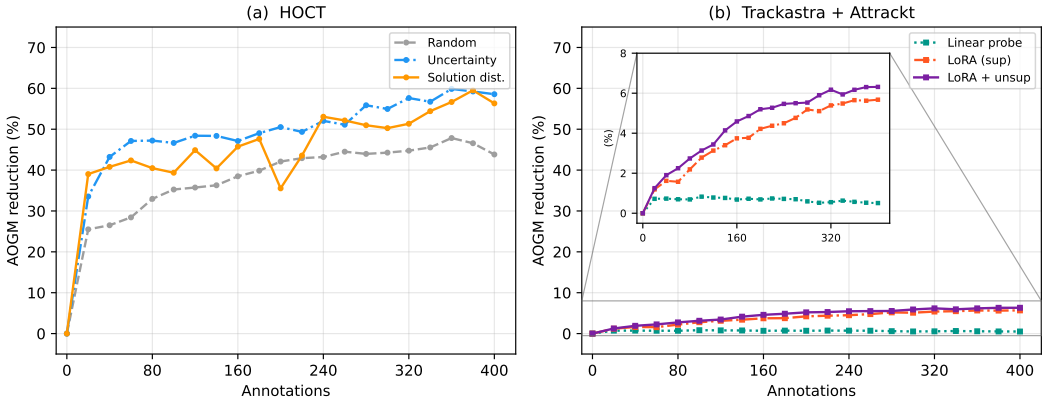


Figure 2: **Relative AOGM reduction during incremental correction on bacteria validation sequences.** Both panels show the percentage AOGM reduction from each system's round-0 baseline, enabling direct comparison of correction efficiency despite different starting points (HOCT: $\sim$1400; Trackastra: $\sim$4950). (a) HOCT strategies reduce AOGM by up to 59% within 400 annotations, with *uncertainty* sampling achieving the steepest gains. (b) Trackastra + Attrackt [31] strategies achieve at most 6.3% reduction despite LoRA fine-tuning; the inset zooms into the 0–8% range to resolve strategy differences. *LoRA + unsup* is the strongest Attrackt variant on average once its image autoencoder embeddings and `MLP_E`/`MLP_D` weights are pretrained on the validation data (Appendix I); without that pretraining, the unsupervised reconstruction loss is numerically dominant and degrades performance. The *linear probe* (a single learned matrix applied to both embeddings before the dot product) stays below 1% (-0.5%), far below LoRA fine-tuning.

5 Conclusion

HOCT is an edge-centric Transformer that classifies candidate cell links via attention biased by inter-link geometry, addressing both the connected-manifold problem at divisions and the near-random topological label agreement in candidate tracking graphs. With only simple geometric features and no pre-trained image encoder, it matches or surpasses prior methods across the Cell Tracking Challenge and a bacteria-division benchmark; a logistic head on frozen HOCT features further outperforms LoRA fine-tuning of a Trackastra baseline in a 400-annotation human-in-the-loop setting.

Limitations. Edge attention memory cost is $O(|E|^2)$, compared with $O(|V|^2)$ for node-association Transformers; with the k-nearest-neighbor candidate graph ($|E| \approx k|V|$) this is $O(k^2|V|^2)$, further reduced in practice by the spatial mask. The model is compact (C=288, 2×4 layers), so inference on all public CTC sequences runs untiled on a 24 GB GPU (Appendix A). Very large fields of view require tiling, which is implemented but evaluating it is beyond the scope of this work. HOCT also

inherits the quality of its inputs: segmentation errors propagate to the tracks (Appendix G), and a true association missing from the candidate graph cannot be recovered. Finally, our conclusions hold for the settings evaluated, which are limited due to the lack of ground-truth tracking datasets.

Broader impact. Accurate cell tracking advances biological research by enabling quantitative analysis of cell behavior in development, regeneration, and disease models. Potential downstream applications include drug screening, cancer-metastasis assays, and developmental-biology studies. As with other biological tools, there is a risk of misuse in unethical research contexts, though we expect the primary impact to be beneficial.

Acknowledgments and Disclosure of Funding

We are grateful to Teun Huijben and Thibaut Goldsborough for the discussion and feedback, and Bernhard Magerl for testing the HOCT model. This work was funded by the Biohub San Francisco. We are deeply grateful to Biohub donors Priscilla Chan and Mark Zuckerberg for their generous support, which made this research possible.

References

[1] Cyril Achard. Leveraging learned object features for transformer-based cell tracking in microscopy. Master's thesis, École Polytechnique Fédérale de Lausanne, 2025.

[2] Tal Ben-Haim and Tammy Riklin Raviv. Graph neural network for cell tracking in microscopy videos. In *European Conference on Computer Vision*, pages 610–626. Springer, 2022.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *IEEE International Conference on Computer Vision*, pages 941–951, 2019.

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing*, pages 3464–3468. IEEE, 2016.

[5] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 3950–3957, 2021.

[6] Matthew B Bouchard, Venkatakaushik Voleti, César S Mendes, Clay Lacefield, Wesley B Grueber, Richard S Mann, Randy M Bruno, and Elizabeth MC Hillman. Swept confocally-aligned planar excitation (SCAPE) microscopy for high-speed volumetric imaging of behaving organisms. *Nature Photonics*, 9(2):113–119, 2015.

[7] Jordão Bragantini, Merlin Lange, and Loïc Royer. Large-scale multi-hypotheses cell tracking using ultrametric contours maps. In *European Conference on Computer Vision*, 2024.

[8] Jordão Bragantini, Ilan Theodoro, Xiang Zhao, Teun APM Huijben, Eduardo Hirata-Miyasaki, Shruthi VijayKumar, Akilandeswari Balasubramanian, Tiger Lao, Richa Agrawal, Sheng Xiao, et al. Ultrack: pushing the limits of cell tracking across biological scales. *Nature Methods*, pages 1–14, 2025.

[9] Guillem Braso and Laura Leal-Taixe. Learning a neural solver for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6246–6256. IEEE, 2020. doi: 10.1109/cvpr42600.2020.00628.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[11] Çağrı Çevrim, Béryl Laplace-Builhé, Ko Sugawara, Maria Lorenza Rusciano, Nicolas Labert, Jacques Brocard, Alba Almazán, and Michalis Averof. Long-term live imaging, cell identification and cell tracking in regenerating crustacean legs. *eLife*, 14:RP107534, 2025.

[12] Bi-Chang Chen, Wesley R Legant, Kai Wang, Lin Shao, Daniel E Milkie, Michael W Davidson, Chris Janetopoulos, Xufeng S Wu, John A Hammer III, Zhe Liu, et al. Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science*, 346 (6208):1257998, 2014.

[13] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[14] Draga Doncila Pop. TrackTour, 2025. URL `https://github.com/DragaDoncila/tracktour`.

[15] David Feldman, Avtar Singh, Jonathan L Schmid-Burgk, Rebecca J Carlson, Anja Mezger, Anthony J Garrity, Feng Zhang, and Paul C Blainey. Optical pooled screens in human cells. *Cell*, 179(3):787–799, 2019.

[16] Benjamin Gallusser. divisualisation: a napari plugin to visualise cell-tracking errors, 2026. URL `https://github.com/bentaculum/divisualisation`.

[17] Benjamin Gallusser and Martin Weigert. Trackastra: Transformer-based cell tracking for live-cell microscopy. In *European Conference on Computer Vision*, 2024.

[18] Benjamin Gallusser and Martin Weigert. Trackastra: code and pretrained weights, 2024. URL `https://github.com/weigertlab/trackastra`.

[19] Haejun Han and Hang Lu. ASCENT: Annotation-free self-supervised contrastive embeddings for 3d neuron tracking in fluorescence microscopy. In *IEEE International Conference on Computer Vision*, pages 14676–14687, 2025.

[20] Junya Hayashida, Kazuya Nishimura, and Ryoma Bise. Mpm: Joint representation of motion and position map for cell tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3823–3832, 2020.

[21] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[22] Robert Hooke. Micrographia. In *Literature and Science, 1660-1834, Part II vol 5*, pages 1–14. Routledge, 1665.

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[24] Khuloud Jaqaman, Dinah Loerke, Marcel Mettlen, Hirotaka Kuwata, Sergio Grinstein, Sandra L Schmid, and Gaudenz Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, 5(8):695–702, 2008.

[25] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

[26] Timo Kaiser, Maximilian Schier, and Bodo Rosenhahn. Cell tracking according to biological needs-strong mitosis-aware multi-hypothesis tracker with aleatoric uncertainty. *IEEE Transactions on Medical Imaging*, 2025.

[27] Philipp J Keller, Annette D Schmidt, Joachim Wittbrodt, and Ernst HK Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322 (5904):1065–1069, 2008.

[28] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[29] Manan Lalit and Jan Funke. Attrackt, 2025. URL `https://github.com/funkelab/attrackt`.

[30] Manan Lalit and Jan Funke. Attrackt experiments, 2025. URL `https://github.com/funkelab/attrackt_experiments`.

[31] Manan Lalit and Jan Funke. An investigation of unsupervised cell tracking and interactive fine-tuning. In *IEEE International Conference on Computer Vision Workshops*, pages 5792–5800, 2025.

[32] Merlin Lange, Alejandro Granados, Shruthi VijayKumar, Jordão Bragantini, Sarah Ancheta, Yang-Joon Kim, Sreejith Santhosh, Michael Borja, Hirofumi Kobayashi, Erin McGeever, et al. A multimodal zebrafish developmental atlas reveals the state-transition dynamics of late-vertebrate pluripotent axial progenitors. *Cell*, 187(23):6742–6759, 2024.

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[34] Haiping Liu, Lijing Lin, Jingyuan Sun, Zhegong Shangguan, Mauricio A Alvarez, and Hongpeng Zhou. Rethinking rope: A mathematical blueprint for n-dimensional positional embedding. *arXiv preprint arXiv:2504.06308*, 2025.

[35] Klas EG Magnusson. *Segmentation and tracking of cells and particles in time-lapse microscopy*. PhD thesis, KTH Royal Institute of Technology, 2016.

[36] Caroline Malin-Mayor, Peter Hirsch, Leo Guignard, Katie McDole, Yinan Wan, William C Lemon, Dagmar Kainmueller, Philipp J Keller, Stephan Preibisch, and Jan Funke. Automated reconstruction of whole-embryo cell lineages by learning from sparse annotations. *Nature Biotechnology*, pages 1–6, 2022.

[37] Martin Maška, Vladimír Ulman, Pablo Delgado-Rodriguez, Estibaliz Gómez-de Mariscal, Tereza Nečasová, Fidel A Guerrero Peña, Tsang Ing Ren, Elliot M Meyerowitz, Tim Scherr, Katharina Löffler, et al. The cell tracking challenge: 10 years of objective benchmarking. *Nature Methods*, pages 1–11, 2023.

[38] Pavel Matula, Martin Maška, Dmitry V Sorokin, Petr Matula, Carlos Ortiz-de Solórzano, and Michal Kozubek. Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. *PloS one*, 10(12):e0144959, 2015.

[39] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[40] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022.

[41] Marius Pachitariu, Michael Rariden, and Carsen Stringer. Cellpose-sam: superhuman generalization for cellular segmentation. *bioRxiv*, pages 2025–04, 2025.

[42] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 164–173, 2021.

[43] Jesús Pineda, Benjamin Midtvedt, Harshith Bachimanchi, Sergio Noé, Daniel Midtvedt, Giovanni Volpe, and Carlo Manzo. Geometric deep learning reveals the spatiotemporal features of microscopic motion. *Nature Machine Intelligence*, 5(1):71–82, 2023.

[44] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023.

[45] Draga Doncila Pop, Pierre Le Bodic, and Juan Nunez-Iglesias. Needles in the haystack-rapid error detection for cell tracking solutions. In *Int. Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2025.

[46] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

[47] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

[48] Raphael Reme, Alasdair Newson, Elsa Angelini, Jean-Christophe Olivo-Marin, and Thibault Lagache. Particle tracking in biological images with optical-flow enhanced kalman filtering. In *Int. Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2024.

[49] Benedek Rozemberczki, Peter Englert, Amol Kapoor, Martin Blais, and Bryan Perozzi. Pathfinder discovery networks for neural message passing. In *Proceedings of the Web Conference 2021*, pages 2547–2558, 2021.

[50] Martin Schiegg, Philipp Hanslovsky, Bernhard X Kausler, Lars Hufnagel, and Fred A Hamprecht. Conservation tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2928–2935, 2013.

[51] Yu Shi, Jimmy S Tabet, Daniel E Milkie, Timothy A Daugird, Chelsea Q Yang, Alex T Ritter, Andrea Giovannucci, and Wesley R Legant. Smart lattice light-sheet microscopy for imaging rare and complex cellular events. *Nature Methods*, pages 1–10, 2024.

[52] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[53] Ko Sugawara, Çağrı Çevrim, and Michalis Averof. Tracking cell lineages in 3d by incremental deep learning. *eLife*, 11:e69380, 2022.

[54] Engin Türetken, Xinchao Wang, Carlos J Becker, Carsten Haubold, and Pascal Fua. Network flow integer programming to track elliptical cells in time-lapse sequences. *IEEE Transactions on Medical Imaging*, 36(4):942–951, 2016.

[55] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14(12):1141–1152, 2017.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[57] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[58] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):583–598, 1991.

[59] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, et al. Sam-med3d: a vision foundation model for general-purpose segmentation on volumetric medical images. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[60] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing*, pages 3645–3649. IEEE, 2017.

[61] Bin Yang, Merlin Lange, Alfred Millett-Sikking, Xiang Zhao, Jordão Bragantini, Shruthi Vijay Kumar, Mason Kamb, Rafael Gómez-Sjöberg, Ahmet Can Solak, Wanpeng Wang, et al. Daxi—high-resolution, large imaging volume and multi-view single-objective light-sheet microscopy. *Nature Methods*, 19(4):461–469, 2022.

[62] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[63] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022.

[64] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.

[65] Hongyu Zhou, Seonghoon Kim, Zhifeng Zhao, Jiaqi Fan, Wen Huang, Xinghua Sui, Lizhi Shao, Haoran An, Jing-Ren Zhang, Jiamin Wu, et al. CELLECT: contrastive embedding learning for large-scale efficient cell tracking. *Nature Methods*, pages 1–12, 2025.

[66] Jiayuan Zhu, Abdullah Hamdi, Yunli Qi, Yueming Jin, and Junde Wu. Medical sam 2: Segment medical images as video via segment anything model 2. *arXiv preprint arXiv:2408.00874*, 2024.

[67] Zi Lab. zTrack4CTC, 2024. URL `https://github.com/Zi-Lab/zTrack4CTC`.

A Implementation details

Architecture. The full architecture specifications are: input dimension d=19, hidden dimension C=288, H=4 attention heads (head dimension D=72), MLP expansion factor 2 (hidden MLP dimension 576), attention dropout 0.05, MLP dropout 0.0, gated attention [46] in all layers. The node attention stage uses L_n=4 layers without distance biases (only RoPE and spatial masking). The edge attention stage uses L_e=4 layers with both attractive and repulsive distance biases ($\sigma_h \in \{+1, -1\}$), with independent bias parameters per head. The distance bias scalar α_h is initialized to -5 per head (so $\mathrm{softplus}(-5) \approx 0.007$, starting near zero). The edge gatherer uses concatenation mode: $\mathrm{MLP}_{\mathrm{gather}} : \mathbb{R}^{2C} \to \mathbb{R}^{C}$ (two linear layers with GELU). The first edge-attention layer performs cross-attention with raw edge features $\mathbf{f}_e = \mathbf{W}_{\mathrm{edge}}([\mathbf{x}_i \| \mathbf{x}_j])$ as queries and $\mathbf{h}_e^{(0)}$ as keys/values, grounding edge representations in raw feature relationships; subsequent layers perform self-attention among edge tokens. The classification head is LayerNorm(C) $\to$ Linear(C, 1).

Input features. Each node i carries a d=19-dimensional feature vector $\mathbf{x}_i$ derived from the segmentation mask: spatiotemporal position (t, z, y, x), equivalent diameter, intensity statistics (min, max, mean, standard deviation), the 3×3 inertia tensor (9 values), and the distance to the nearest field-of-view border. All features are standardized per dataset.

RoPE configuration. Each RoPE instance has learnable frequencies and learnable Householder reflection vectors, with independent parameters per head. With D=72 and d_{pos}=3, each head has $D/(2 \cdot d_{\mathrm{pos}}) = 12$ frequency bands.

Optimizer. We use a hybrid Muon–Adam optimizer. Muon (MomentUm Orthogonalized by Newton-Schulz) [25] is used for 2D weight matrices in hidden layers (lr=0.015, momentum=0.85). Adam is applied to biases, norms, input/output projections (lr=0.0017), and positional encoding parameters (lr=0.007). Weight decay is 1.5×10^{-5}. The learning rate schedule is cosine annealing from 100 warmup steps to 50k total steps, with minimum lr 5.5×10^{-5}. EMA with decay 0.98 is maintained throughout training.

Data augmentation. Training augmentations: random spatial crop (256×512×512), 3D affine (rotation ±180°, scale 0.75–1.25, shear ±0.1), random offset (±3), axis flips (p=0.5), intensity power augmentation (exponent 0.75–1.25), and feature dropout (p=0.2 per feature group). Validation uses only spatial cropping and standardization.

Datasets. Training uses 16 CTC datasets with per-dataset sampling weights (range 1.0–4.0) to balance dataset sizes. Training uses sequence 01, validation uses sequence 02 for each dataset. Datasets span brightfield (BF-C2DL-HSC, BF-C2DL-MuSC), DIC (DIC-C2DH-HeLa), fluorescence (Fluo-C2DL-MSC, Fluo-C3DH-A549, Fluo-C3DH-A549-SIM, Fluo-C3DH-H157, Fluo-C3DL-MDA231,

Fluo-N2DH-GOWT1, Fluo-N2DH-SIM+, Fluo-N2DL-HeLa, Fluo-N3DH-CE, Fluo-N3DH-CHO, Fluo-N3DH-SIM+), and phase contrast (PhC-C2DH-U373, PhC-C2DL-PSC) microscopy.

Loss. Focal loss with γ=3.5, no positive class reweighting ($\mathrm{pos_weight}$=1.0), division weight 3.5×, global averaging (not per-batch).

ILP configuration. Temporal decay λ=0.5, appearance weight 0.5, disappearance weight 0.25, division weight 0.25, node weight -10.0 (strong incentive to include nodes), test-time augmentation with 6 random augmentations. The ILP parameters were found by using half of the datasets from CTC for training and the other half for validation.

Compute resources. HOCT models were trained on a single NVIDIA H200 GPU node (1 GPU, 15 CPU cores, 128 GB RAM); one full run (50k steps over all 16 datasets jointly) takes ∼18 h with peak GPU memory ∼60 GB. This is the cost per cross-validation split or per ablation row, not per dataset. The incremental correction experiments (Sec. 4.5) were run on a desktop with two NVIDIA RTX 3090 GPUs (16 CPU cores, 128 GB RAM); per-round wall-clocks are listed in Appendix I.

Untiled inference scaling. With at most k candidate links per detection, $|E| \approx k|V|$, so edge attention costs $O(k^2|V|^2)$ compared with $O(|V|^2)$ node attention. Table 6 measures the resulting practical range on both public sequences of all 16 CTC linking datasets. We time only the model-inference loop: temporal-window materialization, host-to-device copies, the TorchScript forward pass, masking and exponentiation, and output copies to CPU. Candidate-graph loading or construction, post-loop aggregation, ILP solving, gap filling, and serialization are excluded. Each sequence is warmed up once and then measured three times with CUDA synchronization; the table reports the conservative maximum across its two sequences. Inference uses bfloat16 autocast on one NVIDIA RTX 3090 (24 GB), no test-time augmentation, and no spatial tiling. Peak allocated and reserved memory are measured with `torch.cuda.max_memory_allocated` and `torch.cuda.max_memory_reserved`, respectively. The active dataset configurations use three neighbors per preceding frame for GOWT1, CHO, and PSC and five for the other datasets; GOWT1 and CHO use 16-frame windows and the remainder use five-frame windows. All 32 sequences complete without tiling. The largest case, PSC, reaches 650 detections in one frame and 19,473 candidate edges in one temporal window, using 14.32 GiB allocated (18.43 GiB reserved). Thus the current CTC benchmark fits on a 24 GB device, but the narrow worst-case headroom does not remove the quadratic scaling limit; larger candidate graphs require tiling, whose effect on tracking consistency remains untested.

Table 6: **Untiled HOCT network-inference cost by CTC dataset.** For each dataset, entries are the maximum across public sequences 01 and 02; time is first taken as the median of three runs for each sequence. "Cells/frame" gives the larger sequence mean followed by the maximum single-frame count. Time covers only the network-inference loop defined above, with test-time augmentation disabled. Standard CTC dataset abbreviations are used.

Dataset	Cells/frame	Max edges/window	Time/sequence (s)	Time/window (ms)	Peak alloc./reserved (GiB)
HSC	36.6/159	7 950	80.55	45.8	2.45/3.14
MuSC	5.2/23	1 090	31.50	23.0	0.08/0.11
$\mathrm{HeLa}_{\mathrm{DIC}}$	13.1/18	880	1.52	19.0	0.07/0.09
MSC	8.9/11	304	0.78	17.8	0.04/0.05
A549	1.0/1	10	0.42	16.3	0.03/0.04
A549-SIM	1.0/1	10	0.43	16.5	0.03/0.04
H157	4.0/5	100	0.93	16.6	0.03/0.05
MDA231	48.5/53	2 570	0.27	33.8	0.30/0.36
GOWT1	24.9/29	10 119	12.71	165.1	3.93/4.99
$\mathrm{SIM+}_{\mathrm{2D}}$	40.1/54	2 640	3.46	31.5	0.30/0.39
$\mathrm{HeLa}_{\mathrm{Fluo}}$	275.8/364	18 150	34.02	386.6	12.45/16.07
CE	122.0/362	18 070	27.21	142.4	12.34/15.76
CHO	9.8/15	4 785	3.36	43.6	0.92/1.16
$\mathrm{SIM+}_{\mathrm{3D}}$	43.2/55	2 720	3.14	31.4	0.32/0.40
U373	6.7/7	312	2.01	18.1	0.04/0.05
PSC	236.9/650	19 473	49.86	168.4	14.32/18.43

Line-to-line distance. The minimum distance between two finite line segments in $\mathbb{R}^n$ (Eq. 2) admits an analytical solution. Each segment $e_m = (i_m, j_m)$ is parameterized as $\mathbf{p}_{i_m} + s\,\mathbf{d}_m$ with $s \in [0, 1]$ and $\mathbf{d}_m = \mathbf{p}_{j_m} - \mathbf{p}_{i_m}$. For a pair of segments e_m, e_n, let $\mathbf{v} = \mathbf{p}_{i_m} - \mathbf{p}_{i_n}$ and define:

$$a = \mathbf{d}_m \cdot \mathbf{d}_m, \quad b = \mathbf{d}_m \cdot \mathbf{d}_n, \quad c = \mathbf{d}_n \cdot \mathbf{d}_n, \quad f = \mathbf{d}_m \cdot \mathbf{v}, \quad g = \mathbf{d}_n \cdot \mathbf{v} \tag{7}$$

Minimizing $\|\mathbf{v} + s\,\mathbf{d}_m - u\,\mathbf{d}_n\|^2$ over unconstrained s, u yields the linear system:

$$\begin{pmatrix} a & -b \\ -b & c \end{pmatrix} \begin{pmatrix} s \\ u \end{pmatrix} = \begin{pmatrix} -f \\ g \end{pmatrix} \tag{8}$$

For non-parallel segments ($ac - b^2 > 0$), the unconstrained solution is:

$$s^* = \frac{bg - cf}{ac - b^2}, \qquad u^* = \frac{bs^* + g}{c} \tag{9}$$

Since s, u must lie in $[0, 1]$ for finite segments, we apply iterative projection:

1. $s_0 = \mathrm{clamp}(s^*,\ 0,\ 1)$,
2. $u_0 = \mathrm{clamp}\big(\frac{b\,s_0+g}{c},\ 0,\ 1\big)$,
3. $s_1 = \mathrm{clamp}\big(\frac{b\,u_0-f}{a},\ 0,\ 1\big)$.

For nearly parallel segments ($ac - b^2 < \epsilon$), we fix $s_0 = 0$ and set $u_0 = \mathrm{clamp}(g/c,\ 0,\ 1)$. The line-to-line distance is then:

$$d_{\mathrm{line}}(e_m, e_n) = \big\|(\mathbf{p}_{i_m} + s_1\,\mathbf{d}_m) - (\mathbf{p}_{i_n} + u_0\,\mathbf{d}_n)\big\| \tag{10}$$

The computation is fully vectorized, computed only once per forward pass, and detached from the autograd graph (distances are fixed geometric constants per batch).

B ILP formulation

We formulate tracking as a binary linear program adapting the ILP formulation from [62, 54]. Candidate edges are directed forward in time, $(i, j) \in E_C \Rightarrow t_i < t_j$ (Sec. 3). For each node $i \in V_C$, we define binary variables: y_i (node is active), a_i (appearance, starts a new track), b_i (disappearance, ends a track), and δ_i (division, node has two daughters). For each candidate edge $e \in E_C$, we define a binary variable x_e. The ILP minimizes the total weighted score:

$$\min_{\mathbf{x},\mathbf{y}} \quad \sum_{e \in E_C} w_e\,x_e + \sum_{i \in V_C} \big(w_n\,y_i + w_a(i)\,a_i + w_b\,b_i + w_\delta\,\delta_i\big) \tag{11}$$

subject to flow conservation at each node:

$$y_j = a_j + \sum_{i:\,(i,j) \in E_C} x_{(i,j)} \qquad \forall\, j \in V_C \tag{12}$$

$$y_i + \delta_i = b_i + \sum_{j:\,(i,j) \in E_C} x_{(i,j)} \qquad \forall\, i \in V_C \tag{13}$$

$$y_i \geq \delta_i \qquad \forall\, i \in V_C \tag{14}$$

with $y_i, a_i, b_i, \delta_i \in \{0, 1\}$ for all $i \in V_C$ and $x_e \in \{0, 1\}$ for all $e \in E_C$. Eq. (12) ensures each active node either appears or has exactly one incoming edge (parent). Eq. (13) balances outflow: an active node either disappears or produces outgoing edges, with division allowing two. Eq. (14) ensures only active nodes can divide.

Edge weights w_e are derived from the predicted parental softmax probabilities and damped by the temporal gap,

$$w_e = (-p_e + 0.5)\,\exp\big(-\lambda(\Delta t_e - 1)\big) \tag{15}$$

so that high-confidence edges receive a negative cost (favoring selection) and longer-range candidates are progressively penalized; λ is the same temporal-decay constant used in Eq. (6). The node incentive $w_n < 0$ rewards keeping detections, so the slack variables (a_i, b_i, δ_i) are activated only when no edge configuration explains the node. In the standard formulation, the slack costs (w_a, w_b, w_δ) are fixed non-negative constants. We replace the appearance constant with a variable per-node cost

$$w_a(i) = \bar{w}_a\,\big(1 - p_\alpha(i)\big) \tag{16}$$

where $\bar{w}_a$ is the base appearance weight and $t_{\min}$ is the first frame (Sec. 3.3): cells the model confidently identifies as appearing pay no penalty ($w_a(i) = 0$ when $p_\alpha(i) = 1$), and detections in the first frame appear at zero cost. Numerical values for w_n, $\bar{w}_a$, w_b, w_δ, and λ are listed in Appendix A.

C Edge heterophily analysis

We quantify the topological label information in candidate tracking graphs by analyzing their line graphs. Given a candidate graph $G_C = (V_C, E_C)$, we construct the line graph $L(G_C)$: each edge $e \in E_C$ becomes a node in $L(G_C)$, and two nodes in $L(G_C)$ are connected whenever the corresponding edges in G_C share an endpoint. Each line-graph node inherits the binary ground-truth label of its original edge (positive if the link is in the GT lineage, negative otherwise). Let E_L denote the edge set of $L(G_C)$.

We report three metrics:

Edge homophily ratio [64]. The fraction of line-graph edges whose two endpoints carry the same label:

$$\mathcal{H}_{\text{edge}} = \frac{|\{(e_m, e_n) \in E_L : y_{e_m} = y_{e_n}\}|}{|E_L|} \tag{17}$$

This metric is sensitive to class imbalance: when most edges are negative (as in tracking, with positive rate $\sim 20\%$), negative–negative agreement inflates $\mathcal{H}_{\text{edge}}$ even under random labeling.

Adjusted homophily [44]. Corrects $\mathcal{H}_{\text{edge}}$ by subtracting its expected value under random label assignment:

$$\mathcal{H}_{\text{adj}} = \frac{\mathcal{H}_{\text{edge}} - \sum_c (D_c/2|E_L|)^2}{1 - \sum_c (D_c/2|E_L|)^2} \tag{18}$$

where D_c is the sum of degrees of line-graph nodes with label c. $\mathcal{H}_{\text{adj}} = 0$ indicates random-level agreement; positive values indicate homophily; negative values indicate heterophily.

Positive-neighbor ratio. For each true (GT) edge, the fraction of its co-incident edges that are also true. This directly measures how isolated correct links are among their graph neighbors.

Results. Across the 28 non-degenerate CTC sequences (excluding 4 sequences where tracking is trivial and every candidate edge is correct), we obtain: $\mathcal{H}_{\text{edge}} = 0.645 \pm 0.051$, $\mathcal{H}_{\text{adj}} = 0.012 \pm 0.044$, positive-neighbor ratio $= 0.29 \pm 0.15$. The near-zero adjusted homophily means that first-order label agreement on $L(G_C)$ is at chance level: the label of an edge does not predict the labels of its topological neighbors. For each true link, 71% of its graph neighbors carry a conflicting label, so topology-driven aggregation (GNN message passing or RoPE-biased attention) dilutes the signal of correct associations. For reference, standard homophilic benchmarks (Cora, CiteSeer) have $\mathcal{H}_{\text{adj}} > 0.5$ [44], where message passing is effective because topology is informative. This makes candidate tracking graphs a difficult case for methods that rely on first-order label agreement, such as the GNN baselines of Table 2. The statistic does not, however, rule out directional, relation-aware, or higher-order models that exploit structure it does not capture.

D Per-dataset edge-stage ablation

Table 7 breaks down the CLB results of Table 2 by CTC dataset. Each entry is the mean over the replicate runs and both cross-validation splits used in the aggregate comparison. Results vary across datasets: HOCT obtains the highest displayed score on twelve datasets (including ties), while the aggregate gain is largest on BF-C2DL-HSC and Fluo-C3DL-MDA231.

E Embedding-space analysis

In Sec. 1 we argue that node-token trackers place the detections of a lineage on a connected manifold, so a division does not open a gap in the embedding and the branch must be resolved by splitting a region the encoder is trained to keep together. To test this, we project the embeddings of Trackastra and HOCT on window 1199 of BF-C2DL-HSC 02, which contains two divisions, using UMAP [39] (Euclidean metric, 15 neighbors, minimum distance 0.1). Note that the two architectures embed different objects. HOCT tokenizes candidate *links*, so Fig. 3a shows one point per candidate edge, taken from the last edge-attention layer before the linear classification head and colored by the ground-truth edge label (green for a true link, red for a competing candidate). Trackastra tokenizes *detections*, so Fig. 3b,c show one point per detection, taken after the two projection heads whose

Table 7: Per-dataset CLB for the edge-stage aggregation ablation of Table 2. Values are means over replicate runs and both cross-validation splits; standard deviations are omitted. Bold marks the highest displayed value in each row.

Dataset	FAGCN	Edge-Transformer	GAT	HOCT
BF-C2DL-HSC	0.878	0.905	0.904	**0.927**
BF-C2DL-MuSC	0.791	0.801	**0.808**	0.803
DIC-C2DH-HeLa	0.866	**0.883**	0.873	**0.883**
Fluo-C2DL-MSC	0.953	0.940	**0.983**	**0.983**
Fluo-C3DH-A549	**1.000**	**1.000**	**1.000**	**1.000**
Fluo-C3DH-A549-SIM	**1.000**	**1.000**	**1.000**	**1.000**
Fluo-C3DH-H157	**0.965**	0.964	0.964	0.964
Fluo-C3DL-MDA231	0.822	0.831	0.824	**0.851**
Fluo-N2DH-GOWT1	**0.730**	**0.730**	**0.730**	**0.730**
Fluo-N2DH-SIM+	**0.986**	0.982	0.985	0.983
Fluo-N2DL-HeLa	0.949	**0.958**	0.954	**0.958**
Fluo-N3DH-CE	0.916	0.934	0.929	**0.940**
Fluo-N3DH-CHO	0.805	**0.826**	0.822	0.822
Fluo-N3DH-SIM+	0.979	0.973	0.983	**0.986**
PhC-C2DH-U373	**0.964**	**0.964**	**0.964**	**0.964**
PhC-C2DL-PSC	0.932	0.935	0.935	**0.937**
Mean	0.909	0.914	0.916	**0.921**

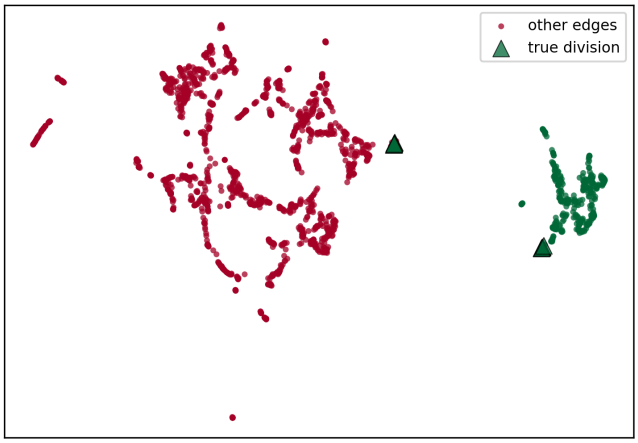
outer product forms the association matrix [17] and colored by ground-truth tracklet, i.e. the maximal simple path of the lineage forest containing it. Triangles mark ground-truth divisions.

Divisions remain connected in node space. Since UMAP does not preserve distances, we measure them in the original embedding space. Ranking all 151 other detections by distance from each dividing parent in the encoder head, the two daughters are at ranks 1 and 4 for the first division and 2 and 8 for the second, so parent and daughters remain in the same neighborhood. This neighborhood also contains detections from unrelated lineages interleaved with the daughters (zero and two for the two divisions in the encoder head, three and five in the cross-attention head). Proximity alone therefore does not distinguish the true branch from its competitors, and the decision is left to the outer product of the two heads. In HOCT's edge space, in contrast, the label is a property of the token: the two classes are already separated in the representation read by the classification head, and both division edges lie away from the competing candidates.

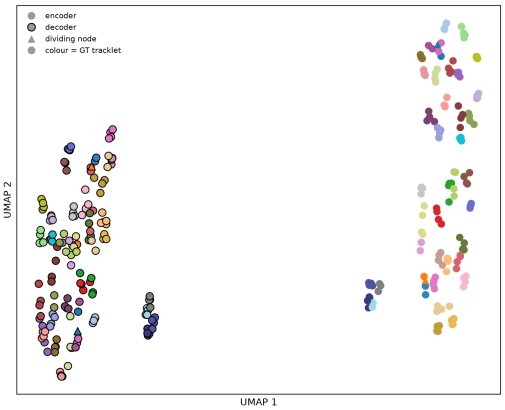
F Feature sensitivity analysis

HOCT is trained with *feature dropout* (Appendix A): with probability $p{=}0.2$ each of four non-coordinate feature groups is independently set to raw zero before standardization. Beyond leave-one-out sensitivity, we evaluate all 2^4 subsets of retained feature groups to measure their joint contribution and redundancy: equivalent diameter (D), intensity statistics (I), the $3{\times}3$ inertia tensor (T), and border distance (B). For every subset we rerun the complete pipeline with the same model, test-time augmentation, gap filling, and ILP. The $\varnothing$ condition zeros all four groups, leaving spatiotemporal coordinates (t, z, y, x) as the only node features. This is a *coordinates-only* tracker, not a feature-free one: coordinates are still used to build the candidate graph and in RoPE and the geometric attention, masks still define the detections, and images are used by the gap filler. We evaluate three 2D/3D fluorescence datasets with the jointly trained checkpoint.

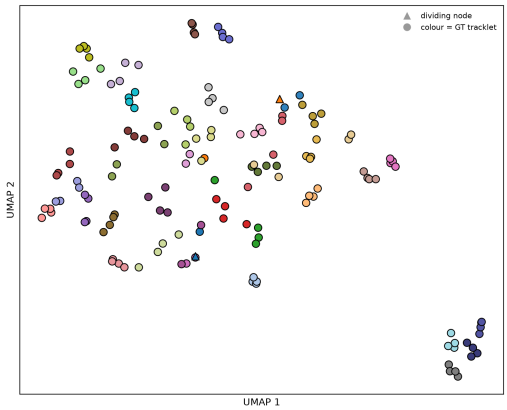
Table 8 shows that the four groups are useful but largely redundant. Relative to coordinates only, all features add 0.028 CLB on Fluo-N2DL-HeLa and 0.039 on Fluo-C3DL-MDA231. The best singleton recovers 76% and 64% of those respective gains (inertia tensor for HeLa; diameter or intensity for MDA231), while the best pair recovers 97% and 100% (diameter–intensity and intensity–inertia, respectively). Exact Shapley contributions across all subset contexts are $(D, I, T, B) = (0.0126, 0.0029, 0.0127, 0.0000)$ on HeLa and $(0.0067, 0.0164, 0.0155, 0.0000)$ on MDA231; they sum to the all-features minus coordinates-only difference. Border distance has zero marginal contribution in every subset. We included it because proximity to the field boundary could inform appearance and disappearance, but it can be removed without loss on these benchmarks; we keep it only for compatibility with the released checkpoint. Because subset scores are not monotone, a single cumulative ordering of the groups would be misleading. Finally, this is an inference-time analysis of a model trained under feature dropout, not the performance of separately retrained subset-



(a) HOCT, candidate-edge embeddings



(b) Trackastra, node embeddings, both heads



(c) Trackastra, decoder head only

Figure 3: **UMAP projections of the learned representations on window** 1199 **of BF-C2DL-HSC 02. (a)** HOCT embeds candidate links, so the ground-truth label is a property of each token. True links and competing candidates occupy separate regions of the space read by the classification head, and neither division edge (triangles) falls among the competing candidates. **(b,c)** Trackastra embeds detections. Each tracklet forms a tight cluster, but dividing detections sit inside ordinary clusters, indistinguishable from non-dividing ones, and a parent stays close to both its daughters and their competitors. In (b), the two heads occupy disjoint regions; (c) shows the cross-attention head alone, where several tracklets split into separate clusters.

specific models; the all-zero combination was in the training corruption support but occurred with probability only $0.2^4 = 0.0016$.

G Robustness to segmentation quality

HOCT consumes an upstream instance segmentation. To characterize its sensitivity to segmentation errors we degrade the input masks with a controlled per-instance perturbation and measure end-to-end tracking as a function of the resulting segmentation quality.

Perturbation model. For a severity level s, each instance mask is independently eroded or dilated in the x–y plane (sign chosen uniformly at random) by a disk of radius $r \sim \mathrm{Uniform}\{0, \ldots, s\}$. For 3D stacks the footprint has a z extent of one, avoiding morphology along the short z axis; dilations write only into background and erosions never remove an instance entirely. The set of detections per frame (their count and identities) is therefore preserved, isolating boundary and shape quality, the aspect measured by intersection-over-union, from detection recall. We report the realized segmentation quality as the mean IoU between each perturbed mask and its clean counterpart, averaged over

Table 8: Exhaustive feature sensitivity: CLB for all retained subsets of diameter (D), intensity (I), inertia tensor (T), and border distance (B). The coordinates-only row zeros all four groups. The two non-saturated datasets gain 0.028 and 0.039 CLB from all four groups, but one or two groups recover most of that gain; Fluo-C3DH-H157 is saturated throughout.

$\|S\|$	Retained set S	Fluo-N2DL-HeLa	Fluo-C3DL-MDA231	Fluo-C3DH-H157
0	$\varnothing$ (coordinates only)	0.940	0.903	1.000
1	D	0.960	0.928	1.000
	I	0.936	0.928	1.000
	T	0.961	0.922	1.000
	B	0.940	0.903	1.000
2	D, I	0.967	0.919	1.000
	D, T	0.962	0.922	1.000
	D, B	0.960	0.928	1.000
	I, T	0.967	0.942	1.000
	I, B	0.936	0.928	1.000
	T, B	0.961	0.922	1.000
3	D, I, T	0.968	0.942	1.000
	D, I, B	0.967	0.919	1.000
	D, T, B	0.962	0.922	1.000
	I, T, B	0.967	0.942	1.000
4	D, I, T, B (all)	0.968	0.942	1.000

all instances and frames; $s{=}0$ is the clean input ($\mathrm{mIoU}{=}1$). Because the centroids, size, intensity statistics, inertia tensor, and candidate-edge geometry are all recomputed from the perturbed masks, the perturbation propagates through every stage of the pipeline.

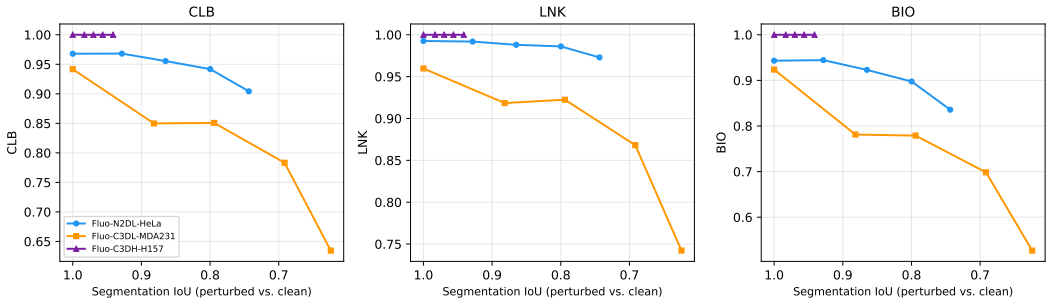


Figure 4: Tracking accuracy (CLB, LNK, BIO) as a function of input segmentation quality (mean IoU between perturbed and clean masks; clean input on the left). Per-instance masks are randomly eroded or dilated in x, y with increasing radius while the per-frame detection set is held fixed. Degradation is gradual rather than catastrophic, and linking accuracy (LNK) is at least as robust as the biological score (BIO).

Figure 4 and Table 9 show a graceful degradation under boundary error. On Fluo-N2DL-HeLa, lowering mIoU to 0.74 costs only 0.064 CLB ($0.968 \to 0.904$) and keeps LNK above 0.97. Fluo-C3DH-H157 remains perfect down to mIoU 0.94, whereas on Fluo-C3DL-MDA231 CLB declines from 0.942 to 0.634 as mIoU reaches 0.62.

Image-derived detection and topology errors. As a *tracking-by-detection* method, HOCT associates instances supplied by an upstream segmenter, and removing, duplicating, or merging those instances changes the tracking problem itself. We therefore measure how tracking degrades as input detection quality (DET) decreases, and whether HOCT's long-range skip edges recover short gaps caused by missing detections. The main benchmark already uses imperfect CTC-provided segmentations; here we widen the range of segmentation errors.

To assess the impact of image-derived segmentation errors, we segment every raw frame of the 2D nuclear-fluorescence Fluo-N2DL-HeLa sequence independently with the pretrained Cellpose-SAM `cpsam` checkpoint (Cellpose 4.0.7) [41] and with a classical watershed from minima [58]. Cellpose-SAM uses its default acceptance thresholds and no dataset-specific training. For watershed, we use a fixed pipeline with percentile normalization, Gaussian smoothing ($\sigma{=}2$), Otsu foreground extraction,

Table 9: Segmentation robustness (numbers behind Fig. 4). s is the severity (maximum in-plane disk radius in pixels); mIoU is the realized mean IoU between perturbed and clean masks.

Dataset	s	mIoU	CLB	LNK	BIO	TRA
Fluo-N2DL-HeLa	0	1.000	0.968	0.993	0.943	0.996
	1	0.928	0.968	0.992	0.944	0.995
	2	0.865	0.956	0.988	0.923	0.994
	3	0.800	0.942	0.986	0.898	0.993
	4	0.744	0.904	0.973	0.836	0.988
Fluo-C3DL-MDA231	0	1.000	0.942	0.960	0.924	0.969
	1	0.882	0.850	0.918	0.781	0.953
	2	0.794	0.851	0.922	0.779	0.953
	3	0.692	0.783	0.868	0.698	0.910
	4	0.624	0.634	0.742	0.527	0.832
Fluo-C3DH-H157	0	1.000	1.000	1.000	1.000	1.000
	1	0.983	1.000	1.000	1.000	1.000
	2	0.970	1.000	1.000	1.000	1.000
	3	0.956	1.000	1.000	1.000	1.000
	4	0.941	1.000	1.000	1.000	1.000

a minimum component size of 30 pixels, the Euclidean distance transform of the foreground mask, and a minimum distance between seeds of 12 pixels. A split or merge is counted when each participating object contributes at least 10% of its area to the overlap.

We run two tracker configurations on every mask set: an association-only ablation with gap filling disabled, and the complete HOCT pipeline. In the complete pipeline, predicted $\Delta t = 2$ skip edges identify one-frame gaps and the image-driven filler reconstructs the missing masks. This allows HOCT to bridge short false-negative gaps, but it cannot split merged detections or recombine over-segmented fragments. The provided CTC masks serve as a reference, not as ground-truth segmentation, and input DET is measured before running either tracker configuration.

Table 10: Image-derived segmentation topology and downstream tracking on Fluo-N2DL-HeLa. N/N_{GT} is the input-object count ratio, P/R is object precision/recall, and S/M gives split/merge counts under the 10% bidirectional-overlap rule. "Off" ablates skip-edge gap filling; "on" is the complete HOCT pipeline.

Input masks	N/N_{GT}	P/R	S/M	Input DET	LNK off	LNK on	CLB off	CLB on
Provided CTC	0.996	1.000/0.996	1/1	0.996	0.991	**0.993**	0.969	0.968
Cellpose-SAM	1.042	0.908/0.947	30/4	0.937	0.922	**0.935**	0.837	**0.839**
Watershed	0.851	0.881/0.765	22/122	0.747	0.707	**0.709**	0.666	**0.669**

Gap filling improves LNK, TRA, and output DET in all three conditions. For Cellpose-SAM, it raises LNK from 0.922 to 0.935, output DET from 0.937 to 0.943, and TRA from 0.935 to 0.942. For the watershed masks, LNK rises from 0.707 to 0.709 and CLB from 0.666 to 0.669. Averaged over the three inputs, gap filling adds 0.0050 LNK and 0.0016 CLB. For the provided masks, the small LNK improvement is offset by BIO decreasing from 0.946 to 0.943, resulting in a 0.0008 decrease in CLB.

With untuned Cellpose-SAM masks, input DET is 0.937 and HOCT reaches LNK 0.935, compared with 0.993 using the CTC masks. The watershed masks contain 22 splits and 122 merges, reducing input DET to 0.747 and full-pipeline CLB to 0.669. Overall, skip edges recover short missing detections, but association alone cannot separate merged instances or recombine fragmented detections.

H Qualitative comparison on bacteria

Aggregate scores do not show *where* a tracker fails. To visualize failure modes, we render the 2D+time tracks of a bacteria test sequence as a 3D *spacetime* view with `divisualisation` [16], highlighting link errors in color.

Setup. We show `trpL/150309-04`, the longest test sequence (92 frames, 6387 detections, 351 ground-truth divisions), to show a full lineage forest rather than a short excerpt. All methods consume the *same* detections (the ground-truth masks, as in the linking-only protocol of this benchmark), so detection accuracy is 1.0 for every panel and only the association differs. Errors are obtained from `traccuracy`'s CTC matcher: a *missed* link is a ground-truth edge absent from the solution, a *spurious* link is a predicted edge absent from the ground truth. As competing methods we run the two publicly released Trackastra checkpoints on this sequence ourselves: `general_2d_w_SAM2_features`, which augments the node tokens with SAM2.1 features [47] and is the strongest public variant, and the plain `general_2d` checkpoint, each with the ILP configuration shipped with that package.

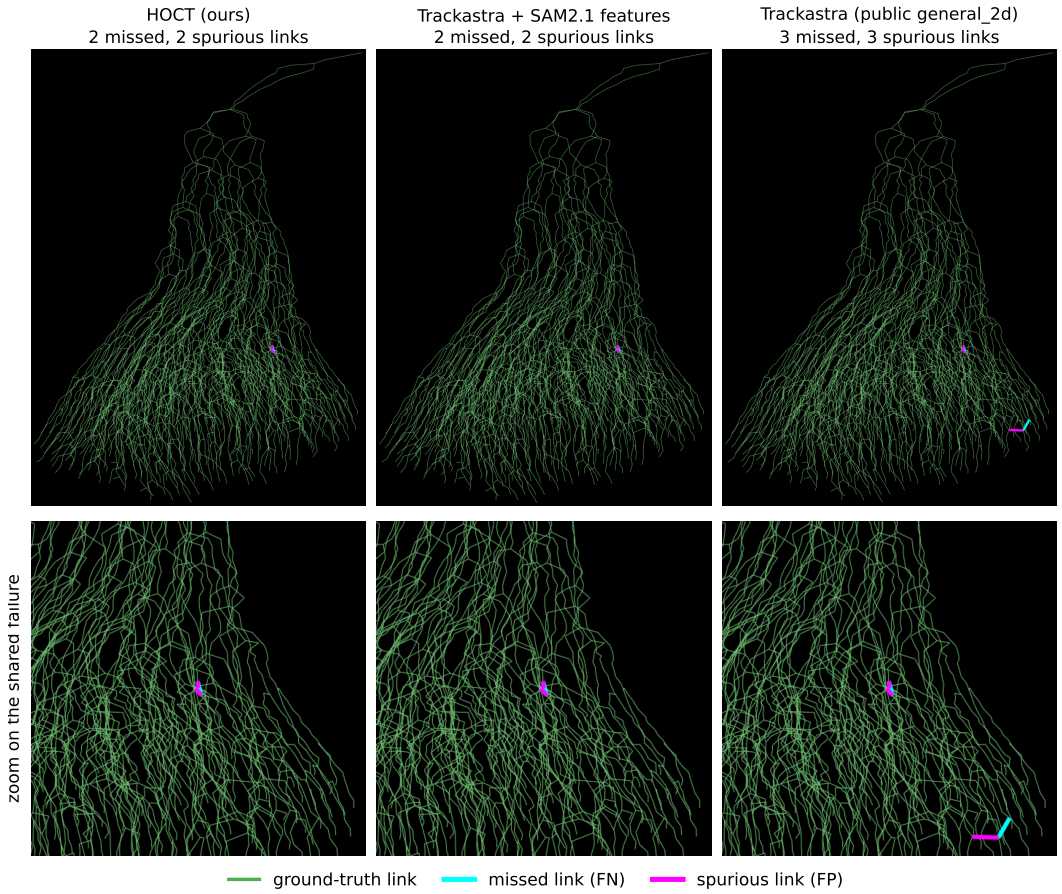


Figure 5: **Qualitative comparison on a bacteria test sequence** (`trpL/150309-04`; 92 frames, 6387 detections), lifted into spacetime with time as the vertical axis. Green traces the ground-truth lineage forest; cyan marks links the method missed and magenta links it invented. Top: the whole forest. Bottom: the same window in all three panels, centered on the errors of the methods. Out of $\sim$6300 links, HOCT (left) and the SAM2.1-feature variant (middle) each get all but two, and they fail on the *same* two: a single identity swap between two adjacent cells at one frame transition. The plain public checkpoint (right) reproduces that swap and adds a further missed link elsewhere (bottom right of the zoom) plus two wrong-semantic edges. HOCT matches the SAM2.1-feature variant here without any pretrained image encoder, using only the 19 hand-crafted features of Sec. 4.

I Incremental correction baseline details

HOCT correction mechanism. The HOCT backbone is kept frozen throughout correction. At each round, a single-layer logistic regression head ($\mathbf{w}^\top \mathbf{x} + b$) is fitted from scratch on the edge-level features extracted by the backbone for all accumulated labeled edges. The head is always re-initialized from the backbone's original classification layer (no warm-start across rounds). Training uses full-

batch L-BFGS (500 iterations) with class-balanced binary cross-entropy ($\text{pos_weight} = n_{\text{neg}}/n_{\text{pos}}$) and two regularization terms: (i) *adaptive L2*, $\lambda_{\ell_2} \cdot \frac{d}{n}\|\mathbf{w}\|^2$, which penalizes large head weights but weakens as more labels n accumulate (d is the feature dimension, λ_{ℓ_2}=1); (ii) *consistency loss*, $\lambda_c \cdot \frac{d}{n}\,\text{BCE}(\hat{y}_u, y_u^{\text{ILP}})$, which anchors predictions on *unlabeled* edges to the current ILP solution, preventing local corrections from propagating destructively (λ_c=0.25). Features are extracted with 5 test-time augmentations (overlapping temporal windows). Labeled edges are not pinned in the ILP; they influence the tracking only through the refitted head. Each sequence is corrected in an independent loop with its own labeled set and head.

The three sampling strategies select which 20 edges to label each round: *random* samples uniformly (balanced across positive/negative ILP labels); *uncertainty* selects edges closest to the 0.5 decision boundary; *solution distance* selects edges with the largest disagreement between the ILP solution and the predicted similarity, restricted to the ambiguity band $[0.15, 0.85]$.

Trackastra + Attrackt correction mechanism. The Trackastra pipeline produces per-window association matrices $A(v)$ giving the predicted probability that each candidate edge is active; like ours, an ILP solver is used to select a globally consistent tracking solution. As for HOCT, each sequence is corrected in an independent loop. At each correction round, the oracle annotates 20 *nodes* (cell detections) and provides their ground-truth outgoing link(s). Nodes are selected with Attrackt's *confidence* score $|\sum A(v) - \sum A_{\text{solver}}(v)|$, i.e. the absolute difference between the sum of predicted association weights and the number of edges selected by the ILP for that node; nodes with the highest disagreement are annotated first. As for HOCT, annotated edges are not pinned in the ILP; they are used only as supervised training targets. *Linear probe* keeps the Trackastra backbone frozen and learns a single shared linear map $L \in \mathbb{R}^{d\times d}$ (d=512, $\approx$ 262k parameters, 1.1% of the model) applied to both query and key embeddings before the outer-product association matrix, so $A_{ij} = \langle L\,\text{head}_x(\mathbf{z}_i),\, L\,\text{head}_y(\mathbf{z}_j)\rangle$. L is initialized to the identity (so round 0 is unchanged), trained with the same supervised + top-10% pseudo-supervised loss as *LoRA (sup)* (500 gradient steps per round, continuing from the previous round's L), and used at inference in place of the identity. This is the closest analogue to HOCT's logistic-head probe within Trackastra's dot-product association: only the embedding-space metric is updated, never the network weights. *LoRA (sup)* additionally fine-tunes the Trackastra transformer via LoRA [23] (rank 32, applied to attention Q/K/V/projection layers) using $\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{pseudo-sup}}$ (2000 gradient steps per round, continuing from the previous round's adapter), where the pseudo-supervised loss treats the ILP solution for the top-10% most confident nodes as soft labels. *LoRA + unsup* extends this with the full Attrackt three-component loss $\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{pseudo-sup}} + \mathcal{L}_{\text{unsup}}$; the unsupervised term is the Attrackt feature-reconstruction loss over all frames, where the autoencoder embeddings and the encoder/decoder MLPs are pretrained on the validation sequences before round 0 (see the dedicated bullet below) and the LoRA adapters then continue training jointly with the MLPs.

Trackastra baseline differences from [31]. We reproduce the interactive fine-tuning pipeline with the following differences from the original paper:

- **Starting checkpoint.** The original paper trains a Trackastra model from scratch using the Attrackt unsupervised loss, then fine-tunes interactively. We instead start from the publicly available `ctc` pretrained Trackastra checkpoint [17], which was trained on the same CTC datasets with the standard supervised loss. This represents the most accessible starting point for a practitioner.
- **Cell embeddings.** The Attrackt unsupervised loss requires per-detection embeddings from a frozen image autoencoder. We follow the paper recipe and train an autoencoder on detection-centered crops from the validation sequences, then run inference to obtain a 64-dimensional embedding per detection (full configuration in the *Pretraining for* LoRA + unsup bullet below).
- **Loss weighting.** Following the original paper, we weight all three fine-tuning loss terms equally ($\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{pseudo-sup}} + \mathcal{L}_{\text{unsup}}$) and use the top-10% most confident nodes as pseudo-labels.
- **Annotation granularity.** Each system follows its native annotation protocol: the Trackastra baseline annotates 20 *nodes* per round (as defined by Attrackt), while HOCT annotates 20 *edges* per round, in both cases per sequence. Node annotation is arguably easier because each node yields 1–2 ground-truth edges, so the Trackastra baseline receives slightly more edge-level supervision per round. This favors the baseline, making the comparison conservative for HOCT. The smallest sequence (151031-03, 417 detections over 23 frames) has only

$\sim$325 nodes with a ground-truth successor; Trackastra's node budget therefore covers every link of this sequence after 17 rounds, at which point the loop stops and the last value is carried forward when averaging over sequences.

- **Pretraining for *LoRA + unsup*.** The Attrackt unsupervised reconstruction loss (*LoRA + unsup*) requires per-detection autoencoder embeddings as targets and two small MLPs ("MLP_E"/"MLP_D") that wrap them inside the loss. The publicly released `ctc` Trackastra checkpoint ships neither, so we pretrain both directly on the bacteria validation sequences before round 0: (i) train an image autoencoder (paper recipe: 64×64 crops, four downsample stages, $4 \times 4 \times 4$ latent flattened to a 64-dim per-detection embedding) on detection-centered crops for 50k iterations; (ii) run the encoder over every detection in the validation CSV to produce the embeddings file consumed by Attrackt's data loader; (iii) pretrain "MLP_E"/"MLP_D" (each $\approx$17k parameters, d_{model}=64) against those embeddings using `common_unsupervised_step` for 10k iterations with the `ctc` backbone frozen, so the round-0 baseline is unchanged. The pretrained MLPs warm-start round 1 of the correction loop; subsequent rounds continue from the previous round's saved MLPs. Without this pretraining, the unsupervised loss is initially $\sim$1000$\times$ larger than the supervised one and dominates the gradient (the supervised signal becomes noise); with the pretraining the three losses are within an order of magnitude at the start of each round, and *LoRA + unsup* becomes the strongest Attrackt variant on average.
- **Computational cost.** All times below are full per-round wall clock on the three bacteria validation sequences combined and on the same RTX 3090, including head/adapter fit, inference (with TTA where applicable), motile ILP, and traccuracy evaluation. HOCT takes $\sim$2 min per round, dominated by the per-sequence ILP solve; the logistic head refit itself is $<$1 s. The linear probe takes $\sim$10 min per round (5–6 min fitting L + 3–4 min eight-way TTA inference + ILP), roughly $4\times$ cheaper than *LoRA (sup)* because the frozen transformer body runs under `torch.no_grad` and only L is in the autograd graph. LoRA strategies require $\sim$40 min (*LoRA (sup)*) or $\sim$55 min (*LoRA + unsup*) per round, including the same eight-way TTA. *LoRA + unsup* additionally requires a one-off pretraining pass ($\sim$30 min for the autoencoder + 10 min for the MLPs on the same GPU).
- **Code.** Our Trackastra + Attrackt baseline builds on the publicly available Attrackt library [29] and experiment code [30].

J Full CTC leaderboard scores and ranks

Table 11 reports the official CTC scores and current leaderboard ranks[4] of our generalizable HOCT model on the 16 linking challenge test datasets, extending Table 3 from Section 4.1 with the entries that fall outside the top three. Each cell shows the score followed by the rank as SCORE (RANK), where RANK is HOCT's position among the N currently submitted methods on that dataset/metric. The *Generalizability* row reports the CTC-official overall-performance score, which aggregates across datasets. The two datasets where HOCT clearly falls outside the top three are PhC-C2DH-U373 (rank 7–8) and Fluo-N2DH-SIM+ (rank 4–5): both contain few cells and few divisions per frame, so small absolute error counts let dataset-specialist methods pull ahead of a generalist model trained jointly on 16 datasets.

[4] Provided directly by the organizers; the public leaderboard had not yet been refreshed as of 2026-11-02.

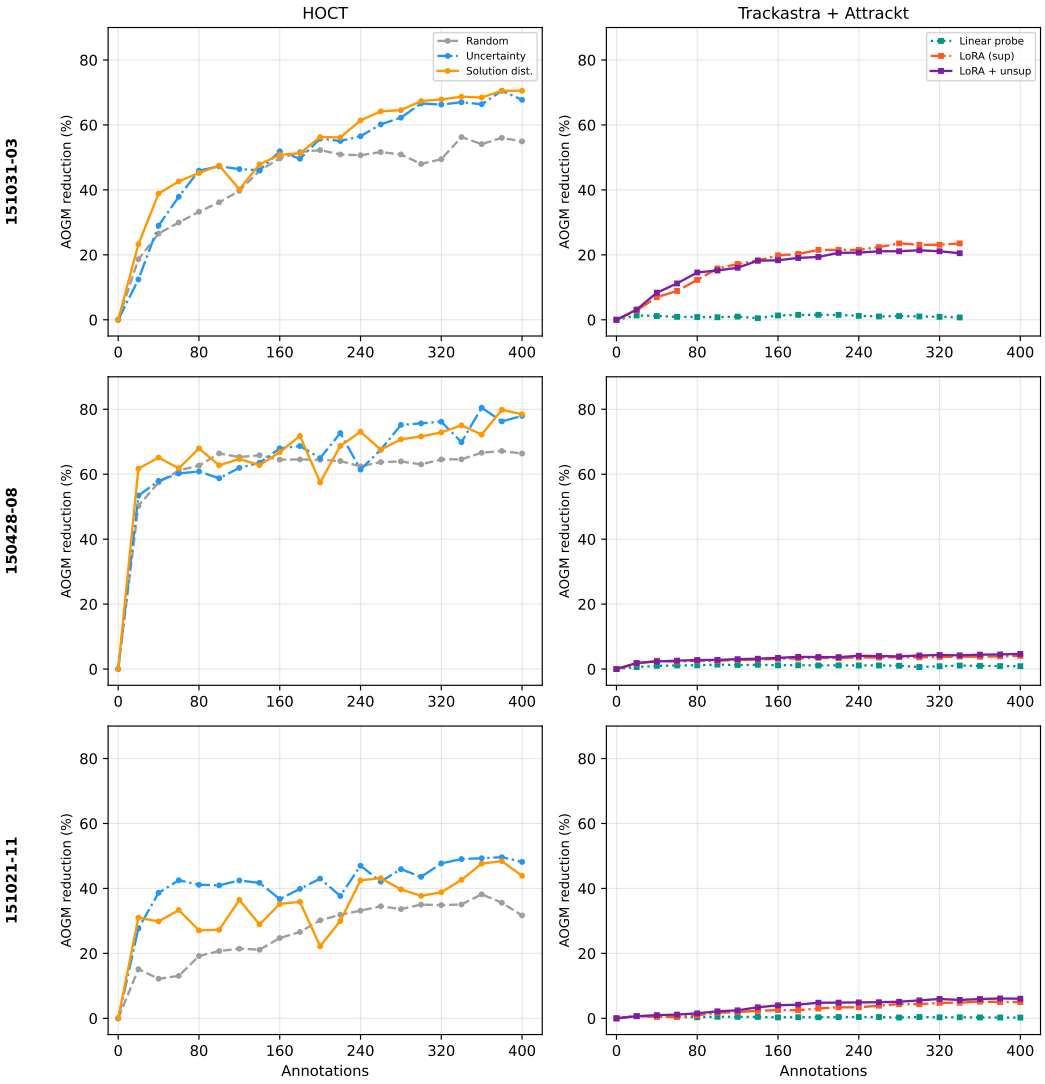


Figure 6: **Per-sequence AOGM progression during incremental correction.** Each row corresponds to one bacteria validation sequence. Left column: HOCT strategies; right column: Trackastra + Attrackt strategies. The gap is consistent across sequences: HOCT reduces AOGM by 48–78%, while the best Trackastra + Attrackt variant achieves between ∼5% (150428-08) and ∼24% (151031-03). *LoRA + unsup* is the strongest variant on the two larger sequences and *LoRA (sup)* on 151031-03, whose Trackastra curves end after 17 rounds because all ∼325 annotatable nodes have been labeled; the linear probe stays below 1% on every sequence.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Sec. 1) list four contributions — edge-centric Transformer architecture, geometric distance encoding, multi-frame parental softmax with ILP improvements, and incremental correction — each supported by corresponding experimental sections (Secs. 4.2–4.5).

   Guidelines:

Table 11: **Full CTC hidden-test results for HOCT.** Official scores and current leaderboard ranks, as SCORE (RANK), of our generalizable HOCT model trained jointly on all 2D and 3D CTC datasets.

Dataset	CLB	LNK	BIO
Generalizability	0.930 (1)	0.985 (1)	0.875 (1)
HSC	0.845 (2)	0.992 (1)	0.699 (2)
MuSC	0.902 (1)	0.986 (1)	0.819 (1)
HeLa_{DIC}	0.943 (2)	0.991 (1)	0.895 (2)
MSC	0.825 (2)	0.936 (2)	0.713 (2)
A549	1.000 (1)	1.000 (1)	1.000 (1)
H157	1.000 (1)	1.000 (1)	1.000 (1)
MDA231	0.929 (1)	0.957 (2)	0.902 (1)
GOWT1	0.882 (2)	0.982 (3)	0.783 (2)
$\text{HeLa}_{\text{Fluo}}$	0.978 (1)	0.997 (1)	0.959 (1)
CE	0.950 (1)	0.986 (1)	0.913 (1)
CHO	0.962 (2)	0.993 (1)	0.931 (2)
U373	0.925 (8)	0.993 (7)	0.856 (7)
PSC	0.948 (1)	0.994 (1)	0.902 (1)
A549_{SIM}	1.000 (1)	1.000 (1)	1.000 (1)
$\text{Fluo2D}_{\text{SIM+}}$	0.982 (5)	0.998 (5)	0.966 (4)
$\text{Fluo3D}_{\text{SIM+}}$	0.993 (2)	0.999 (2)	0.988 (2)

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5, paragraph **Limitations.** discusses the $O(|E|^2)$ attention complexity, the resulting memory increase ($\sim$100$\times$ for k=10 neighbors), and the need for tiling on very large fields of view.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results (theorems, lemmas, or formal proofs). The mathematical content consists of model definitions (architecture, loss, ILP formulation) and an analytical formula for segment-to-segment distance (Appendix A), which is a standard geometric computation.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full architecture specifications, optimizer settings, learning rate schedule, data augmentation pipeline, loss configuration, ILP parameters, input features, and RoPE configuration are provided in Appendix A. The cross-validation protocol and dataset list are described in Sec. 4. The incremental correction protocol is detailed in Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The HOCT source code and trained model weights are publicly available on GitHub (`https://github.com/royerlab/hoct`) under the MIT license. All datasets and baselines used in the experiments are already publicly available: the Cell Tracking Challenge datasets [37, 55], the bacteria tracking benchmark and Trackastra codebase from [17], and the Attrackt code from [31]. Full architecture, hyperparameters, optimizer schedule, data augmentation, ILP configuration, and the incremental correction protocol are documented in Appendices A and I.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4 describes the cross-validation protocol, metrics, and distance threshold. Appendix A provides all hyperparameters: architecture dimensions, optimizer (Muon–Adam

hybrid with per-parameter-group learning rates), learning rate schedule, data augmentation, loss function, ILP configuration, and input features. Appendix I details the incremental correction protocol for both HOCT and the Trackastra baseline.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The bacteria benchmark (Table 4) and edge-stage aggregation ablation (Table 2) report mean $\pm$ standard deviation across cross-validation folds or sequences. The CTC ablation (Table 1) reports results for both cross-validation splits individually rather than aggregating, allowing the reader to assess variability. The splits are fixed by the dataset provider and the ILP solution is deterministic.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix A reports hardware (NVIDIA H200 for training and NVIDIA RTX 3090s for inference/correction), CPU/RAM specifications, training time ($\sim$18 h per cross-validation split or ablation row), and untiled per-dataset network time and peak GPU memory. Appendix I reports full per-round wall-clock times for the correction experiment ($\sim$2 min for HOCT, $\sim$10 min for the Trackastra linear probe, and $\sim$40–55 min for LoRA variants).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available cell tracking benchmarks, involves no human subjects, and poses no foreseeable ethical concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 5, paragraph **Broader impact.** discusses positive applications (drug screening, developmental biology, cancer metastasis research) and acknowledges the risk of misuse in unethical research contexts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The model is a domain-specific cell tracking tool that does not pose risks of misuse comparable to generative models or large language models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets used in this work are publicly available and properly cited: the Cell Tracking Challenge datasets [37, 55], the bacteria tracking benchmark and Trackastra codebase from [17], and the Attrackt code from [31]. They are used as baselines and benchmark data under their original publicly available terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The HOCT source code and trained model weights are released on GitHub (`https://github.com/royerlab/hoct`) under the MIT license, accompanied by documentation covering installation, training, and evaluation, together with the configuration files used to produce the reported results.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. The "incremental correction" experiment uses a simulated oracle with access to ground-truth annotations, not human annotators.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not involve LLMs. No LLM is used as a component of the architecture, training, or evaluation pipeline.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.