ConstTrack: Constellation-Guided Cell Tracking Under Lineage Constraints

Yiwen Xu[1], Yang Song[1], and Erik Meijering[1]

School of Computer Science and Engineering
University of New South Wales, Sydney, NSW 2052, Australia
{yiwen.xu1,yang.song1,erik.meijering}@unsw.edu.au

Abstract. Cell tracking in time-lapse microscopy is a fundamental problem for understanding cellular dynamics and lineage development. However, it remains highly challenging due to dynamic cellular behaviours such as shape variation, appearance drift, and complex division events. Recent deep learning-based approaches have achieved remarkable progress, yet they often rely primarily on appearance and positional cues while overlooking structural relationships among neighbouring cells, limiting their robustness under morphological changes and visual ambiguity. We propose ConstTrack, a context-aware and lineage-constrained cell tracking framework that leverages a contextual constellation signature to encode the geometric configuration of neighbouring cells, enhancing robustness under appearance drift. The framework further employs a track-pool memory to maintain both active and recently disappeared trajectories, together with lineage attributes and spatial information for stable association. A lineage-constrained delayed-decision module commits parent assignments within a sliding temporal window, enforcing biological plausibility. In addition, a spatial partitioning strategy enables memory-efficient detection on dense microscopy scenes, where detections are merged in global coordinates before identity prediction to maintain global identity consistency. Experimental results on multiple cell types and datasets, including those from the Cell Tracking Challenge and DeepCell, demonstrate that our framework achieves competitive lineage accuracy compared with recent state-of-the-art methods. The code is available at https://github.com/YiwenXu/ConstTrack.

Keywords: Cell tracking · Multi-object tracking · Neighbourhood geometry · Track-pool memory.

1 Introduction

The spatiotemporal behaviours of cells in time-lapse microscopy, including migration, proliferation, differentiation, and interaction, encode rich biological information [6]. Reconstructing cell trajectories and lineage trees enables quantitative analysis of how these behaviours evolve under different physiological or pathological conditions, supporting studies in developmental biology [20], regenerative medicine [13], and cancer research [7]. Cell tracking lies at the core

of this process. Given an image sequence, a tracker must localise individual cells, preserve their identities over time, and detect division events to assemble coherent lineage trees [19]. This remains challenging because cells are densely packed, morphologically similar, and deform non-rigidly, while appearance drift and microscopy artefacts weaken reidentification cues. Short-term disappearances and frequent mitosis further amplify ambiguity, making naive proximity- or appearance-based linking prone to identity switches and lineage errors.

Deep learning has substantially advanced cell tracking, and existing approaches are often grouped into segmentation-based, detection-based, and association-focused methods [19, 6]. Segmentation-based trackers infer instance masks and then establish interframe correspondences using overlap ratios, centroid distances, motion priors, or global solvers such as min-cost flow [3, 1]. EmbedTrack learns instance embeddings to better separate touching cells before temporal association [17]. Nonetheless, segmentation bias and domain shift can still propagate to association and lineage reconstruction [23]. Detection-based methods first localise cell centres or instances and then link them across time based on spatial proximity, motion filtering, or learned embeddings, reducing reliance on pixel-accurate masks [4, 16]. Multiframe point matching (MPM) stabilises short-term associations by jointly leveraging appearance and motion [10], with consistency regularisation and division constraints for dense scenes with frequent mitosis [11]. More recent single-stage designs further treat cells as keypoints and learn localisation and association scores end-to-end [14]. Complementarily, association-focused approaches learn data association explicitly with global reasoning, for example with spatiotemporal graphs, structured learning, or attention mechanisms [2, 12, 5, 8]. While these methods strengthen temporal consistency, association is still dominated by per-instance cues and upstream candidate quality, and explicit modelling of local neighbourhood geometry remains limited, leading to identity switches and trajectory breaks in confluent regions.

From a broader computer vision perspective, cell tracking relates closely to multiple object tracking (MOT), which aims to detect instances and maintain their identities over time [18]. Recent MOT methods increasingly use Transformer-based sequence modelling and direct identity prediction to reduce handcrafted cost design and better align training and inference [21, 25, 9]. However, directly transferring these methods to microscopy is not trivial, as cells undergo strong deformation, are densely packed and visually similar, and exhibit division and lineage constraints. Moreover, large fields of view and long sequences require careful control of memory and compute when using attention-based association.

A key observation in microscopy is that, over short time intervals, the relative geometric configuration of a cell and its nearby neighbours can be more stable than raw appearance [15]. This context provides an additional cue for distinguishing look-alike cells and for recognising characteristic local changes around mitosis. Motivated by this, we propose ConstTrack, a unified framework that treats association as online identity prediction augmented with explicit neighbourhood geometry. ConstTrack detects cells with a Deformable DETR detector and augments each detection with a contextual constellation signature that en-

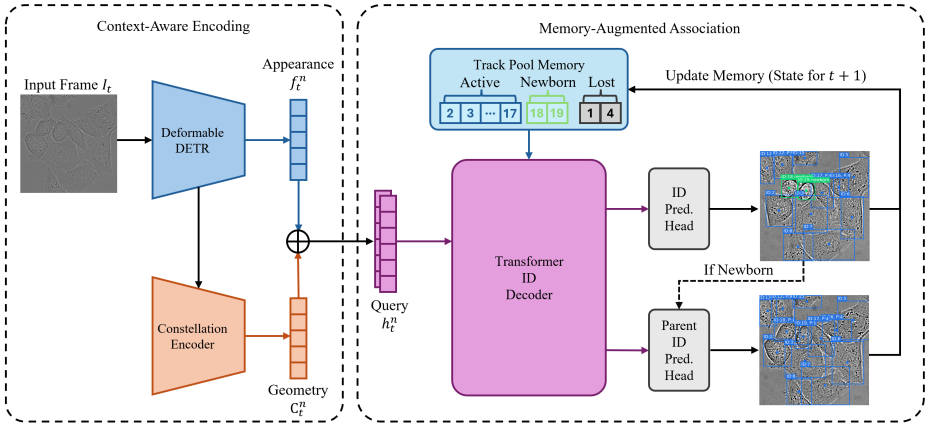

Fig. 1. Our proposed method framework. Tracking-by-detection is formulated as an ID prediction task. Context-Aware Encoding: Instance embeddings from the Deformable DETR detector are fused with a **Constellation Signature** that encodes local neighbourhood geometry via relative displacements to neighbouring detections G_t^n to handle appearance drift and dense packing. Memory-Augmented Association: A Transformer decoder queries a **Track-Pool Memory** to predict cell identities for frame t, and newborn detections are further processed by the Parent ID head for delayed lineage assignment.

codes local neighbourhood geometry via relative displacement feature vectors to neighbouring detections. A track-pool memory stores the states of active and recently missed tracks together with lineage attributes. At each time step, a Transformer ID decoder attends to an eligible subset of this memory to directly predict the identity (or a newborn class) for each detection, while a parent head supports lineage reconstruction at division events. To scale to large dense scenes, we adopt a halo-based spatial partitioning strategy for memory-efficient detection while preserving globally consistent identities.

We evaluate ConstTrack on diverse datasets from the Cell Tracking Challenge [19] and DeepCell [22]. ConstTrack achieves competitive lineage-oriented tracking accuracy against strong recent baselines, and targeted ablations support the benefits of neighbourhood geometry and track-history memory in representative challenging cases.

2 Method

ConstTrack (Fig. 1) performs online tracking by predicting IDs conditioned on track memory, augmented with neighbourhood geometry and committing parent-ID assignments for mitosis with a bounded delay.

2.1 ConstTrack architecture

Cell detector: For each frame $I_t \in \mathbb{R}^{H\times W\times C}$, with height H, width W, and C channels, we employ a Deformable DETR [26] as a cell detector to obtain a set of detections and associated instance embeddings. Given a fixed set of object queries, the decoder predicts for each query n a bounding box $b_t^n = (x_t^n, y_t^n, w_{t,\text{box}}^n, h_{t,\text{box}}^n)$, a classification confidence, and an instance embedding $f_t^n \in \mathbb{R}^{D_f}$. During training, we forward the per-query predictions $\{(b_t^n, f_t^n)\}$ to the downstream tracking modules without confidence thresholding. At inference, we retain high-confidence predictions as detections, yielding N_t candidates in frame t. The detector is applied frame-wise with shared parameters across time.

Constellation-based local geometry encoder: Deformable DETR provides strong appearance and shape cues. In microscopy, neighbouring cells can be visually similar and may undergo appearance drift, so local context provides a useful complement for reliable association. We therefore introduce a constellation encoder that embeds each detection's local spatial configuration as a geometry feature. We use the term "constellation" to denote this local point pattern of neighbours around a cell. For detection n at time t with centre $p_t^n = (x_t^n, y_t^n)$, we take the up to K nearest neighbours in the same frame and form the constellation signature as an unordered set of relative displacement feature vectors

$$G_t^n = \{g_{nj}\}_{j\in\mathcal{N}_K^n}, \quad g_{nj} = [\Delta x_{nj}, \Delta y_{nj}, r_{nj}] \in \mathbb{R}^3. \tag{1}$$

where $\Delta x_{nj} = (x_t^j - x_t^n)/s_t^n$, $\Delta y_{nj} = (y_t^j - y_t^n)/s_t^n$, $r_{nj} = \sqrt{\Delta x_{nj}^2 + \Delta y_{nj}^2}$, and $s_t^n = \sqrt{w_{t,\text{box}}^n h_{t,\text{box}}^n}$ is computed from b_t^n. Here, $\mathcal{N}_K^n$ denotes the indices of the K nearest detections to n in frame t; if fewer neighbours exist we use all available detections, and if the set is empty we set c_t^n to the zero vector. We include r_{nj} as an explicit rotation-invariant distance cue. We obtain a geometry embedding with a permutation-invariant set encoder $c_t^n = \phi_{\text{geo}}(G_t^n) \in \mathbb{R}^{D_{\text{geo}}}$, where ϕ_{geo} applies a shared MLP to each g_{nj} and uses mean pooling for stable aggregation before an output MLP. We set $K = 6$ in all experiments, since in dense 2D cell layers the local neighbourhood often resembles a near-hexagonal local packing, for which $K = 6$ provides a compact yet informative context without introducing distant, noisy neighbours [15]. We fuse geometry with appearance as

$$h_t^n = W_f f_t^n + W_c c_t^n \in \mathbb{R}^D, \tag{2}$$

where W_f and W_c are learned linear projections. The fused h_t^n is used by the subsequent tracking and lineage modules.

Track-pool memory: We maintain a track-pool memory $\mathcal{M}_t = \{T_t^m\}_{m=1}^{M_t}$ that stores all tracks created up to time t. Each track stores its state history (box, centre and fused embedding) and auxiliary track metadata including last-seen frame, miss counter, and parent ID; association uses only the most recent L states. The miss counter is reset when matched or newborn and incremented when unmatched. For association at time t, we attend to tracks in the eligible

set $\mathcal{A}_{t-1} = \{m : \mathrm{miss}(m) \le \delta\}$ with miss counter at most δ, which reduces interference from tracks with miss counter exceeding δ while keeping their histories in $\mathcal{M}_t$.

Cell ID decoder: At each frame t, a Transformer decoder performs self-attention among detections and cross-attention to eligible track tokens $\{u^m_{t-1}\}_{m\in\mathcal{A}_{t-1}}$, where u^m_{t-1} summarises the last L states of track m. The decoder outputs $\tilde{h}^n_t$ and produces association logits by dot-product similarity to each candidate track, plus a learnable initiation token u_{new}:

$$s_{t,n,m} = \langle W_q\tilde{h}^n_t,\ W_k u^m_{t-1}\rangle, \quad s_{t,n,\mathrm{new}} = \langle W_q\tilde{h}^n_t,\ W_k u_{\mathrm{new}}\rangle. \tag{3}$$

Stacking these scores yields $\ell^n_t \in \mathbb{R}^{|\mathcal{A}_{t-1}|+1}$, where the last index indicates "newborn". At inference, each detection takes the most probable label. If multiple detections select the same track, we keep the highest-confidence one. We enforce one-to-one matching on existing tracks, and assign unmatched detections to the newborn class.

Parent ID prediction for lineage: For detections predicted as newborn, we additionally predict a parent identity in frame $t-1$. We select up to P candidate parents among tracks last seen at $t-1$ within a normalised radius ρ of the newborn centre, using the distance $r^i = \sqrt{(\Delta x^i)^2 + (\Delta y^i)^2}$, where $\Delta x^i = (x^i_{t-1} - x^n_t)/s^n_t$ and $\Delta y^i = (y^i_{t-1} - y^n_t)/s^n_t$. We index candidates by increasing distance r^i and ignore unused candidate slots when fewer than P candidates satisfy the radius constraint. For candidate i with embedding h^i_{t-1} and centre p^i_{t-1}, we compute a displacement feature $g^i = [\Delta x^i, \Delta y^i, r^i]$ and score each pair with a shared MLP over

$$z_{t,n,i} = \left[h^n_t,\ h^i_{t-1},\ h^n_t - h^i_{t-1},\ g^i\right]. \tag{4}$$

The parent head outputs logits $\{a_{t,n,i}\}_{i=1}^{P}$ and an additional $a_{t,n,\varnothing}$ for "no-parent", yielding a $(P{+}1)$-way softmax. We use a sliding-window delayed decision for parent assignment. Starting from the first newborn frame t_0, we accumulate parent logits over the next τ frames for the selected candidates and commit the parent as the maximiser at $t_0 + \tau - 1$. At inference, we enforce binary division by retaining, for each parent, at most two highest-scoring child links.

Spatial partitioning for dense frames: For dense or large-FOV datasets, the number of detections and downstream ID-decoder queries can become large, which can push full-frame processing beyond GPU memory limits. We therefore use overlapping halo tiles during training and inference to keep computation tractable. The tile width w_{tile} and halo overlap o are fixed per dataset/cell type. At inference, tile detections are mapped back to global coordinates, and duplicate hypotheses in overlapping regions are merged or suppressed before tracking. Tracking and lineage reasoning always operate in the global coordinate frame via the shared track pool.

2.2 Training objectives

We train end-to-end on short clips and unroll the tracker over time, using teacher forcing to update $\mathcal{M}_t$ with ground-truth associations.

Detection loss: We follow Deformable DETR [26] and apply Hungarian matching per frame. Let Ω_{det} be the matched prediction-ground-truth triplets (t, n, γ) across all frames. For each match we use focal classification and box regression:

$$\mathcal{L}_{\text{det}} = \frac{1}{|\Omega_{\text{det}}|} \sum_{(t,n,\gamma)\in\Omega_{\text{det}}} \Big(\mathcal{L}_{\text{focal}}(\hat{y}_t^n, y_t^\gamma) + \lambda_{\ell_1} \|b_t^n - b_t^\gamma\|_1 + \lambda_{\text{giou}} \mathcal{L}_{\text{giou}}(b_t^n, b_t^\gamma) \Big). \quad (5)$$

Identity and parent losses: Let Ω_{id} be the set of detections matched to ground-truth cells. The target $y_{t,\text{id}}^n$ indexes the associated ground-truth track within the eligible set $\mathcal{A}_{t-1}$, or the newborn class. The ID decoder predicts ℓ_t^n over the eligible tracks and the newborn class, supervised by cross-entropy:

$$\mathcal{L}_{\text{id}} = \frac{1}{|\Omega_{\text{id}}|} \sum_{(t,n)\in\Omega_{\text{id}}} \text{CE}(\ell_t^n, y_{t,\text{id}}^n). \quad (6)$$

Let Ω_{par} index newborn detections. The target $y_{t,n,\text{par}}$ indexes the selected parent candidate in the $(P+1)$-way classification; for non-division newborns we set $y_{t,n,\text{par}} = 0$ to denote "no-parent". The parent head outputs logits $\{a_{t,n,i}\}_{i=1}^P$ and an additional $a_{t,n,\varnothing}$ for "no-parent", forming a $(P+1)$-way softmax:

$$\mathcal{L}_{\text{parent}} = \frac{1}{|\Omega_{\text{par}}|} \sum_{(t,n)\in\Omega_{\text{par}}} \text{CE}([a_{t,n,\varnothing}, a_{t,n,1}, \ldots, a_{t,n,P}], y_{t,n,\text{par}}). \quad (7)$$

The final training objective is

$$\mathcal{L} = \lambda_{\text{det}} \mathcal{L}_{\text{det}} + \lambda_{\text{id}} \mathcal{L}_{\text{id}} + \lambda_{\text{parent}} \mathcal{L}_{\text{parent}}. \quad (8)$$

3 Experiments

3.1 Datasets and metrics

Datasets: We evaluated ConstTrack on the Cell Tracking Challenge (CTC) and DeepCell DynamicNuclearNet. For CTC, we used the publicly available 2D sequences, with `01` for training and `02` for testing. For DeepCell, we used three 2D nuclear tracking subsets (HeLa, PC-3, RAW264) and adopted the official train/validation/test split.

Metrics: We followed the CTC protocol and used the TRA metric for overall tracking/lineage accuracy, and Cell-HOTA with DetA, AssA and DivA to separately assess detection, association and division performance [19, 23]. For CTC evaluation, we converted predicted boxes to instance-labelled masks by rasterising each detection into a label map, enabling evaluation with the official CTC scripts. Since our focus is tracking and lineage, we report TRA and Cell-HOTA and do not optimise for boundary accuracy measured by the SEG metric.

Table 1. Quantitative results on CTC 2D datasets in terms of TRA and Cell-HOTA with DetA, AssA and DivA. Best performance per metric in bold.

	BF-C2DL-HSC					**Fluo-N2DH-GOWT1**				
Method	TRA↑	HOTA↑	DetA↑	AssA↑	DivA↑	TRA↑	HOTA↑	DetA↑	AssA↑	DivA↑
EmbedTrack	0.973	0.843	0.807	0.836	0.927	0.956	0.749	0.749	0.784	0.717
TrackAstra	0.983	**0.865**	0.823	0.876	**0.945**	0.974	0.810	0.794	**0.854**	0.801
CAP	0.985	0.841	0.826	0.876	0.838	0.972	0.761	0.787	0.846	0.641
Cell-TRACTR	0.986	0.849	0.819	0.868	0.892	**0.976**	**0.858**	**0.799**	0.847	**1.000**
Ours	**0.987**	0.864	**0.829**	**0.885**	0.915	**0.976**	0.857	**0.799**	0.846	**1.000**
	BF-C2DL-MuSC					**Fluo-N2DH-SIM+**				
EmbedTrack	0.973	0.827	0.800	0.822	**0.887**	0.961	0.808	0.766	0.847	0.859
TrackAstra	0.975	0.820	0.791	0.847	0.852	0.973	**0.840**	0.794	0.859	**0.917**
CAP	0.971	0.827	0.797	0.860	0.852	0.970	0.750	0.762	**0.861**	0.633
Cell-TRACTR	0.974	**0.846**	**0.818**	0.863	**0.887**	0.972	0.805	**0.804**	0.842	0.770
Ours	**0.976**	0.810	0.796	**0.865**	0.787	**0.975**	0.818	0.767	0.850	0.897
	DIC-C2DH-HeLa					**Fluo-N2DL-HeLa**				
EmbedTrack	0.932	0.755	0.727	0.789	0.779	0.977	0.817	0.828	0.856	0.761
TrackAstra	0.953	0.733	0.770	0.804	0.607	0.990	0.815	0.846	0.882	0.699
CAP	0.949	0.734	**0.777**	0.793	0.607	0.989	0.790	**0.847**	0.875	0.621
Cell-TRACTR	0.954	0.762	0.773	**0.822**	0.687	**0.991**	0.809	0.842	**0.899**	0.673
Ours	**0.955**	**0.798**	0.770	0.773	**0.882**	**0.991**	**0.843**	0.846	0.890	**0.794**
	Fluo-C2DL-Huh7					**PhC-C2DH-U373**				
EmbedTrack	0.937	0.706	0.681	0.780	0.687	0.971	0.833	0.749	0.856	**1.000**
TrackAstra	0.952	0.779	0.771	0.795	0.779	**0.984**	**0.875**	**0.822**	0.866	**1.000**
CAP	0.954	0.693	**0.772**	0.821	0.472	0.982	0.826	0.811	0.863	0.819
Cell-TRACTR	0.954	**0.835**	0.753	**0.855**	**1.000**	0.983	0.828	0.819	0.855	0.819
Ours	**0.955**	0.826	0.752	0.825	**1.000**	**0.984**	0.869	0.805	**0.878**	**1.000**
	Fluo-C2DL-MSC					**PhC-C2DL-PSC**				
EmbedTrack	0.801	0.619	0.637	0.603	–	0.959	0.768	0.789	0.831	0.674
TrackAstra	0.861	0.646	0.670	0.624	–	0.962	0.807	0.798	0.842	0.793
CAP	0.864	0.652	0.659	0.645	–	0.963	0.783	0.787	0.838	0.725
Cell-TRACTR	0.871	0.620	**0.694**	0.555	–	0.965	0.813	**0.801**	**0.843**	0.808
Ours	**0.873**	**0.663**	0.675	**0.650**	–	**0.967**	**0.815**	0.780	0.826	**0.879**

3.2 Implementation details

We trained Deformable DETR from scratch with a ResNet-50 backbone and hidden size 256 for detection, and a 4-layer Transformer ID decoder with hidden size 256. We trained on 8-frame clips for 30 epochs with AdamW and cosine decay on 4 NVIDIA V100 GPUs. Learning rates were 2×10^{-4} for Transformer modules and 2×10^{-5} for the backbone, with weight decay 10^{-4}. We set $\lambda_{\text{det}} = 2.0$, $\lambda_{\text{id}} = 1.0$, and $\lambda_{\text{parent}} = 1.0$, and used Deformable DETR regression weights $\lambda_{\ell_1} = 5.0$ and $\lambda_{\text{giou}} = 2.0$. We set the history length to $L = 8$ and the delayed-decision window to $\tau = 3$ frames. We used $\delta = 8$, and for parent assignment we considered up to $P = 6$ candidates with $r^i \leq \rho$ and $\rho = 3$. At inference, we retained detections with confidence ≥ 0.3.

Table 2. Quantitative results on the DeepCell datasets in terms of TRA and Cell-HOTA with DetA, AssA and DivA. Best performance per metric in bold.

Method	TRA ↑	HOTA ↑	DetA ↑	AssA ↑	DivA ↑
HeLa					
EmbedTrack	0.919	0.712	0.663	0.701	**0.836**
TrackAstra	0.922	0.725	0.671	0.732	**0.836**
CAP	**0.926**	0.688	0.715	0.773	0.565
Cell-TRACTR	0.923	**0.740**	**0.717**	**0.807**	0.725
Ours	**0.926**	0.716	0.689	0.792	0.700
PC-3					
EmbedTrack	0.935	0.796	0.751	0.712	**1.000**
TrackAstra	0.950	0.829	0.762	0.812	**1.000**
CAP	0.952	**0.838**	**0.764**	**0.845**	**1.000**
Cell-TRACTR	0.950	0.813	0.733	0.814	**1.000**
Ours	**0.953**	0.827	0.761	0.808	**1.000**
RAW264					
EmbedTrack	0.817	0.647	0.598	0.630	0.777
TrackAstra	0.856	0.666	0.621	0.640	0.800
CAP	0.864	0.663	0.634	**0.693**	0.694
Cell-TRACTR	0.863	0.656	0.606	0.689	0.732
Ours	**0.865**	**0.713**	**0.688**	0.663	**0.824**

3.3 Comparisons with state-of-the-art methods

We compared ConstTrack to Cell-TRACTR [23], EmbedTrack [17], TrackAstra [8], and Cell as Point (CAP) [24]. On CTC (Table 1), ConstTrack achieves the best or tied-best TRA across all evaluated sequences, demonstrating strong identity consistency over time. This advantage is consistent with our use of geometry-aware constellation signatures and a track-pool memory, which provide complementary cues when appearance alone is insufficient or when tracks are temporarily missed. In terms of Cell-HOTA, ConstTrack is generally competitive; when it is not the best method, the gap is largely explained by DetA or AssA rather than DivA, suggesting that improving the detector or short-range association would likely yield higher HOTA while the lineage module does not appear to be the primary limiting factor. On DeepCell (Table 2), ConstTrack also achieves competitive TRA and performs strongly on RAW264 and PC-3, while HeLa remains challenging mainly due to DetA or AssA compared with Cell-TRACTR. DivA is evaluated only on ground-truth division edges; it is undefined for Fluo-C2DL-MSC because no division events are present, and it shows a ceiling effect on PC-3 since only two division edges exist, making the score coarse and prone to saturate at 1.0.

3.4 Ablations

Removing the constellation encoder drops TRA on Fluo-N2DH-GOWT1 from 0.976 to 0.833, showing that within-frame neighbourhood geometry aids association in dense scenes. Increasing the neighbourhood size beyond six neighbours yields diminishing returns, and we therefore cap K at 6. Using only the previous frame, $L = 1$, lowers TRA on PhC-C2DH-U373 from 0.984 to 0.724, whereas $L = 8$ achieves 0.984; we adopt $L = 8$ as an accuracy–memory trade-off.

4 Discussion and conclusion

We presented ConstTrack, a context-aware framework for online cell tracking and lineage reconstruction. By combining Deformable DETR detection with a constellation-based geometry encoder, a track-pool memory, and a parent-ID head, ConstTrack improves identity preservation and division handling in dense microscopy. Experiments on the Cell Tracking Challenge and DeepCell datasets show strong performance in TRA and Cell-HOTA. A limitation is the fixed field-of-view assumption, which can truncate lineages under large motion; future work will address moving-stage imaging and longer-range re-identification.

Acknowledgments. This work was supported in part by the ARC Research Hub for Connected Sensors for Health (IH210100040), including industry partner support from Genesys Electronics Design.

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

References

1. Akram, S.U., Kannala, J., Eklund, L., Heikkilä, J.: Joint cell segmentation and tracking using cell proposals. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 920–924 (2016)
2. Ben-Haim, T., Raviv, T.R.: Graph neural network for cell tracking in microscopy videos. In: European Conference on Computer Vision (ECCV). pp. 610–626 (2022)
3. Bensch, R., Ronneberger, O.: Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1220–1223 (2015)
4. Bise, R., Yin, Z., Kanade, T.: Reliable cell tracking by global data association. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1004–1010 (2011)
5. Bragantini, J., Lange, M., Royer, L.: Large-scale multi-hypotheses cell tracking using ultrametric contours maps. In: European Conference on Computer Vision (ECCV). pp. 36–54 (2024)
6. Chai, B., Efstathiou, C., Yue, H., Draviam, V.M.: Opportunities and challenges for deep learning in cell dynamics research. Trends in Cell Biology **34**(11), 955–967 (2024)
7. Galli, B.M.: Microscopy video analysis for breast cancer research: an automatic pipeline for image processing and cell tracking. Ph.D. thesis, Politecnico di Torino (2025)
8. Gallusser, B., Weigert, M.: TrackAstra: Transformer-based cell tracking for live-cell microscopy. In: European Conference on Computer Vision (ECCV). pp. 467–484 (2024)
9. Gao, R., Qi, J., Wang, L.: Multiple object tracking as ID prediction. In: Computer Vision and Pattern Recognition (CVPR). pp. 27883–27893 (2025)

10. Hayashida, J., Nishimura, K., Bise, R.: MPM: Joint representation of motion and position map for cell tracking. In: Computer Vision and Pattern Recognition (CVPR). pp. 3823–3832 (2020)
11. Hayashida, J., Nishimura, K., Bise, R.: Consistent cell tracking in multi-frames with spatio-temporal context by object-level warping loss. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1727–1736 (2022)
12. Hirsch, P., Malin-Mayor, C., Santella, A., Preibisch, S., Kainmueller, D., Funke, J.: Tracking by weakly-supervised learning and graph optimization for whole-embryo C. elegans lineages. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 25–35 (2022)
13. Jin, Y., Li, S., Yu, Q., Chen, T., Liu, D.: Application of stem cells in regeneration medicine. MedComm **4**(4), e291 (2023)
14. Kamiya, S., Toida, K., Tsunoyama, T.a., Hotta, K.: Tracking correction method for rapid and random protein molecules movement. In: Asian Conference on Computer Vision (ACCV). pp. 3022–3037 (2024)
15. Li, Y., Naveed, H., Kachalo, S., Xu, L.X., Liang, J.: Mechanisms of regulating cell topology in proliferating epithelia: impact of division plane, mechanical forces, and cell memory. PLoS One **7**(8), e43108 (2012)
16. Liu, A.A., Lu, Y., Chen, M., Su, Y.T.: Mitosis detection in phase contrast microscopy image sequences of stem cell populations: A critical review. IEEE Transactions on Big Data **3**(4), 443–457 (2017)
17. Löffler, K., Mikut, R.: EmbedTrack–simultaneous cell segmentation and tracking through learning offsets and clustering bandwidths. IEEE Access **10**, 77147–77157 (2022)
18. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Kim, T.K.: Multiple object tracking: A literature review. Artificial Intelligence **293**, 103448 (2021)
19. Maška, M., Ulman, V., Delgado-Rodriguez, P., Gómez-de Mariscal, E., Nečasová, T., Guerrero Peña, F.A., Ren, T.I., Meyerowitz, E.M., Scherr, T., Löffler, K., et al.: The Cell Tracking Challenge: 10 years of objective benchmarking. Nature Methods **20**(7), 1010–1020 (2023)
20. Meijering, E., Dzyubachyk, O., Smal, I., van Cappellen, W.A.: Tracking in cell and developmental biology. Seminars in Cell & Developmental Biology **20**(8), 894–902 (2009)
21. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: TrackFormer: Multi-object tracking with transformers. In: Computer Vision and Pattern Recognition (CVPR). pp. 8844–8854 (2022)
22. Moen, E., et al.: Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. bioRxiv:803205 (2019)
23. O'Connor, O.M., Dunlop, M.J.: Cell-TRACTR: A transformer-based model for end-to-end segmentation and tracking of cells. PLoS Computational Biology **21**(5), e1013071 (2025)
24. Song, Y., Fan, J., Huang, H., Chen, M., Cai, W.: Cell as Point: One-stage framework for efficient cell tracking. arXiv 2411.14833 (2024)
25. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: MOTR: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision (ECCV). pp. 659–675 (2022)
26. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159 (2020)