DiSTNet2D: Leveraging Long-Range Temporal Information for Efficient Segmentation and Tracking

Jean Ollion [1,*] Martin Maliet [2] Caroline Giuglaris [3] Élise Vacher [3] and Maxime Deforet [2,†]

[1] *SABILab, Die 26150, France*

[2] *Laboratoire Jean Perrin, Institut de Biologie Paris-Seine (IBPS), Sorbonne Université, Centre National de la Recherche Scientifique, Paris 75005, France*

[3] *Laboratoire PhysicoChimie Curie UMR168, Institut Curie, Paris Sciences et Lettres, Centre National de la Recherche Scientifique, Sorbonne Université, Paris 75005, France*

(Received 11 December 2023; accepted 15 March 2024; published 16 April 2024)

Extracting long tracks and lineages from videomicroscopy requires an extremely low error rate, which is challenging on complex data sets of dense or deforming cells. Leveraging temporal context is key to overcoming this challenge. We propose DiSTNet2D, a new deep neural network architecture for two-dimensional (2D) cell segmentation and tracking that leverages both mid- and long-term temporal information. DiSTNet2D considers seven frames at the input and uses a postprocessing procedure that exploits information from the entire video to correct segmentation errors. DiSTNet2D outperforms two recent methods on two experimental data sets, one containing densely packed bacterial cells and the other containing eukaryotic cells. It is integrated into an ImageJ-based graphical user interface for 2D data visualization, curation, and training. Finally, we demonstrate the performance of DiSTNet2D on correlating the size and shape of cells with their transport properties over large statistics, for both bacterial and eukaryotic cells.

DOI: 10.1103/PRXLife.2.023004

I. INTRODUCTION

Automated image analysis is revolutionizing the study of cells, enabling scientists to measure their position, velocity, shape, and fluorescent signal intensity. When cells are motile, cell segmentation (localizing cell boundaries) and tracking (associating cells in consecutive images) are combined to follow cell properties over time [1]. These data have already revealed fundamental mechanisms in cell motion, especially *in vitro* [2–5].

Further improving performance of cell segmentation and tracking will enable us to understand long-term correlation and extend analysis to more and more complex systems. However, extracting long lineages demands an extremely low error rate, which can be difficult to achieve with complex data sets.

Challenges commonly found in biological images stem from the following.

(i) Cell morphology: The shape and size of cells can differ from one another or be very similar within a cell population, and this can change over time, making cell morphology unreliable as a means to segment and track cells accurately.

(ii) Cell boundaries: The boundaries between cells can sometimes be difficult to distinguish, as they often touch each other.

(iii) Cell movement: The movement of cells can be drastically different from one to the next, or in time, for instance with abrupt changes of direction of motion.

(iv) Cellular events: Cellular events, such as mitosis or apoptosis, alter lineages and prove difficult to track using conventional general-object tracking methods.

For such complex data sets, improving the analysis of individual images is insufficient to overcome the challenges of cell segmentation and tracking. To increase performance, it is necessary to leverage temporal context by considering multiple frames simultaneously.

A. Cell segmentation

The main challenge in cell segmentation is not only foreground-background classification, but also the separation of adjacent cells. The emergence of deep neural networks (DNNs) has greatly improved the efficiency and robustness of segmentation methods. Early DNN-based methods were mostly based on pixelwise classification [6], which resulted in subefficient separation of adjacent cells. Another popular family of methods involves two sequential DNNs: a region proposal network that generates axis-aligned bounding box candidates followed by a classification network that filters and classifies them [7]. Although this type of method is very efficient on natural images, it is complex to train and has been shown to be unable to cope with certain cellular shapes [8]. Other methods do not directly segment cell instances but predict (in this article, the output computed by a DNN is referred to as prediction) pixelwise features as proxy for segmentation that are subsequently fed to a clustering algorithm: multidimensional embedding with a loss function that pushed dissimilarity between neighbors was proposed in Ref. [9], and

*jean.ollion@polytechnique.org

†maxime.deforet@sorbonne-universite.fr

it was proposed in Ref. [10] to predict the Euclidean distance map (EDM), fed to a watershed algorithm. Compared to a binary probability map, EDM has the advantage of emphasizing the boundary between touching cells while being independent of morphology. A popular method that is similar (but not equivalent) predicts radial distance between center and boundaries at predefined angles, which limits the application to convex objects [8]. An efficient method was proposed in Ref. [11] that jointly predicts, for each cell pixel, an offset vector pointing to the cell center and a clustering bandwidth. Similarly, in Ref. [12] a normalized offset vector pointing to the cell center was predicted. In the case of filamentous bacteria, this method tends to produce oversegmentation. This problem was reduced in Ref. [13] by predicting the EDM along with an offset vector pointing to the cell medial axis (skeleton), defined by the local maxima of the EDM.

B. Cell tracking

The most straightforward approach to cell segmentation and tracking runs in two independent successive steps: object detection followed by temporal association of detected objects. The recent method DeLTA 2.0 [14] uses, for the tracking step, a classification neural network to predict the next cell for each cell. However, because predictions are made independently for each cell, this method is likely to produce inconsistent results. A two-step approach can allow long-term temporal information to feed the tracking algorithm, as segmentation enables data compaction. Notably, a graph neural network was used in Ref. [15] to model the entire time-lapse sequence, resulting in very effective tracking. The main drawback of the two-step approach is that it is directly limited by the accuracy of the segmentation step. In difficult cases such as high density of similar cells, even a trained expert requires temporal context to perform accurate segmentation.

C. Combined segmentation and tracking

Temporal information can be leveraged by combining segmentation and tracking into a single operation. Several recent methods simultaneously train a bounding box detector with a tracker that associates bounding box candidates between successive frames [16,17]. In the context of cell tracking, one limitation of this kind of method is that they have restricted access to the spatial context around the bounding boxes (such as the position of neighboring objects), which is crucial when cells have similar aspects. An emerging trend is the prediction of the displacement vector of each cell between two successive frames as a proxy for tracking [18–21], along with a proxy for segmentation or detection. The actual association of cells is performed in a postprocessing step. One advantage of this strategy is that it enables simultaneous segmentation and tracking of all cells present in a time window using a single DNN, which likely yields more consistent results for both tasks. It is noteworthy that Refs. [18,21] do not segment cells but only detect their centers. Reference [19] introduced an attention layer [22] in the neural network, and showed that it captures long-range spatial information, in the one-dimensional case of bacterial cells growing in a microfluidic device. In Ref. [20] a DNN architecture was used that performs segmentation independently for each frame and thus cannot leverage temporal context for segmentation. However, several works have shown that performing joint segmentation and tracking improves segmentation by leveraging sequential information [9,19]. Due to memory limitations, these methods can only use a small temporal neighborhood; e.g., pairs of successive frames $(t, t+1)$ were used in Refs. [18–20]. More recently, it was shown in Ref. [21] that tracking and detection performance can be improved by using a larger temporal neighborhood of six frames as well as a carefully designed loss function that penalizes inconsistencies between detection and tracking.

In this work, we describe DiSTNet2D, a novel two-dimensional (2D) cell segmentation and tracking method, with a carefully designed DNN architecture that leverages mid- and long-term temporal context for both tasks. Mid-term temporal context is incorporated at the input of the DNN: our method typically considers a 15-frame time window, but this size is adaptable to the features of the data set and can be much wider if needed. Long-term temporal context is incorporated through a postprocessing procedure that uses information from the whole video to correct segmentation errors. We compare DiSTNet2D to two recent methods (DeLTA 2.0 [14] and EmbedTrack [20]) on two experimental data sets that we publish along with this work: a data set containing phase contrast images of dense communities of motile bacterial cells, and a data set of fluorescence images of adherent migrating eukaryotic cells. We also adapted the graphical user interface of BACMMAN software [23] for 2D data visualization, curation, and training. Finally, we demonstrate how DiSTNet2D's performance enables us to correlate the size and shape of cells with their motion properties over large statistics, for both bacterial and eukaryotic cells.

II. RESULTS

Following the work of Refs. [19–21], we developed a method that performs segmentation and tracking simultaneously with a single DNN. This strategy has several advantages over methods that perform the tasks independently. First, it leverages temporal information for segmentation, improving the accuracy of the results. Second, it is easier to train and use a single DNN than two separate networks. Our method is based on a novel DNN architecture that incorporates a sequence of operations designed to blend the information gathered from the different input frames, enabling the use of this information for both segmentation and tracking (see details in the Supplemental Material [24]). Specifically, several frames are fed to the DNN, which predicts proxies for both segmentation and tracking [Figs. 1(a) and 1(b)]. Using a larger time window is expected to increase temporal consistency, but the number of considered frames is limited by GPU memory. We chose to consider seven frames: three frames before and three frames after the current frame. To further extend the temporal range without exceeding the GPU's memory capacity, the selected frames are spaced farther apart within a larger time window (see Fig. S1 [24] for a diagram). The gap between considered frames depends on the data set; in this work we used values of one and three frames (depending on the data set). This strategy is referred to as temporal subsampling.

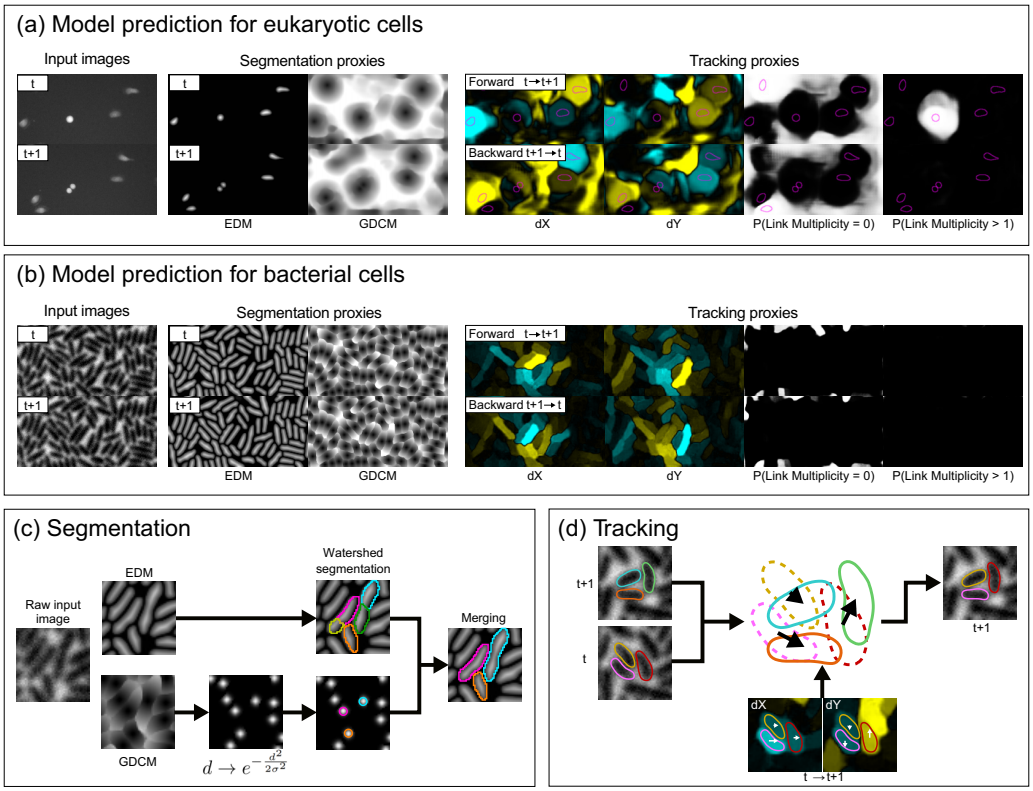

FIG. 1. Overview of the method. (a) Model output for a eukaryotic data set frame pair. For each frame, EDM is the map of the Euclidean distance to the edge of each cell, and GDCM is the map of the geodesic distance to the center. For each pair of frames, for both forward ($t \to t+1$) and backward ($t+1 \to t$) directions, dX and dY are the cell center displacements from the previous frame for each axis. Displacements toward the right (or downward) are indicated in blue, while displacements toward the left (or upward) are shown in yellow. P(Link multiplicity $= 0$) and P(Link multiplicity >1) are the probabilities that the link multiplicity is zero (no linked cell in the other frame) and strictly greater than one (several linked cells in the other frame, i.e., in the case of a division), respectively. Note that only one frame pair $(t, t+1)$ is represented, but the model inputs a larger temporal neighborhood and predicts these maps for more frame pairs. For a better readability, contours of segmented cells are represented in the eight rightmost images. (b) Model output for a bacterial data set frame pair. In this case, P(Link multiplicity >1) is null, reflecting the absence of cell division or merging events in this data set. (c) Segmentation procedure: A watershed transform is applied to the EDM using regional maxima as seeds, which likely produces oversegmentation. A Gaussian function is applied to each pixel of the predicted GDCM and a watershed algorithm on the Laplacian transform is used to detect centers, which are used to reduce oversegmentation (see main text for details). (d) Tracking procedure: The centers of each cell at t are shifted by the predicted displacement between t and $t+1$ (dX and dY). Each cell at t is associated with the cell at $t+1$ in which the shifted center falls (the panel illustrates simple links; for more complex links, like division and fusion links, see Fig. S4 [24]).

A. Segmentation

The network predicts two complementary proxies for segmentation, the EDM, which aims at identifying the cell shape, and the geodesic distance to the center map (GDCM), which aims at identifying the cell center. More formally, let B be the background, F the foreground, and C_j the jth cell ($F = \bigcup_j C_j$), c_j its center, d the Euclidean distance function, and d_g the geodesic distance function (see the Supplemental Material, Sec. 6 [24]), for each pixel i:

$$\mathrm{EDM}_i = \begin{cases} \min(d(i,B), d(i, F \setminus C_j)) & \text{if } i \in C_j \\ -1 & \text{if } i \in B, \end{cases}$$

$$\mathrm{GDCM}_i = d_g(i, c_j) \text{ if } i \in C_j.$$

The medoid center of the cells is used in order to ensure it is contained in the cell, even for nonconvex shapes. For simplicity, it will be referred to as the center in this work. GDCM is not used outside cells; therefore, we do not define it in B and the corresponding loss is only computed within cells.

EDM-based segmentation is efficient even on nonconvex cell morphologies, such as bent bacterial cells. It is performed by applying a watershed algorithm on EDM. Watershed is naturally limited to positive values (as the background is set to -1) and seeded from regional maxima of the EDM, which can easily produce oversegmentation, especially in long cells or cells with complex shapes that may contain several local maxima.

To reduce oversegmentation, we introduced a merging criterion based on prediction of centers. Centers are segmented by performing a watershed algorithm on the Laplacian transform of the image that results from the Gaussian function applied to each pixel of the GDCM (see the Supplemental Material, Sec. 6 [24], for details, and Fig. S2). All pairs of adjacent segmented regions are sorted based on the EDM values at their mutual boundary, meaning that the pair with

the smallest EDM value at their interface is considered first for merging. Two segmented regions in contact are merged if either one of them or both do not contain a segmented center, or if the ratio of intensity amplitude of the centers is below a user-defined threshold [see Fig. 1(c)].

Moreover, we also observed that predicting a unique center per cell improves EDM prediction especially in distinguishing neighboring cell instances.

B. Tracking

Tracking is performed using the prediction of the displacements of the cell centers along the X and Y axes that occurs between two frames. The center of each cell at t is shifted by its predicted displacement between t and $t+1$; if the shifted center falls into a segmented cell at $t+1$, then the two cell instances are associated [see Fig. 1(d)]. This is similar to the procedure used in Ref. [20].

To assist the tracking procedure and manage more complex cases, we also predict the *link multiplicity* category in both the forward ($t \rightarrow t+1$) and backward ($t+1 \rightarrow t$) directions, accounting for the expected number of links for each cell (Fig. S3 [24]). The possible values for forward link multiplicity are no next cell (the cell will leave the field of view, or will die), one next cell (regular case, or the cell will fuse with another cell), and multiple next cells (the cell will divide). The possible values for backward link multiplicity are no previous cell (the cell has entered the field of view, or just appeared), one previous cell (regular case, or the cell has just divided), and multiple previous cells (the cell has just fused). Formally, we predict each time three probability maps: $P(\text{Link Multiplicity} = 0)$, $P(\text{Link Multiplicity} = 1)$, $P(\text{Link Multiplicity} > 1)$, summing to 1. We assign the link multiplicity category to each cell as the multiplicity with the highest median probability within the cell.

Cells that are predicted to have a single next cell are linked by forward tracking using the predicted forward displacement. Cells with multiple next cells (either because of oversegmentation at $t+1$ and not at t, undersegmentation at t and not at $t+1$, or a predicted division event) remain unlinked after forward tracking. Backward tracking is then applied on unlinked cells that are predicted to have a single previous cell, using predicted backward displacement (Fig. S4 [24]).

Forward and backward tracking allows to assign both merge links—in which several cells at t are associated to a single cell at $t+1$—and split links—in which several cells at $t+1$ are associated to a single cell at t. When merge links or split links are not confirmed by the link multiplicity category, it means they arise from over- or undersegmentation errors and they will be corrected in the postprocessing stage (see Sec. II C). Identifying merge and split links also enables a finer definition of metrics and a more accurate diagnosis of origin of errors (incorrect segmentation vs incorrect linking). See Sec. II F for details.

C. Postprocessing: Segmentation correction

A set of rules was designed to correct over- or undersegmentation errors using the tracking information. We especially observed such errors in the PhC-C2DH-PA14 data set, which consists of high-speed acquisition videos (typically 100 frames per second for a few seconds) of motile rod-shaped bacteria that divide typically every hour. The invagination of the cell membrane at the center of the mother cell is the last stage of the cell cycle before the separation of the two daughter cells. In phase contrast imaging, this invagination appears as a bright region within the cell body, which is similar to the bright area that connects two separate cells when they are in contact (Fig. S5 [24]). It is virtually impossible to determine from a single frame whether an object represents a late-dividing long single cell or two adjacent short cells. The DNN architecture already helps avoid such errors by considering a mid-term temporal context (typically 15 frames).

To consider long-term temporal information (which can cover an extended period of time, up to the duration of the entire video), we analyze the trajectories obtained during the tracking step. We focus on the merge and split links. Merge and split links consistent with predicted link multiplicity are considered mitosis or fusion, and are left untouched. In contrast, merge and split links that contradict predicted link multiplicity are suspected errors. We treat them using the following general principle: if an object has always been seen as one cell, it should remain as one; however, if it was detected as two distant cells at any point in the past or future, this indicates that it should be considered two cells [Fig. 2(a)]. This approach is based on the assumption that errors are rare and can be corrected by looking at errorless past and future. The high efficiency of our DNN-based combined segmentation and tracking algorithm supports this assumption.

In practice, for each merge link, we check if all cells before the link are in contact (two objects are considered to be in contact if the distance between their contours is lower than a threshold; for rod-shaped bacteria, an alignment criterion is also used). If they are, then we merge all of the cells. Otherwise, we split the objects following the link by applying a watershed transform on the EDM [Fig. 2(a-i)]. Cell fragments are linked to the previous objects using the same procedure as in Sec. II B. If the watershed algorithm generates more fragments than there are previous objects, fragments linked to the same previous object are merged. Similarly, for each split link, if all cells after the link are in contact, then we merge all of the cells. Otherwise, we split the objects before the link [Fig. 2(a-ii)]. Common examples are depicted in Figs. 2(b)–2(d).

D. Model architecture

The DNN has an encoder-decoder architecture, with a single encoder and one decoder per output type (EDM, GDCM, displacement, and link multiplicity). The encoder and decoders are shared between frames for better training efficiency. For segmentation outputs (EDM and GDCM), one prediction per frame is made, whereas for tracking outputs (displacement and link multiplicity), one prediction per frame pair is made. For a DNN time window of size $(N-3)\delta+3$ centered on frame t, the N considered frames are the three central frames $(t-1, t, t+1)$, and $m=(N-3)/2$ frames on each side of the central frames $(t-1-m\delta, \ldots, t-1-2\delta,$

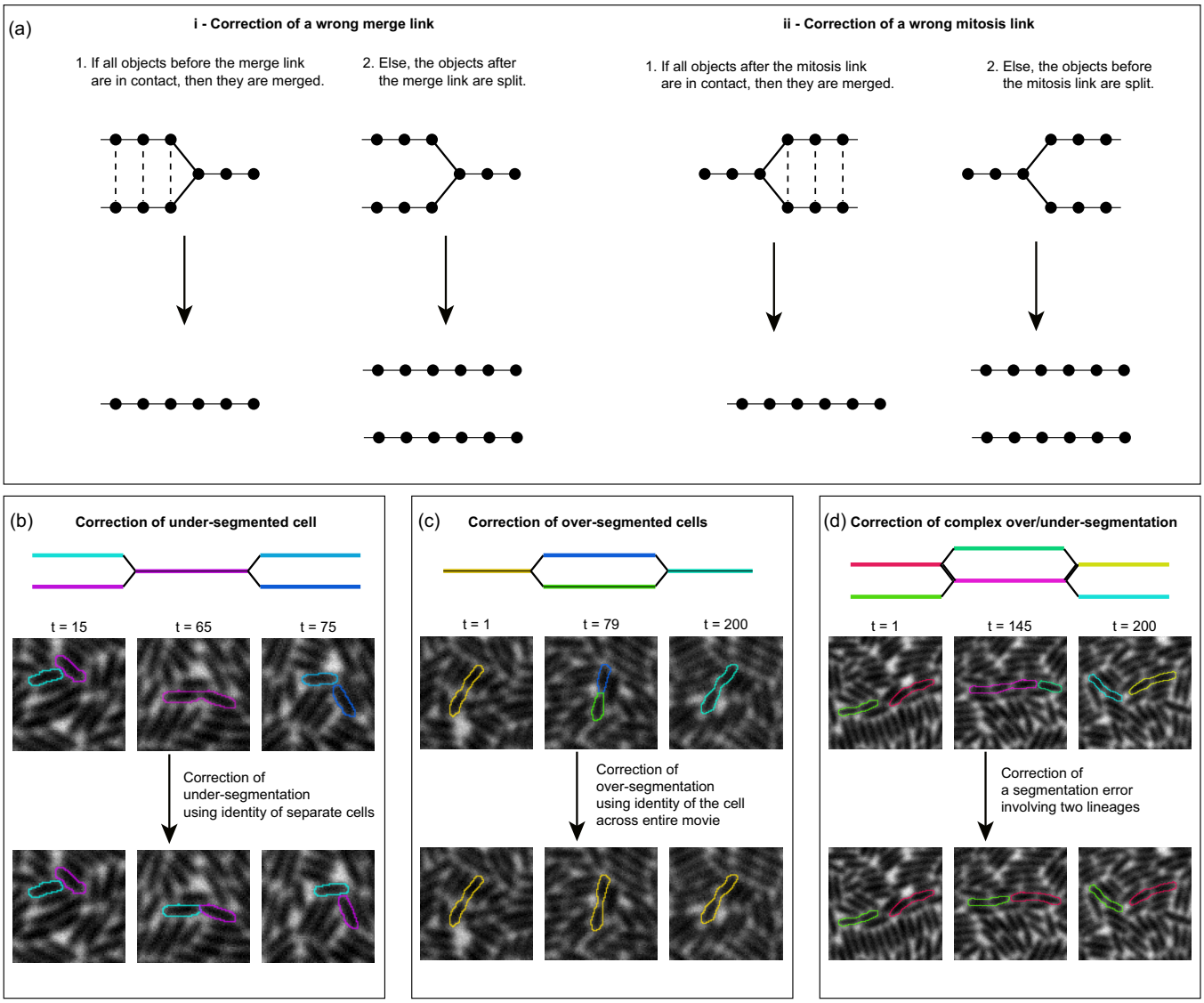

FIG. 2. Postprocessing uses temporal information on a large timescale to correct segmentation errors. (a) Diagrams presenting the procedure to correct wrong links. (b) Illustrative example of two distinct cells that are transiently detected as one object (undersegmented in frame $t = 65$). Because objects involved in the merge link are sometimes seen separated, we can assume the object in frame $t = 65$ should be split. (c) Illustrative example of one cell that is transiently detected as two objects (oversegmented in frame $t = 79$). Because the cell is detected as a single object throughout the entire video (200 frames), except for one frame in which it is seen as two objects, we can assume that the two objects should be merged into one. (d) Illustrative example of a more complex error that implies more lineages. In frame $t = 145$, one cell is undersegmented and one cell is oversegmented. Here again, temporal information at the scale of the entire video enables us to correct the segmentation errors.

$t - 1 - \delta$ and $t + 1 + \delta, t + 1 + 2\delta, \ldots, t + 1 + m\delta$) (Fig. S1 [24]). $2N - 4$ frame pairs are defined as follow (Fig. S6 [24]): (i) $N - 1$ frame pairs between consecutive considered frames, for short-range displacements, and (ii) $N - 3$ frame pairs between the central frame (frame t) and each other frame except frames $t - 1$ and $t + 1$, for mid-range displacements.

Figure 3 displays the global architecture. The detailed architecture of each box is described in the Supplemental Material (Sec. 1 [24]). The encoder and decoders are mainly composed of residual blocks of two successive convolutions. Between the encoder and the decoders, we introduced two blending modules (sequences of operations that blend the encoded features together): the *pair blender* module blends encoded features of each considered frame pair to generate encoded feature pairs, and the *blender* module blends all encoded features and encoded feature pairs together. Additionally, the *segmentation extractor* and the *tracking extractor* modules respectively extract one feature per frame and per frame pair, which are fed to the segmentation and tracking decoders. Extraction sequences simply contain a distinct convolution per frame (frame pair). Resulting tensors are combined with encoded features (feature pairs). We define the combine operation as a 1×1 convolution applied to the concatenation of two tensors.

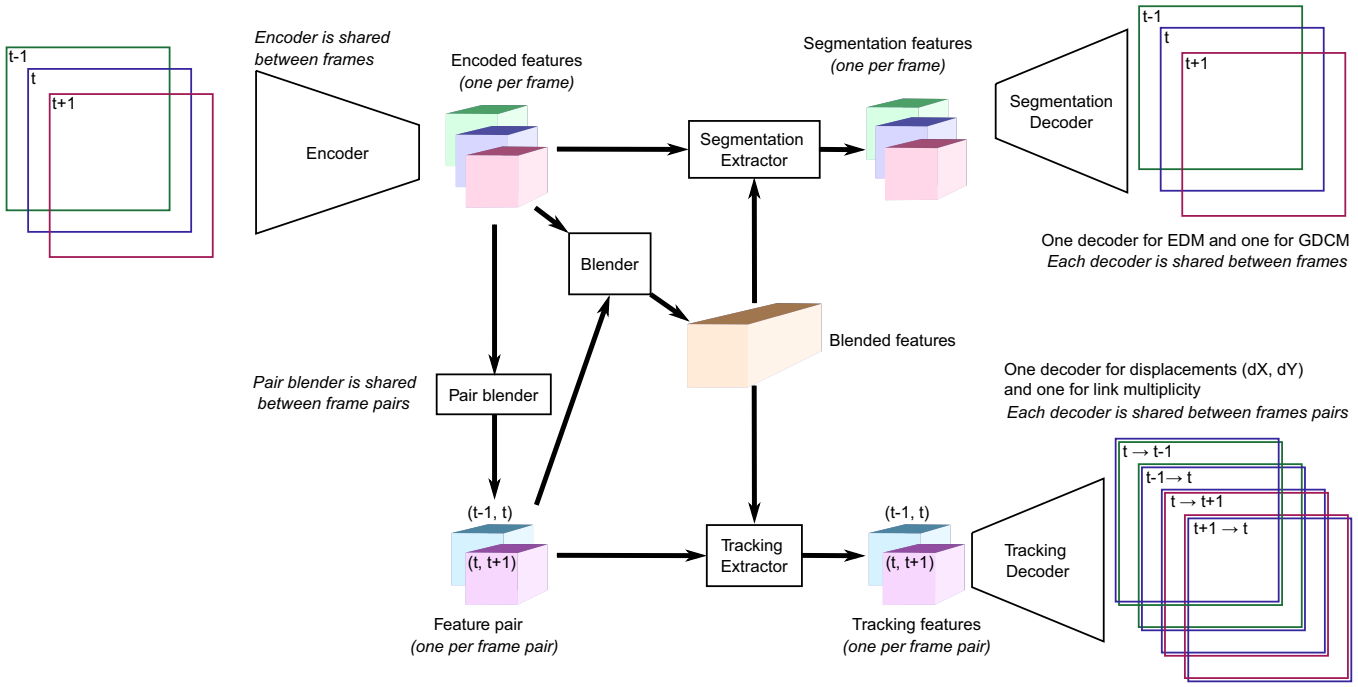

FIG. 3. Model architecture. The encoder is fed by successive frames (green, blue, and red rectangles) and produces encoded features (green, blue, and red cubes). Features are processed in pairs (corresponding to successive frames) by the pair blender module, which produces feature pairs. Encoded features and feature pairs are blended together by the blender module (see Fig. S7 [24] for details). The segmentation extractor generates three segmentation features corresponding to each frame for both EDM and GDCM, that are decoded by two distinct decoders to produce images of the same size as the input image. Likewise, the tracking extractor generates two tracking features corresponding to each frame pair for both displacement and link multiplicity, that are decoded by two distinct decoders. For simplicity only three frames have been represented but we considered seven frames in this work, and only one segmentation decoder and one tracking decoder are represented instead of two.

The blending-extraction sequence makes the information from the whole time window available for the prediction of each output at each frame. This contrasts with Ref. [20], in which segmentation is performed independently on each other frame.

This architecture has the advantage of allowing proxy predictions to be made only for the central frame and the frame pair $(t, t+1)$ during the prediction phase, which improves speed and reduces memory consumption.

E. Software

The software associated with our method is BACMMAN [23], an open-source ImageJ [25] plug-in that was initially developed for analysis of bacterial cells growing in microchannels, with displacement along the microchannel axis. Such data were naturally displayed on kymographs, in which the horizontal axis represents time. In order to display 2D data, we added the hyperstack visualization mode (see Fig. S8 [24]), in which lineage information is displayed as colored contours. All the features of the graphical user interface of BACMMAN such as interactive navigation through images, manual curation, and two-way interplay with R/PYTHON for statistical analysis, are thus available for 2D data. BACMMAN was also augmented with new features: generation of training sets as well as DiSTNet2D training and prediction can now be performed directly from the software. BACMMAN also provides a command-line interface, enabling its use on a computational cluster.

F. Evaluation metrics

Objective metrics for segmentation and tracking were previously introduced in Ref. [26] for the Cell Tracking Challenge. In that work, cell tracking results were represented using an acyclic oriented graph, in which nodes corresponded to the detected cells and edges represented links (i.e., temporal relations) between them. Metrics were based on the number of operations required to transform the result graph into the reference graph. Those operations were `split/delete/add` a node and `delete/add/change the semantics of` an edge (e.g., a change between a split link and a normal link). Correspondence between a reference (R) and a result (S) segmented object were established using the following criterion: $|R \cap S| > 0.5|R|$, which implied that each reference cell could correspond to one result cell at most. We consider this a limitation because it does not allow to take oversegmentation into account; oversegmented cells were thus systematically considered false positives. Instead, we used the following criterion: $|R \cap S| > 0.5 \min(|R|, |S|)$ OR $|R \cap S| > C$ with C a user-defined constant that is typically 50% of the average reference cell size. The second term accounts for cases of undersegmentation and partial overlap with ground truth, where the relative overlap is too low but the absolute overlap is significant.

This approach identifies four types of segmentation errors: false positives (result cells with no reference counterpart), false negatives (reference cells with no result counterpart), oversegmentation (when N result cells match a given

TABLE I. Comparison of DiSTNet2D, DeLTA 2.0, and EmbedTrack on data sets PhC-C2DH-PA14 and Fluo-C2DH-HBEC. Segmentation error is the sum of false positive, false negative, and under- and oversegmentation, divided by the number of cells in the ground truth. Tracking error is the sum of false positive and false negative links divided by the number of links in the ground truth. Incomplete lineages is the number of lineages with at least one segmentation or tracking error divided by the number of lineages in the ground truth. For data sets PhC-C2DH-PA14 and Fluo-C2DH-HBEC, respectively, the total number of cells is 123 057 and 4 273, the total number of links is 145 306 and 4 890, and the total number of lineages is 561 and 60. As explained in the main text, only cells that do not touch edges (and lineages with no cell touching edges) are taken into account.

Data set	Method	Segmentation error (%)	Tracking error (%)	Incomplete lineages (%)	Prediction time (s/frame)
PhC-C2DH-PA14	DeLTA 2.0	1.1	0.22	21	22
	EmbedTrack	1.2	0.14	7.1	2.7
	DiSTNet2D	0.17	0.00	0.53	0.51
Fluo-C2DH-HBEC	DeLTA 2.0	1.9	0.92	37	0.84
	EmbedTrack	0.89	0.51	25	1.4
	DiSTNet2D	0.49	0.10	5.0	0.27

reference cell, $N-1$ oversegmentations are counted), and undersegmentation (when N reference cells match a given result cell, $N-1$ undersegmentations are counted).

We also observed that the procedure proposed in Ref. [26] does not fully distinguish between tracking and segmentation errors: for instance, oversegmentation of one object into two parts is counted as a false positive segmentation error as well as a false negative link. However, if the oversegmented cell parts were all linked to the correct cell(s), no tracking error should be counted (Fig. S9 [24]). We thus developed a procedure inspired by Ref. [26] to identify tracking errors that are independent of segmentation errors. In other words, our procedure evaluates the tracking efficiency *per se*, given the segmentation errors.

To do so, for each frame pair $(t, t+1)$, we transform the nodes of the result graph so that they match with the nodes of the reference graph by applying four successive operations (Fig. S9 [24]): splitting undersegmented cells at $t+1$, splitting undersegmented cells at t, merging oversegmented cells at t, and merging oversegmented cells at $t+1$. At each split or merge operation, links are propagated to the resulting nodes. In the case of splitting, if this implies linking M nodes at t to N nodes at $t+1$, where $M>1$ and with $N>1$, links are determined by a simple linear assignment algorithm that minimizes the distance between cell centers. In the case of merging, all links are simply added to the resulting node. After these transformations, all nodes of the transformed result graph correspond to a single node in the reference graph, except for false positives, which have no counterparts in the result graph. This enables counting false positive and false negative links: the former are links found in the transformed result graph and not in the reference graph, except links automatically added by splitting an undersegmented object. The latter are links from the reference graph that are not found in the transformed result graph and that do not involve false negative objects. Our choice of link propagation in the case of merging during graph transformation would miss some false negative links, that are thus added to the count: in the case a result cell was merged at frame $t+1$ but has no link with cells at t and the corresponding reference cell is linked, a false negative link is counted. The same applies for a result cell merged at frame t that has no link with cells at $t+1$.

This procedure is applied for each pair of frames, but this does not tell us how these errors are distributed among the different cell lineages. This information is crucial for evaluating an algorithm, as recovering more error-free cell lineages can be more useful even if it makes more frame-pair-wise errors [27]. We therefore counted the number of error-free cell lineages, allowing for a user-defined tolerance to the frame at which mitosis is detected.

Lastly, we noticed that many errors arise from cells that are partially out of bounds. We believe that these errors can be easily removed automatically and should not be counted as errors. Therefore, for segmentation and tracking error metrics, the procedure simply ignores errors that are related to cells that touch edges. For the incomplete lineage metric, the procedure simply ignores errors that are related to lineages that contain at least one cell that touches an edge.

G. Evaluations

DeLTA 2.0 independently performs segmentation and tracking using two independent U-Net models, with *ad hoc* procedures specifically designed for rod-shaped bacteria, such as a skeletonization of the cell body shape to set a maximal weight at the center of the cell during training. EmbedTrack simultaneously performs segmentation and tracking using a single DNN, but it uses a total time window of only two frames $(t, t+1)$ and only forward predictions.

DiSTNet2D outperforms DeLTA 2.0 and EmbedTrack on all our metrics: segmentation errors, tracking errors, and incomplete lineages (Table I and Supplemental Material Videos S1–S3 for visualizing the segmentation and tracking results [24]). Notably, DiSTNet2D achieved perfect tracking accuracy on the bacterial data set. Using the Cell Tracking Challenge metrics [26] confirms the performance of DiSTNet2D (Table S1 [24]). Additionally, DiSTNet2D ran faster than the other two methods on both data sets. It is also noteworthy that DiSTNet2D effectively manages large displacements, as shown in Fig. S10 [24], in which displacements greater than the cell size are predicted.

TABLE II. Ablation experiments on data set PhC-C2DH-PA14 (the total number of cells is 123 057, links 145 306, and lineages 561). The DNN time window of 2 corresponds to a simplified version of the DNN that considers only frames $(t, t+1)$. The last case, which corresponds to DiSTNet2D, has a DNN time window of 15 frames ($N = 7$ considered frames and $\delta = 3$ using the subsampling definition presented in Fig. S1 [24]), included postprocessing, and achieved the best performance.

Ablations		Results		
Postprocessing	DNN time window	Segmentation error (%)	Tracking error (%)	Incomplete lineages (%)
No	2	0.65	0.00	10
No	15	0.47	0.00	5.3
Yes	2	0.50	0.00	0.89
Yes	15	0.24	0.00	0.53

We also evaluated the contribution of several components of our method by performing ablation experiments (Table II). Switching the DNN time window from 2 frames [a single frame pair $(t, t+1)$, as in EmbedTrack and DeLTA 2.0] to 15 frames, without postprocessing, increased segmentation performance and notably decreased the number of incomplete lineages. This shows that mid-term context is leveraged for segmentation and brings temporal consistency. While the two-frame version without postprocessing exhibits fewer segmentation errors compared to EmbedTrack (0.65% vs 1.2%), it also results in more incomplete lineages (10% vs 7.1%). This difference arises from the contrasting undersegmentation-to-oversegmentation ratios observed in the two methods, with undersegmentations having a more pronounced impact on incomplete lineages.

By leveraging temporal information across the entire video, postprocessing effectively corrected suspected errors (defined by merge links and split links that are not confirmed by predicted link multiplicity), leading to a substantial reduction in both objectwise segmentation errors and, more notably, lineagewise errors, enhancing the overall temporal coherence of predictions. We acknowledge that postprocessing might introduce new errors that will be propagated over entire tracks, but we consider this an acceptable drawback considering its great performance in terms of reducing incomplete lineages. We also concede that postprocessing works well because the DNN already makes few errors. Postprocessing and the DNN work synergistically, as evidenced by the degraded performance observed when either or both components are disabled, compared to when they are both operational.

We then tested how DiSTNet2D's performances in leveraging long-term information are affected by frame sampling. First, we varied how the seven considered frames were spread apart, by playing with the parameter δ, which changed the range of the DNN time window [Fig. 4(a)]. We found that a wider DNN time window improved the segmentation performance (the tracking efficiency was already very high, even for small δ). This suggests that DiSTNet2D benefits from having access to a longer temporal context. The accuracy over entire lineages was also improved with greater δ. However, this effect was eliminated after postprocessing, possibly because the number of remaining incomplete lineages is too small (lower than 5).

To further assess the influence of time sampling, we compared the performance of DiSTNet2D, EmbedTrack, and DeLTA 2.0 on subsampled evaluation data sets [Fig. 4(b)]. While the accuracy of segmentation stayed roughly the same for all methods across the tested range, the accuracy of tracking was sensitive to subsampling, both at the level of individual links or entire tracks. However, DiSTNet2D remained fairly accurate for tracking, unlike the other two methods which quickly lost their effectiveness. For example, DiSTNet2D performed just as well on the sixfold subsampled data set as the other two methods did on the full data set. Its robustness to subsampling can be explained by the awareness of mid-range temporal information and by the random subsampling performed during data augmentation (see the Supplemental Material, Sec. 3 [24]).

Overall, the ablation experiments and subsampling experiments confirm that leveraging temporal information improves the segmentation and tracking performance.

H. Showcase of DiSTNet2D's performance on biological systems

We demonstrate the potential of DiSTNet2D by applying it to bacterial and eukaryotic data sets.

1. Monolayer of bacterial cells

We measured the mean-squared displacement (MSD) of *Pseudomonas aeruginosa* cells at the surface of an agar gel, at a surface fraction of $\phi = 0.719$, where the cell monolayer appears "jammed." As shown in Sec. II G, DiSTNet2D was able to extract long tracks: 3404 1000-frame tracks out of an average of 3620 cells that were visible in one field of view. Track duration statistics indicates very few errors (Table S2 [24]). The MSD scales approximately with t, confirming diffusive behavior over three decades of time [Fig. 5(a)]. At lower surface fraction ($\phi = 0.466$), cells were more motile and a larger fraction of them left the field of view within the duration of the video (1413 1000-frame tracks out of an average of 2911 visible cells). Accordingly, the behavior at short timescales was overdiffusive but remained diffusive at longer timescales.

We also correlated the length of each cell with its speed. At low density, short cells moved faster than longer cells [Fig. 5(b)]. We hypothesize that this is a signature of single-cell motility, as viscous drag varies monotonously with the length of a rod [28]. This trend disappeared at high density as cells collectively blocked each other, regardless of their length.

2. Eukaryotic cells

The ability of DiSTNet2D to extract precise cell contours and long trajectories enables the correlation of migrating speed and direction with cellular shape. Early in this experiment, HBEC cells at low density typically migrate perpendicular to their major axis [Fig. 5(c)]. After a few rounds of divisions, cell density becomes higher, and collisions reorient the cells, leading to a less peaked angle distribution.

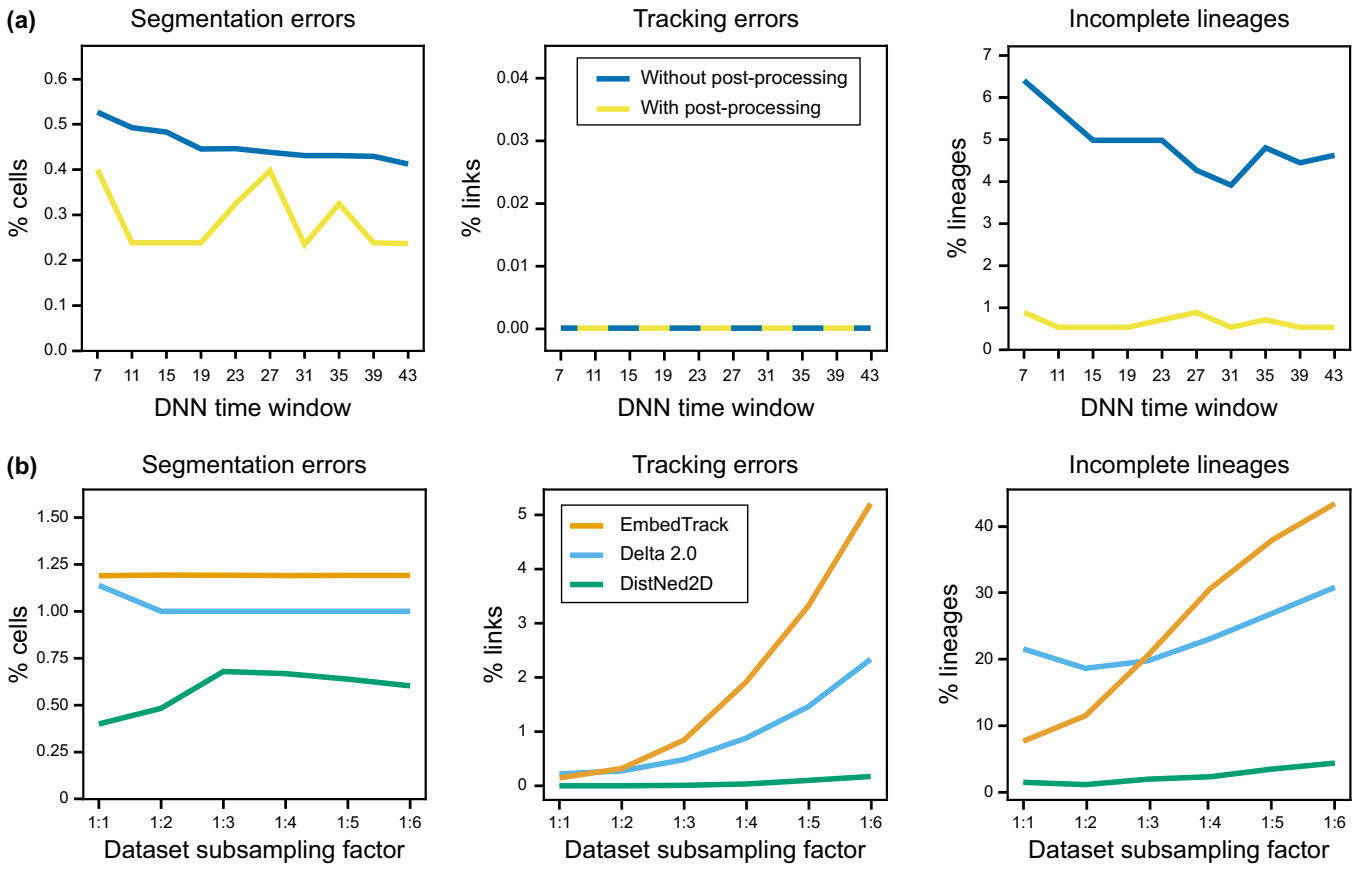

FIG. 4. Temporal insights of the DiSTNet2D's performance, calculated with data set PhC-C2DH-PA14. (a) Influence of DNN time window in predictions. DNN time window was varied by changing the gap between considered frames (δ) from 1 to 10, while keeping the number of considered frames constant ($N = 7$). (b) Robustness to acquisition subsampling. We compared the three methods (trained with the complete training data set) on computationally generated time-subsampled versions of the evaluation data set. Note that for DiSTNet2D, δ was set to 1 for all points, which explains the difference in segmentation errors compared to Table I, where $\delta = 3$.

Interestingly, while the largest displacements are perpendicular to the cell body major axis at low density, the trend reverses at high density as some cells move along other cells [Fig. 5(d)].

III. DISCUSSION

DiSTNet2D introduces several novel components to address the challenges of segmentation and tracking in bioimages. These components include a novel segmentation method that combines EDM and GDCM, improving the separation of adjacent cells, while being able to segment a wide range of cell morphologies. DiSTNet2D also employs a novel approach to predicting backward and forward tracking proxies to handle cell division and fusion events and improve robustness of tracking to under- or oversegmentation errors. Long-range temporal context is leveraged in a novel postprocessing stage that corrects incorrect merge and split links, relying on the predicted link multiplicity. This stage strongly reduces the lineagewise error rate by analyzing entire lineages and correcting them when necessary. This strategy is efficient even if errors are also generated at the lineage scale as long as the cellwise error rate before postprocessing is very low.

We developed a series of carefully chosen innovations for the implementation of the DiSTNet2D algorithm. Tracking and segmentation proxies are predicted by a DNN with a novel architecture designed for leveraging mid-range temporal information for segmentation, while being optimized for size and training efficiency. This range is further increased thanks to a gapped input strategy at no additional GPU memory cost. Following Ref. [19], we tried to introduce an attention layer at the pair blender module, but this did not improve performance. This is likely because the two data sets used in the study only contained short-range displacements, which could be captured by dilated convolutions. However, an attention layer may be useful for processing data sets with longer-range displacements or displacements that depend on location, such as in microfluidic devices [29,30]. The loss function was chosen in order to effectively guide the training process by generating gradients of similar magnitude between segmentation and tracking proxies, regardless of cell size and displacement amplitude. Carefully designed data augmentation allows generalization to diverse imaging conditions without requiring retraining. In particular, we introduced on-the-fly random frame subsampling, which improves robustness to changes in acquisition rate, but also increases tracking performance by diversifying displacements during training. Each of these components plays a crucial role in

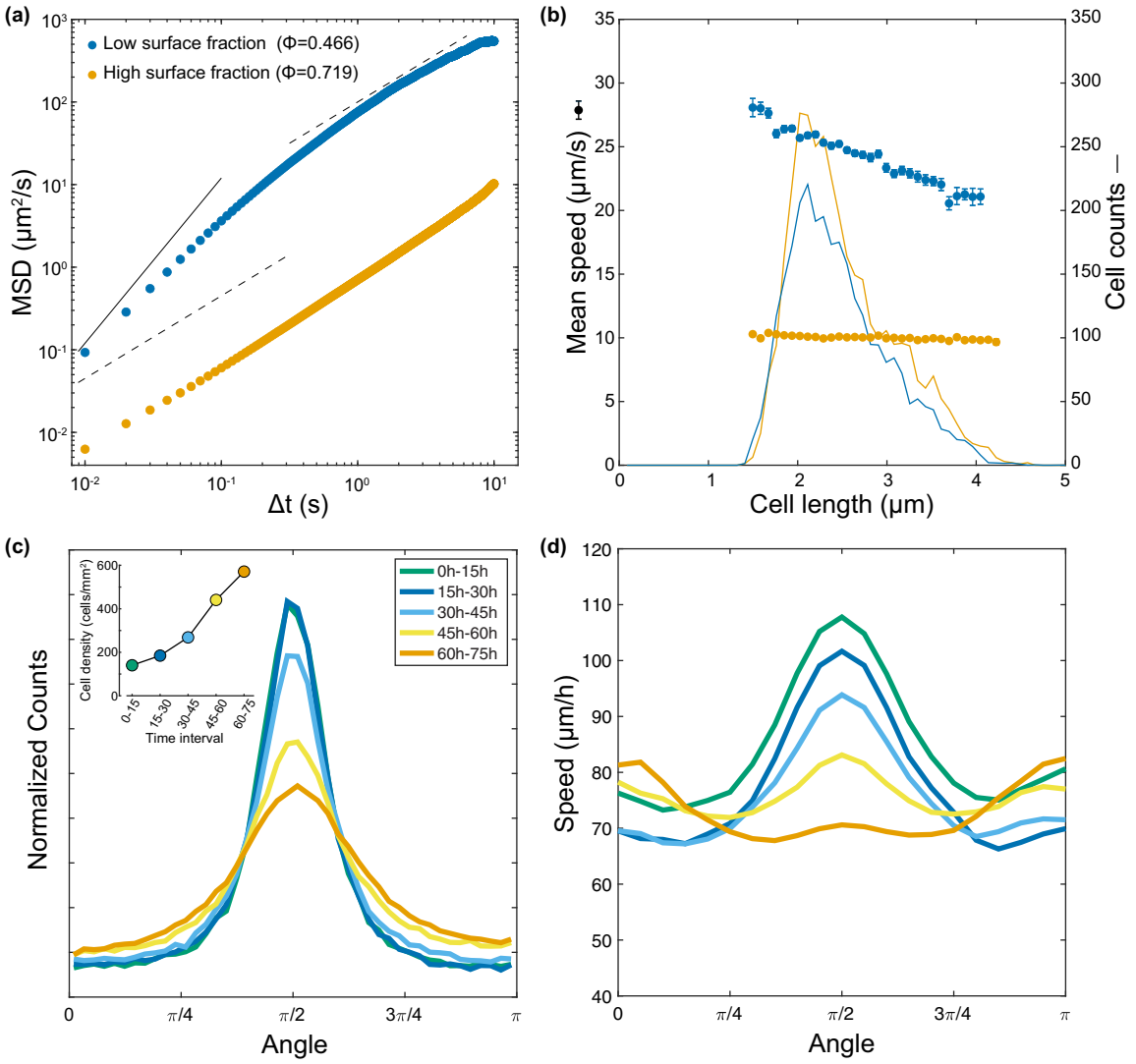

FIG. 5. (a) Mean-squared displacement measured on bacterial cells at low surface fraction ($\phi = 0.466$, blue dots) and high surface fraction ($\phi = 0.719$, orange dots). Guide lines have exponent 1 (dashed lines) and 2 (solid line). (b) Cell speed as a function of cell length, using the same color code. Error bars are standard errors of the mean (error bars are hidden behind dots for the high surface fraction data set). Each dot is the mean speed of all cells binned by length, with a bin size of 0.088 μm (one pixel). Averaging is weighted by the duration of trajectories. The solid lines represent the number of cells in each bin, weighted by the duration of trajectories (a value of 1 is attributed to a cell tracked for 1000 frames). (c) Histogram of angles between the velocity vector and the major axis of the HBEC cell body (obtained by ellipse fitting), for five time intervals (0–15 h, 15–30 h, 30–45 h, 45–60 h, 60–75 h). Each interval includes $N = 40\,939$, 53 284, 76 305, 115 860, and 138 749 data points (respectively). No chirality is measured in the data. Inset: Average cell density for each time interval. (d) Norm of the velocity vector with respect to the angle between the velocity vector and the major axis of the HBEC cell body, for the same five time intervals.

enhancing the performance of DiSTNet2D, making it a powerful tool for analyzing biological processes involving cell movement.

We demonstrated the performance of DiSTNet2D on two different data sets: a bacterial data set, where cells are densely packed, have similar shape, and only differ in size, and a eukaryotic data set where cells are sparse, change shape, and divide. Some methods are specifically designed for a precise type of data set, with *ad hoc* procedures. This is the case for DeLTA 2.0, which was designed for bacterial data sets. Despite its specificity, DeLTA 2.0 underperforms compared to DiSTNet2D. A major strength of DiSTNet2D is its ability to leverage both mid- and long-range temporal context, unlike other considered methods; DeLTA 2.0 performs segmentation and tracking separately and does not try to leverage temporal information for segmentation. EmbedTrack performs segmentation and tracking jointly, but its DNN does not blend temporal information and thus does not make it available for all frames, thus segmentation is performed on each frame independently. In both DeLTA 2.0 and EmbedTrack, segmentation does not have access to temporal context.

We believe DiSTNet2D can be used to segment and track many types of extended objects in a 2D setting: living or inert, with changing or fixed shape, undergoing division or fusion, at any surface density. Any type of imaging modalities can be used: fluorescence, phase contrast, brightfield, etc.

Nonetheless, several areas may benefit from further development. First, segmentation and tracking of punctual objects,

such as those found in single-molecule imaging, may pose a significant challenge as our segmentation method is based on EDM and GDCM which are not informative for nonextended objects. Second, occlusions (where objects of the same class hide each other) or disappearances (where objects move out of the field of view or out of focus) are a limit for our method. They both disrupt tracking (as they yield discontinuity of object detection across frames), and occlusions disrupt segmentation as well (as they can lead to merged representations of distinct objects, complicating accurate delineation). In such scenarios, establishing a precise ground truth can be challenging, if not impossible. Future developments will focus on enhancing DiSTNet2D's robustness to these complexities, aiming to accurately handle both occlusions and disappearances. Finally, while extending DiSTNet2D to three-dimensional data sets is a compelling avenue for future work, it would introduce complexities due to significantly increased memory demands. This would necessitate an optimization of the network architecture and training procedures to handle the additional data volume efficiently.

Like any supervised DNN-based method, training DiSTNet2D requires a training data set, which can be cumbersome to generate. While data augmentation during training makes DiSTNet2D robust to some changes in data set, its direct application to substantially different data sets may not yield optimal results. In such instances, a fine-tuning strategy can potentially reduce the amount of new data required and simplify the retraining process. If this is not possible, a *de novo* data set needs to be created. To assist in the development of new data sets, we recommend using the classical segment-then-track approach. The BACMMAN graphical user interface supports the entire pipeline needed for training data set creation. It includes a DNN-based segmentation method with very low annotation requirements [31], alongside various automated tracking methods (such those in Trackmate [32], an ImageJ plug-in that BACMMAN is connected to). Furthermore, BACMMAN streamlines training, manual correction, retraining, distribution of the trained DNN weights to other laboratories, and data export as tabular data or as label images, making it more accessible and practical for a broad range of researchers. Moreover, BACMMAN is able to handle multiple classes of object simultaneously, for instance cell membrane and cell nucleus, cells and foci, head and tail. This could be of particular interest in microbiology, cell biology, soft matter, active matter, but also ethology. Together, DiSTNet2D and BACMMAN form a versatile framework for analyzing biological processes involving cell movement.

All source code (PYTHON) for the training of DiSTNet2D is available at Ref. [33]. All source code (JAVA) for BACMMAN and the computation of DiSTNet2D predictions within BACMMAN is available at Ref. [34].

ACKNOWLEDGMENTS

This project has received financial support from the ANR X-BACAMAT project (Grant No. ANR-21-CE30-0025). We are grateful to Pascal Silberzan and Charles Ollion for helpful discussions.

J.O. and M.D. were involved in conceptualization, planning, and supervised the work. J.O. developed and implemented DiSTNet2D and BACMMAN. M.M. and M.D. generated the bacterial data sets. E.V. and C.G. generated the eukaryotic data sets. M.M. analyzed the bacterial data and E.V. analyzed the eukaryotic data. M.M., M.D., and J.O. participated in the manual correction of training data sets. J.O. and M.M. performed training, evaluations, and ablation experiments. All authors wrote the manuscript.

J.O. is the founder and director of the company SABILab. Other authors declare no competing interest.

[1] A. Boquet-Pujadas, J.-C. Olivo-Marin, and N. Guillén, Bioimage analysis and cell motility, Patterns **2**, 100170 (2021).

[2] P. Friedl, and D. Gilmour, Collective cell migration in morphogenesis, regeneration, and cancer, Nat. Rev. Mol. Cell Biol. **10**, 445 (2009).

[3] C. D. Paul, P. Mistriotis, and K. Konstantopoulos, Cancer cell motility: Lessons from migration in confined spaces, Nat. Rev. Cancer **17**, 131 (2017).

[4] S. SenGupta, C. A. Parent, and J. E. Bear, The principles of directed cell migration, Nat. Rev. Mol. Cell Biol. **22**, 529 (2021).

[5] N. Wadhwa and H. C. Berg, Bacterial motility: Machinery, and mechanisms, Nat. Rev. Microbiol. **20**, 161 (2022).

[6] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing, and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part II* (Springer, Berlin, 2015), pp. 234–241.

[7] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137 (2016).

[8] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, Cell detection with star-convex polygons, in *Medical Image Computing, and Computer Assisted Intervention—MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II* (Springer, Berlin, 2018), pp. 265–273.

[9] C. Payer, D. Štern, M. Feiner, H. Bischof, and M. Urschler, Segmenting, and tracking cell instances with cosine embeddings, and recurrent hourglass networks, Med. Image Anal. **57**, 106 (2019).

[10] P. Naylor, M. Laé, F. Reyal, and T. Walter, Segmentation of nuclei in histopathology images by deep regression of the distance map, IEEE Trans. Med. Imaging **38**, 448 (2018).

[11] D. Neven, B. De Brabandere, M. Proesmans, and L. V. Gool, Instance segmentation by jointly optimizing spatial embeddings, and clustering bandwidth, in *Proceedings of the IEEE/CVF Conference on Computer Vision, and Pattern Recognition* (IEEE, Piscataway, NJ, 2019), pp. 8837–8845.

[12] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, Cellpose: A generalist algorithm for cellular segmentation, Nat. Methods **18**, 100 (2021).

[13] K. J. Cutler, C. Stringer, T. W. Lo, L. Rappez, N. Stroustrup, S. B. Peterson, P. A. Wiggins, and J. D. Mougous, Omnipose: A high-precision morphology-independent solution for bacterial cell segmentation, Nat. Methods **19**, 1438 (2022).

[14] O. M. O'Connor, R. N. Alnahhas, J.-B. Lugagne, and M. J. Dunlop, DeLTA 2.0: A deep learning pipeline for quantifying single-cell spatial, and temporal dynamics, PLoS Comput. Biol. **18**, e1009797 (2022).

[15] T. Ben-Haim and T. Riklin Raviv, Graph neural network for cell tracking in microscopy videos, in *Proceedings of the 17th European Conference on Computer Vision–ECCV 2022, Part XXI* (Springer, Berlin, 2022), pp. 610–626.

[16] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, Siam R-CNN: Visual tracking by re-detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision, and Pattern Recognition* (IEEE, Piscataway, NJ, 2020), pp. 6578–6588.

[17] Y. Chen, Y. Song, C. Zhang, F. Zhang, L. O'Donnell, W. Chrzanowski, and W. Cai, Celltrack R-CNN: A novel end-to-end deep neural network for cell segmentation, and tracking in microscopy images, in *Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* (IEEE, Piscataway, NJ, 2021), pp. 779–782.

[18] J. Hayashida, K. Nishimura, and R. Bise, MPM: Joint representation of motion, and position map for cell tracking, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2020), pp. 3823–3832.

[19] J. Ollion and C. Ollion, DistNet: Deep tracking by displacement regression: Application to bacteria growing in the mother machine, in *Proceedings of the International Conference on Medical Image Computing, and Computer-Assisted Intervention* (Springer, Berlin, 2020), pp. 215–225.

[20] K. Löffler and R. Mikut, Embedtrack-simultaneous cell segmentation, and tracking through learning offsets, and clustering bandwidths, IEEE Access **10**, 77147 (2022).

[21] J. Hayashida, K. Nishimura, and R. Bise, Consistent cell tracking in multi-frames with spatio-temporal context by object-level warping loss, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (IEEE, Piscataway, NJ, 2022), pp. 1727–1736.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. **30**, 6000 (2017).

[23] J. Ollion, M. Elez, and L. Robert, High-throughput detection, and tracking of cells, and intracellular spots in mother machine experiments, Nat. Protocols **14**, 3144 (2019).

[24] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PRXLife.2.023004 for additional information about the experimental and numerical methods, which includes Refs. [31,32,35–39].

[25] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, NIH Image to ImageJ: 25 years of image analysis, Nat. Methods **9**, 671 (2012).

[26] P. Matula, M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de Solórzano, and M. Kozubek, Cell tracking accuracy measurement based on comparison of acyclic oriented graphs, PLoS ONE **10**, e0144959 (2015).

[27] M. Maška, V. Ulman, P. Delgado-Rodriguez, E. Gómez-de Mariscal, T. Nečasová, F. A. Guerrero Peña, T. I. Ren, E. M. Meyerowitz, T. Scherr, K. Löffler *et al.*, The cell tracking challenge: 10 years of objective benchmarking, Nat. Methods **20**, 1010 (2023).

[28] S. Kamdar, D. Ghosh, W. Lee, M. Tătulea-Codrean, Y. Kim, S. Ghosh, Y. Kim, T. Cheepuru, E. Lauga, S. Lim, and X. Cheng, Multiflagellarity leads to the size-independent swimming speed of peritrichous bacteria, Proc. Natl. Acad. Sci. USA **120**, e2310952120 (2023).

[29] L. Robert, J. Ollion, J. Robert, X. Song, I. Matic, and M. Elez, Mutation dynamics, and fitness effects followed in single cells, Science **359**, 1283 (2018).

[30] A. Dal Co, S. van Vliet, D. J. Kiviet, S. Schlegel, and M. Ackermann, Short-range interactions govern the dynamics, and functions of microbial communities, Nat. Ecol. Evol. **4**, 366 (2020).

[31] L. Ludvikova, E. Simon, M. Deygas, T. Panier, M.-A. Plamont, J. Ollion, A. Tebo, M. Piel, L. Jullien, L. Robert *et al.*, Near-infrared co-illumination of fluorescent proteins reduces photobleaching, and phototoxicity, Nat. Biotechnol. 1 (2023).

[32] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri, Trackmate: An open, and extensible platform for single-particle tracking, Methods **115**, 80 (2017).

[33] https://github.com/jeanollion/distnet2d/

[34] https://github.com/jeanollion/bacmman/

[35] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, Deterministic edge-preserving regularization in computed imaging, IEEE Trans. Image Process. **6**, 298 (1997).

[36] J. Friedman, R. Tibshirani, and T. Hastie, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer-Verlag, New York, 2009).

[37] G. V. Tulder, Zenodo, elasticdeform: Elastic deformations for N-dimensional images, https://doi.org/10.5281/zenodo.7102577 (2021).

[38] D. P. Kingma, and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[39] M. Deforet, Long-range alteration of the physical environment mediates cooperation between *Pseudomonas aeruginosa* swarming colonies, Environ. Microbiol. **25**, 1451 (2023).