Graph Neural Network for Cell Tracking in Microscopy Videos

Tal Ben-Haim and Tammy Riklin Raviv

School of Electrical and Computer Engineering, Ben-Gurion University
{benhait@post, rrtammy@ee}.bgu.ac.il

Abstract. We present a novel graph neural network (GNN) approach for cell tracking in high-throughput microscopy videos. By modeling the entire time-lapse sequence as a direct graph where cell instances are represented by its nodes and their associations by its edges, we extract the entire set of cell trajectories by looking for the maximal paths in the graph. This is accomplished by several key contributions incorporated into an end-to-end deep learning framework. We exploit a deep metric learning algorithm to extract cell feature vectors that distinguish between instances of different biological cells and assemble same cell instances. We introduce a new GNN block type which enables a mutual update of node and edge feature vectors, thus facilitating the underlying message passing process. The message passing concept, whose extent is determined by the number of GNN blocks, is of fundamental importance as it enables the 'flow' of information between nodes and edges much behind their neighbors in consecutive frames. Finally, we solve an edge classification problem and use the identified active edges to construct the cells' tracks and lineage trees.

We demonstrate the strengths of the proposed cell tracking approach by applying it to 2D and 3D datasets of different cell types, imaging setups, and experimental conditions. We show that our framework outperforms current state-of-the-art methods on most of the evaluated datasets. The code is available at our repository[1].

Keywords: Cell Tracking · Graph Neural Network · Microscopy Videos

1 Introduction

Time-lapse microscopy imaging is a common tool for the study of biological systems and processes. However, probing complex and dynamic cellular events requires quantitative analysis at the single cell level of a huge amount of data, which far surpasses human annotators' abilities. Therefore, automatic cell tracking which aims to identify and associate instances of the same biological cells and their offspring along microscopy sequences is an active field of study.

The automatic construction of cell trajectories is a challenging problem since same-sequence cells often have similar visual traits yet individual cells may

---

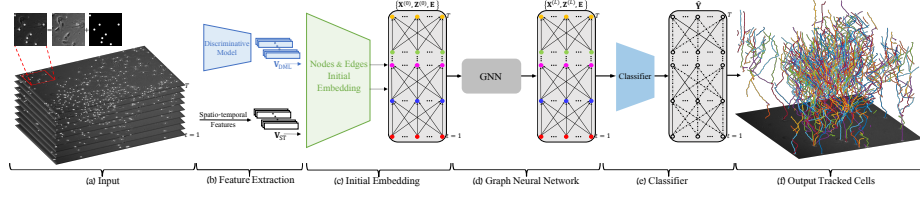[1] For our repository, refer to `https://github.com/talbenha/cell-tracker-gnn`.

Fig. 1. An outline of the proposed cell tracking framework. (a) The input is composed of a live cell microscopy sequence of length T and the corresponding sequence of label maps. (b) Each cell instance in the sequence is represented by a feature vector which includes DML and spatio-temporal features. (c) The entire microscopy sequence is encoded as a direct graph where the cell instances are represented by its nodes and their associations are represented by the graph edges. Each node and edge in the graph has its own embedded feature vector. (d) These feature vectors are encoded and updated using Graph Neural Network (GNN). The GNN (which is illustrated in Fig. 2(a)) is composed of L message passing blocks which enable an update of edge and node features by their L-th order neighbors (i.e., cell instances which are up to L frames apart). (e) The GNN's edge feature output is the input for an edge classifier network which classifies the edges into active (solid lines) and non-active (dashed lines). During training, the predicted classification $\hat{\mathbf{Y}}$ is compared to the GT classification $\mathbf{Y}$ for the loss computation. Since all the framework components are connected in an end-to-end manner the loss backpropogates throughout the entire network. (f) At inference time, cell tracks are constructed by concatenating sequences of active edges that connect cells in consecutive frames.

change their appearance or even divide over the course of time. In addition, a cell may temporarily occlude another, exit and reenter the frame field of view. On top of these, frequent mitotic events, the high cell density, high cell migration rate and low frame rate render cell tracking even more difficult.

Classical cell tracking methods include cell association algorithms which solve the tracking problem in a temporarily local manner, and global frameworks which aim to solve cell tracking for an entire time-lapse microscopy sequence by simultaneous extraction of entire tracks. Frame-to-frame association algorithms connect cell instances in consecutive frames based on their visual traits and pose similarity [4,21] as well as cell motion prediction [1,2]. Methods that look for globally optimal cell tracking solutions usually represent a microscopy sequence by a graph, encoding detected cell instances and their potential links to neighboring frames by the graph nodes and edges, respectively. In most cases, linear programming and combinatorial algorithms were used to extract, multiple, non-overlapping cell trajectories [5,17,18,20,27,30,31,35,36].

Despite their increasing popularity and state-of-the-art performance in various fields, deep learning approaches for tracing cells in a sequence became prevalent only lately. Many recent methods use convolutional neural networks (CNNs) for cell segmentation, but the construction of cell trajectories is still performed using classical methods e.g., [15,34,37]. A pioneering complete deep learning method for cell tracking was proposed in [14] however it is limited to single cell

tracking. In [11] two separate U-nets [32] were proposed for cell likelihood detection and movement estimation. In a later work by that group, a single U-Net was exploited to perform both tasks at once [12]. A further elaboration was proposed in [26] using both backward and forward propagation between nearby frames. Both [29] and [38] presented recurrent neural network (RNN) approaches for the time-series analysis of microscopy videos. Other recent cell tracking approaches include a deep reinforcement learning method [41] and a pipeline of Siamese networks [8,28]. All of these deep learning approaches predict frame-to-frame cell associations rather than providing a global tracking solution. In contrast, for the first time (to the best of our knowledge), we present a graph neural network (GNN) framework for the tracking of multiple cell instances (represented by the graph nodes and associated by its edges) in high-throughput microscopy sequences. This approach allows us to simultaneously extract entire cell tracks for the construction of complete lineage trees.

A GNN is a neural network that operates on data structured as a graph. It is designed to capture graph dependencies via message passing between its nodes. Specifically, node feature vectors are computed by recursively aggregating and transforming feature vectors of their neighbors [10]. The number of message passing neural network (MPNN) blocks determines the extent of the long range interactions of nodes and edges in detached frames. Many GNN variants with different neighborhood aggregation schemes have been proposed and applied to node classification, including Graph Convolutional Network (GCN) [19] and Graph Attention Network (GAT) [40]. The latter is an extension of the GCN in which the weighting of neighboring nodes is inferred by an attention mechanism. In a very recent framework called Pathfinder Discovery Network Convolution (PDN-Conv), it has been proposed to use incoming edge features to weight their influence on the nodes they connect [33].

We extend the PDN-Conv block and design a unique MPNN paradigm based on an edge encoding network and a mutual update mechanism of node and edge features that facilitate the message passing process. Moreover, since we formulate the construction of cell trajectories by a problem of finding paths in a graph, we can address it by solving a binary edge classification problem, where 'active' edges are those which connect cell instances in consecutive frames that are associated with the same biological cell. The update of the edge features is therefore an important aspect of our method. In [7,43] GNNs have been used with different message passing schemes for vehicle and pedestrian tracking. Yet, cell tracking poses different challenges than tracking cars or people. Cell dynamics appear completely random, as there are no paths or roads neither common motion directions or speed. Moreover, cells within the same sequence may look very similar, yet they change their appearance along the sequence and even divide. Furthermore, in many cases, due to frequent mitotic events (cell divisions), cell population increases very rapidly such that frames, mainly toward the end of the sequence, become extremely dense and cluttered. There might be hundreds of cell instances in a frame, and therefore, frequent cell overlap and occlusions [9,25].

Cell overlap, visual similarity, and gradual change in appearance often render manually selected features such as position, intensity, and shape insufficient for accurate tracking. To address some of these difficulties, Arbelle et al. exploited the Kalman filter to predict changes in cell shapes in addition to their estimated pose and motion [2]. Some CNN-based methods learn cell features implicitly but mainly for the purpose of cell segmentation and separation of nearby cell instances. In this work, we use a deep metric learning (DML) technique to learn discriminative cell instance features. Specifically, we utilize the multi-similarity loss as proposed by [42] to differentiate between instances of different cells and to associate instances of the same biological cells. The DML features of each cell instance are the input to the proposed GNN.

Fig. 1 presents the outline of our end-to-end cell tracking framework. The input to our framework is composed of a time-lapse microscopy sequence of living cells and their annotations. The outputs are the cell trajectories and lineage trees. The feature embedding networks, the GNN, and the edge classifier are connected such that the classifier loss backpropagates throughout the entire network compound. Our method is based on several key steps and contributions: **1)** We address cell tracking by the simultaneous construction of all cell trajectories in the entire sequence. **2)** We are the first to use deep metric learning to learn features that distinguish between instances of different biological cells and to assemble same cell instances. **3)** We represent cell instances and their features as nodes in a direct graph and connect them by edges which represent their potential associations in consecutive frames. **4)** We are the first to apply a GNN and a message passing mechanism as a solution to the cell tracking task. Specifically, we look for paths in a graph that models the studied microscopy sequence, where each path represents a cell trajectory. **5)** We address cell tracking as an edge classification problem for identifying active and non-active edges. We, therefore, focus on edge features and introduce a new GNN block that enables mutual node and edge feature update.

We demonstrate our cell tracking framework by applying it to a variety of datasets, both 2D and 3D of different cell types, imaging conditions, and experimental setups. We show that our framework outperforms current state-of-the-art methods on most of the evaluated datasets. Specifically, we perform cell tracking on the challenging C2C12 dataset [9] and show that our method outperforms five other competing methods and achieves state-of-the-art results. We also competed in the Cell Tracking Challenge (CTC) [23,39] as BGU-IL (5). The table published on the CTC website shows that our method was ranked first for three different datasets and second for another one.

2   Method

Addressing the problem of cell tracking via GNN enables a simultaneous construction of all tracks in a sequence. Here, we present the method's building blocks and our contributions. The reader is also referred to Fig. 1 for a visualization of the method outline. We consider cell tracking as a global problem

applied to the entire sequence at once. This tracking model is formulated in Section 2.1. The representation of the entire microscopy sequence as a graph, where cell's instances are encoded by its nodes and their associations by its edges, is introduced in Section 2.2. This representation allows us to address cell tracking as an edge classification problem. In Section 2.3 we discuss cell instance feature extraction using deep metric learning. In Section 2.4 we present the proposed GNN blocks which are designed to facilitate a message passing strategy and introduce our innovative mechanism for mutual node and edge feature update. The edge classifier and the corresponding loss function are presented in Section 2.5. Finally, in Sections 2.6 and 2.7 we discuss our approach which utilizes the proposed representation for inference and mitosis detection, respectively.

2.1   Cell Tracking Problem Formulation

The input to our framework is a sequence of frames $\{I_t\}_{t=1}^T$ and its corresponding sequence of label maps $\{\mathcal{L}_t\}_{t=1}^T$, where T is the length of the sequence. Each frame in the sequence $I_t \colon \Omega \to \mathbb{R}+$ is a gray-scale image presenting K_t cell instances, where Ω denotes a 2D or 3D image domain. The associated label map $\mathcal{L}_t \colon \Omega \to \{l^0, l^1, \ldots, l^{K_t}\}$ partitions Ω into K_t connected components, each corresponds to an individual cell, and a background which is labeled by $l^0 = 0$ regardless of the frame number. We note that the cell labeling in each frame is arbitrary and the number of cell instances K_t may vary from frame to frame due to entrance/exit of cells to/from the field of view and mitotic events. We further define by $s_t^k = \{\forall \rho \in \Omega | \mathcal{L}_t(\rho) = l^k\}$ the set of pixels/voxels ρ that are associated with the $k-$th cell in frame I_t. Note that s_t^k can either include all the pixels/voxels that belong to a particular cell or a representative subset, depending whether $\mathcal{L}_t$ is the segmentation map of the cells in the frame or represents a set of K_t markers at approximately the cell centers.

Let N denote the total number of biological cells that were depicted in the entire sequence, and let $c_t^n = \{s_t^n, t\}$ denote a cell object which represents an instance of the $n-$th cell in frame t. We further denote by t_{init}^n and t_{fin}^n the first and the last time points (respectively) in which a cell is depicted in a sequence n. Note that $1 \le t_{\text{init}} \le t_{\text{fin}} \le T$. To avoid confusion between the indices of the actual cells and their instances in the frame sequence, we define a cell function ψ which, given a cell instance, returns the index of its corresponding biological cell. Note that if $\psi(c_t^k) = \psi(c_{t+1}^l)$ than c_t^k and c_{t+1}^l are instances of the same cell. We aim to compose a set of N trajectories $\{\mathcal{T}_n\}_{n=1}^N$, where $\mathcal{T}_n = \{c_{t_{\text{init}}}^n, \ldots, c_{t_{\text{fin}}}^n\}$ is the maximal-length sequence of associated cell instances in consecutive frames that correspond to the same biological cell. Here, we set $c_t^n \equiv c_t^k$ for different values of k along the sequence if $\psi(c_t^k) = n$.

We assume that cells are depicted continuously; i.e., $t_{i+1} = t_i + 1$. Therefore, if a particular cell disappears due to an occlusion or a temporary exit from the field of view, its reappearance initiates a new track and it gets a new cell index. To identify mitosis we define a parent function $P \colon \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that $P(n) = n'$ when n' is the parent of cell n and $P(n) = 0$ if the cell's appearance is not a result of a mitotic event. For convenience we set $n' < n$.

Note that a cell cannot be a parent of itself, therefore $n' \neq n$. A complete trajectory object of the $n-$th biological cell is defined as follows, where $\mathcal{T_n} = \{\mathcal{T}_n, P(n), t_{\text{init}}^n, t_{\text{fin}}^n\}$. The cell tracking task goal is to find the set $\{\mathcal{T_1}, ..., \mathcal{T_N}\}$ that best explains the observations.

2.2 Graph Formulation

We use a direct, acyclic graph to model cell-to-cell associations in microscopy sequences. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ define a graph represented by its vertices (nodes) $\mathcal{V}$ and edges (links) $\mathcal{E}$. Let $M = \sum_{t=1}^{T} K_t$ represent all cell instances in the entire frame sequence. A graph representation of cells and their associations is composed of $|\mathcal{V}| = M$ vertices, where each node $\nu_i \in \mathcal{V}$, $i = 1, \dots, M$ represents a single cell instance $c_{t=\tau}^k$. For convenience, we can set $i = \sum_{t=1}^{\tau-1} K_t + k$.

An edge $e_{i,j} \in \mathcal{E}$ represents a potential association between a pair of vertices ν_i, ν_j, representing cell instances in consecutive frames. To reduce the number of edges, we connect a pair of cells only if their spatial Euclidean distance is within a neighborhood region, which is calculated based on the training set as detailed in Section A.1 in the Appendix.

We address cell tracking as an edge classification problem. The desired output are labeled sets of edges defined by an edge function $Y \colon \mathcal{E} \to \{0, 1\}$. Let ν_i, ν_j represent cell instances denoted by c_t^k and c_{t+1}^l, respectively.

$$Y(e_{i,j}) = y_{i,j} = \begin{cases} 1, & \text{if } \psi(c_t^k) = \psi(c_{t+1}^l) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Accurate prediction of $Y(e_{i,j})$ for the complete set of graph edges provides the entire cell lineage associated with the observed microscopy sequence. A cell trajectory $\mathcal{T}_n$ can be either defined by a sequence of cell instances represented by the graph's vertices $\{\nu_{i_1}, \nu_{i_2}, \dots \nu_{i_n}\}$, or by a sequence of edges $\mathbf{e}_n = \{e_{i_1 i_2}, \dots, e_{i_{n-1} i_n}\}$, that connect cell instances in consecutive frames, where $\{e_{ij} \in \mathbf{e}_n \mid Y(e_{ij}) = 1\}$.

We assume that a non-dividing cell instance has at most a single successor while a cell that undergoes mitosis may have two successors and even more (in rare occurrences). Ideally, if two (or more) different nodes in a frame $\nu_j \neq \nu_{j'}$ are connected to the same node ν_i in a previous frame, i.e.; $Y(e_{i,j}) = Y(e_{i,j'}) = 1$, then we can assume that $P(\psi(c_{t+1}^l)) = P(\psi(c_{t+1}^{l'})) = \psi(c_t^k)$ and detect a mitosis event. In practice, often the visual features of daughter cells differ from those of their parent and an additional process is required to identify and validate parent-daughter relations. The complete representation of the proposed graph is defined by the following attribute matrices: i) A node feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d_\mathcal{V}}$ with $d_\mathcal{V}$ features per node. ii) A graph connectivity matrix $\mathbf{E} \in \mathbb{N}^{2 \times |\mathcal{E}|}$ which represents all possible linked cell indices from source to target. iii) An edge feature matrix $\mathbf{Z} \in \mathbb{R}^{|\mathcal{E}| \times d_\mathcal{E}}$, where each row in $\mathbf{Z}$ consists of $d_\mathcal{E}$ features of an edge e_{ij} in the graph. We aim to predict $\hat{\mathbf{Y}} \in \mathbb{R}_{[0,1]}^{|\mathcal{E}| \times 1}$ that represents the probability to represent an actual cell association. Next, we present the initial graph embedding.

2.3 Feature Extraction

The success of cell tracking algorithms depends on correct associations of instances of the same biological cells. We consider instances of the same cell as members of the same class. Altogether, we have N mutually exclusive classes. **Deep Metric Learning (DML) Features.** We use *deep metric learning* to learn cell feature embeddings that allow us to assemble instances of the same biological cells and distinguish between different ones. For this purpose, we use the cell segmentation maps or marker annotations to crop each frame into sub-images of all cell instances. Following [42] we use a *hard mining* strategy and a *multi-similarity loss* function to train a ResNet network [13] to predict such embeddings. Specifically, we generate batches of cell sub-images, where each is composed of m same-class instances from κ classes. Since the cell's appearance gradually changes during the sequence we perform the *m-per-class* sampling [24] using temporally adjacent frames. To calculate the loss, we consider pairs of within-class and between-class cell instances which form the positive and negative examples, respectively. To conduct the hard mining, an affinity matrix $\mathbb{A} \in \mathbb{R}^{m\kappa \times m\kappa}$ is constructed based on the cosine similarity function applied to the 'learned' feature vectors of each of the pairs. A column i in $\mathbb{A}$ presents the proximity of all the batch instances to the $i-$th instance. An instance j forms a *hard negative* example with i if i and j are not within the same class and j is more similar to i than another instance k that form a positive example with it. In the same manner, an instance l forms a *hard positive* example with i if i and l are within the same class and l is less similar to i than another instance r that form a negative example with it. The multi-similarity loss is a measure of the *hard positive* and *hard negative* examples. When there are no such examples the embedded feature vectors are adequately clustered. The node feature vectors extracted using DML approach are denoted in a matrix form as follows: $\mathbf{V}_{\mathrm{DML}} \in \mathbb{R}^{|\mathcal{V}| \times d_{\mathrm{DML}}}$, where d_{DML} denotes the number of DML features.
Spatio-temporal Features. We also consider temporal and global spatial cells' features which include the coordinates of the cell center, its frame number, and intensity statistics (minimum, maximum, and average). In the case where we have an instance segmentation mask, the cell's area, the minor and major axes of a bounding ellipse, and bounding-box coordinates are also considered. We denote by $\mathbf{V}_{\mathrm{ST}} \in \mathbb{R}^{|\mathcal{V}| \times d_{\mathrm{ST}}}$ the spatio-temporal (ST) feature matrix, which is composed of the d_{ST}-dimensional feature vectors of all nodes.
Initial Edge and Node Features. The complete feature matrix of all nodes in the graph includes both the learned and the spatio-temporal features and is denoted by $\mathbf{V}_{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}| \times d_{\mathcal{V}}}$, where, $d_{\mathcal{V}} = d_{\mathrm{ST}} + d_{\mathrm{DML}}$. It is generated by a concatenation of $\mathbf{V}_{\mathrm{DML}}$ and $\mathbf{V}_{\mathrm{ST}}$. Having $\mathbf{V}_{\mathcal{V}}$ we construct the initial edge feature matrix $\mathbf{V}_{\mathcal{E}}$ using the *distance & similarity* operation defined in Eq. 3. Since $\mathbf{V}_{\mathrm{DML}}$ and $\mathbf{V}_{\mathrm{ST}}$ are from different sources and are in different scales, we homogenize them and reduce the complete feature vector dimension via mapping by multi-layer perceptron (MLP) networks. These MLPs are connected to the proposed GNN (see Section 2.4) in an end-to-end manner. We denote the initial node feature vector of a vertex ν_i by $\mathbf{x}_i^{(0)}$, where $\mathbf{x}_i^{(0)}$ is the i-th row in $\mathbf{X}^{(0)}$.
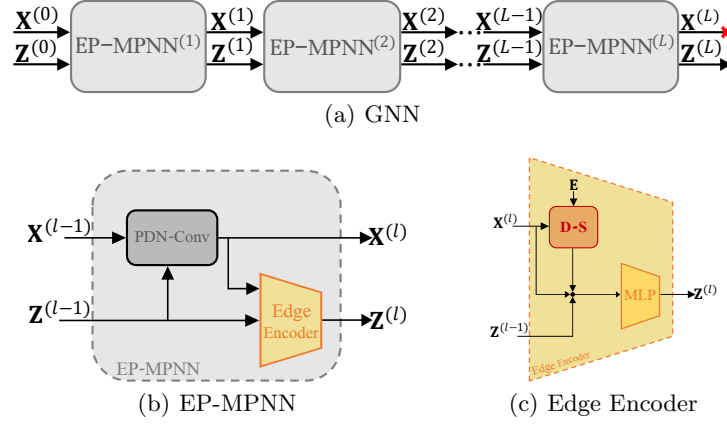
(a) GNN

(b) EP-MPNN

(c) Edge Encoder

Fig. 2. (a) A Graph Neural Network (GNN). The GNN is composed of L EP-MPNN blocks where L determines the message passing extent. The l-th EP-MPNN block updates the nodes and edge features, i.e., $\mathbf{X}^{(l-1)} \to \mathbf{X}^{(l)}$ and $\mathbf{Z}^{(l-1)} \to \mathbf{Z}^{(l)}$. **(b) Edge-oriented Pathfinder - Message Passing Network (EP-MPNN) Layer.** The basic layer in the graph neural network step comprises a PDN-Conv and an edge encoder. The PDN-Conv updates the node feature vectors based on their current values and an edge attention model. **(c) Edge encoder.** Updates the edge feature vectors. Its **D-S** block calculates the distance and similarity between the feature vectors of each pair of nodes by an edge. The output of the **D-S** block along with the nodes and the current edge features compose the input of an MLP which outputs the new feature vectors of the edges. The concatenation (denoted by '$\otimes$') of the **D-S** block's output along with the current node and edge features formulated in Eq. 4, is the input for an MLP which is trained to learn a new edges representation.

2.4 Graph Neural Network

The core of the proposed cell tracking framework is the graph neural network (GNN) illustrated in Fig. 2(a). Exploiting the GNN model and the message passing paradigm allows us to simultaneously trace entire cells tracks rather than locally associate cell instances in a frame-by-frame manner. One of our main contributions is the graph message passing block presented in Fig. 2(b) called the Edge-oriented Pathfinder Message Passing Neural Network (EP-MPNN). Specifically, we extend the MPNN block presented in [33] by introducing an edge encoder, thus enabling an interplay between the edge and node feature update simultaneously with an edge-attention mechanism.

The GNN is composed of L EP-MPNN blocks where L determines the message passing extent. In other words, the associations of cell instances in consecutive frames are affected by the respective connections along the sequence up to L frames away. The input to the $l+1$-th EP-MPNN block (which is, in fact, the output of the l-th EP-MPNN block) is composed of the updated node and edge features, denoted by $\mathbf{X}^{(l)} = \{x_i^{(l)}\}_{\nu_i \in \mathcal{V}}$ and $\mathbf{Z}^{(l)} = \{z_{ij}^{(l)}\}_{e_{ij} \in \mathcal{E}}$, respectively, where $l = 0, \ldots, L$. The nodes are updated using the pathfinder discovery network con-

volution (PDN-Conv) [33] which is one of the two EP-MPNN components. The other component is the *edge encoder* (illustrated in Fig. 2(c)) which is trained to embed the edge features based on the node features. We introduce the edge encoder in the following.

Node Feature Update.   The features of each node ν_i are updated based on the weights of the incoming edges. These weights are learned using an attention mechanism. Let $f_{\text{edge}}^{\text{PDN}} \colon \mathbb{R}^{d_{\mathcal{E}}} \to \mathbb{R}$ define a function implemented by an MLP that is trained to output scalars which represent the weights of the edges, given their current features. Let $f_{\text{node}}^{\text{PDN}} \colon \mathbb{R}^{d_{\mathcal{V}}} \to \mathbb{R}^{d_{\mathcal{V}}}$ define a vector function implemented by an MLP that is trained to output updated feature vectors given the current ones. The updated feature vector of a node ν_i is obtained by a weighted sum of its own and its neighboring nodes, as follows:

$$\mathbf{x}_i^{(l)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \overbrace{f_{\text{edge},l}^{\text{PDN}}(\mathbf{z}_{j,i}^{(l)})}^{\omega_{ji}^l} \overbrace{f_{\text{node},l}^{\text{PDN}}(\mathbf{x}_j^{(l-1)})}^{\tilde{\mathbf{x}}_j^{(l-1)}}, \tag{2}$$

where $\mathcal{N}(i)$ denotes the neighbors of ν_i; i.e., all the nodes ν_j for which there exist $e_{j,i} \in \mathcal{E}$. Note that $f_{\text{edge},l}^{\text{PDN}}$ and $f_{\text{node},l}^{\text{PDN}}$ are trained separately for each block. Eq. 2 can be interpreted as attention through edges, where, $\omega_{ji}^l = f_{edge,l}^{\text{PDN}}(\mathbf{z}_{j,i}^{(l)})$ is the predicted attention parameter of an edge $e_{j,i}$ ($\omega_{ii}^l = 1$) and $\tilde{\mathbf{x}}_j^{(l-1)} = f_{\text{node},l}^{\text{PDN}}(\mathbf{x}_j^{(l-1)})$ is the *mapped* feature vector of a node ν_j in the l-th EP-MPNN.

Edge Feature Update.   The main contribution of the proposed GNN framework is a mechanism for edge feature update that enhances the message passing process. Unlike the GNN presented in [33] here, the edge and node features are alternately updated. We denote by $\boldsymbol{D\text{-}S}$ a function that returns the distance & similarity vector of two feature vectors of connected nodes as follows:

$$\boldsymbol{D\text{-}S}(\mathbf{v}_i, \mathbf{v}_j) = \left[|v_i^1 - v_j^1|, \ldots, |v_i^{d_\nu} - v_j^{d_\nu}|, \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}\right] \tag{3}$$

which is a concatenation of the absolute values of the differences between corresponding elements in $\mathbf{v}_i$ and $\mathbf{v}_j$ and their cosine similarity $\frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}$. In the l-th block $\mathbf{v}_i = \mathbf{x}_i^{(l)}$ and $\mathbf{v}_j = \mathbf{x}_j^{(l)}$. In the initial phase, when the input to the first GNN block is formed, $\mathbf{v}_i$ and $\mathbf{v}_j$ are the i-th and the j-th rows in $\mathbf{V}_{\mathcal{V}}$, respectively. The construction of the initial feature matrix $\mathbf{V}_{\mathcal{V}}$ is described in Section 2.3.

The edge update function $f_{\text{edge},l}^{\text{EE}}$ (implemented as an MLP) returns the updated features of each edge $e_{i,j}$ given the concatenation of the current edge features, the updated feature vectors of the nodes it connects, and the output of the $\boldsymbol{D\text{-}S}$ block applied to these nodes. Formally,

$$\mathbf{z}_{ij}^{(l)} = f_{\text{edge},l}^{\text{EE}}([\mathbf{z}_{ij}^{(l-1)}, \mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}, \boldsymbol{D\text{-}S}(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})]) \tag{4}$$

The simultaneous edge and node feature update facilitates the message passing mechanism. Let ν_i be a node in frame t and let $\mathcal{N}(i)$ define the set of nodes

connected to it in frame $t - 1$. Consider a node $\nu_j \in \mathcal{N}(i)$. In the first GNN block ($l = 1$) the features of an edge $e_{j,i}$ are updated based on the features of ν_i and ν_j. In the same block ν_j is updated based on the features of its neighbors $\mathcal{N}(j)$ that are nodes representing cell instances in frame $t - 2$. The update is determined by the weights of the edges connecting them. In the next GNN block ($l = 2$) the features of an edge $e_{j,i}$ are again updated based on the features of ν_i and ν_j. However, this time the features of a node ν_j are already influenced by its neighbors. Therefore, when the features of ν_i are updated they are already affected by the neighbors of its neighbors which are cell instances that are two frames away. In the same manner, in the L-th block, cell instances and their first-order connections are influenced by cell-to-cell associations and higher-order connections along L frames.

2.5   Classifier and Training

The output edge feature vectors are the inputs to the edge classifier network. The classifier is an MLP with three linear layers each is followed by a ReLU activation, when a Sigmoid function is applied to the output layer. The output is a vector $\hat{\mathbf{Y}} \in \mathbb{R}_{[0,1]}^{|\mathcal{E}| \times 1}$ that represents the probability for each edge to be active (= 1) or not (= 0). We use the ground-truth (GT) edge activation vector $\mathbf{Y}$ to train the model. Since most of the edges in the graph dataset are not active (i.e., do not link nodes), our data are highly imbalanced. Therefore, we use a weighted cross-entropy loss function with adaptive weights which are determined by the average number of neighbors in a batch, i.e., $\left(\frac{1}{|\mathcal{N}|}, \frac{|\mathcal{N}|-1}{|\mathcal{N}|} \right)$. Note that since the size of the neighborhood region remains fixed throughout the sequence, the number of neighbors increases as the frames become denser.

2.6   Cell Tracking Inference

The output of the proposed deep learning framework is a probability matrix which identifies active edges. It is used together with the connection matrix $\mathbf{E}$ to construct candidates for cell trajectories as described in Section 2.2. Specifically, predictions of cell tracks are obtained at the inference phase in the form of a directed graph with soft edge weights (the output of a sigmoid). The edges in the graph represent only outgoing, potential associations between consecutive frames. The soft weights (association probabilities) allow us to partition the graph edges into active and non-active. Considering the outgoing/incoming edges of a specific node, there could be one of the following outcomes: 1) All outgoing/incoming edges are non-active - which may indicate end/beginning of a track. 2) Only one outgoing edge is active. 3) Two or more outgoing edges are active which may indicate mitosis (cell division). 4) More than a single incoming edge is active - i.e., when different cell instances are associated to the same cell instance. Above 99% of incoming/outgoing edges conflicts are avoided thanks to the proposed training scheme. We note that the network is implicitly trained to prefer bijection cell associations thanks to the attention-based mechanism of the
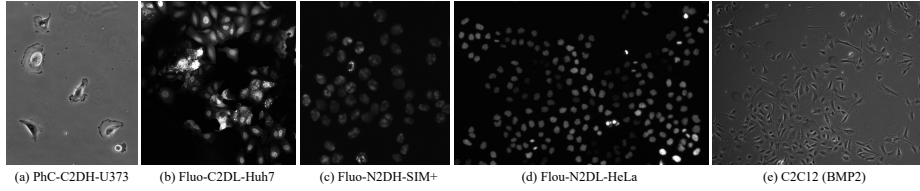
(a) PhC-C2DH-U373   (b) Fluo-C2DL-Huh7   (c) Fluo-N2DH-SIM+   (d) Flou-N2DL-HeLa   (e) C2C12 (BMP2)

Fig. 3. Example frames from five evaluated 2D datasets. Note the different appearance of the cells and the entire frames.

GNN blocks and the weighted loss function (see Section 2.5). Nevertheless, to ensure one-to-two mapping at most (case 3), in case that the association probabilities of more than two outgoing edges are higher than 0.5 (extremely rare events) only the top-2 are considered as active. In addition, to ensure injective mapping (case 4), only the incoming edge with the highest association probability (as long as it is higher than 0.5) is considered active. This obviously feasible ad-hoc strategy ensures a single path to each cell.

2.7   Mitosis Detection

Since daughter cells usually have different visual features than their parent, it is not very frequent for a node to have two active outgoing edges. In most mitotic events the parent track terminates whereas two new tracks initiate. To associate pairs of daughter cells to their parents we consider the detected tracks of all cells. We then look for triplets of trajectories $(\mathcal{T}_k, \mathcal{T}_l, \mathcal{T}_m)$ where $k, l, m \in \{1, \ldots, N\}$ such that $t_{\text{init}}^k = t_{\text{init}}^l = t_{\text{fin}}^m + 1$. If the spatial coordinates of $c_{t_{\text{init}}}^k, c_{t_{\text{init}}}^l$ and $c_{t_{\text{fin}}}^m$ are within the same neighborhood region, then we set $P(k) = P(l) = m$.

3   Experiments

We evaluated our model on six datasets from two different sources: an open dataset [9] we call C2C12 and five (2D and 3D) datasets from the Cell Tracking Challenge (CTC) [23,39]. The C2C12 dataset is composed of 16 sequences acquired with four different cell growth factors. Representative frames are presented in Fig. 3 for each of the 2D data sequences. Implementation details and further experiments including qualitative results (figures and videos) and ablation studies are provided in Sections A and B (respectively) of the appendix.

3.1   C2C12 Myoblast Datasets

In the following experiments, we focus on C2C12 [9], which is considered a challenging cell tracking dataset (see trajectories in Fig. 1). It is composed of high-throughput time-lapse sequences of C2C12 mouse myoblast cells acquired by phase-contrast microscopy under four growth factor conditions, including with fibroblast growth factor 2 (FGF2), bone morphogenetic protein 2 (BMP2),

Table 1. C2C12 cell tracking results [9]. A comparison of our association accuracy (left) and target effectiveness (right) scores to the scores obtained by five other methods. The top five rows present results for the original frame rate (best appears in **bold**) and the bottom two rows refer to the lower (×5) frame rate setup (best is <u>underlined</u>).

Method	Association Accuracy					Target Effectiveness				
	control	FGF2	BMP2	FGF2+ BMP2	Avg.	control	FGF2	BMP2	FGF2+ BMP2	Avg.
AGC [4]	0.604	0.499	0.801	0.689	0.648	0.543	0.448	0.621	0.465	0.519
ST-GDA [5]	0.826	0.775	0.855	0.942	0.843	0.733	0.710	0.788	0.633	0.716
BFP [26]	0.955	0.926	0.982	0.976	0.960	0.869	0.806	0.970	0.911	0.881
MPM [12]	0.947	0.952	**0.991**	**0.987**	0.969	0.803	0.829	0.958	0.911	0.875
Ours	**0.981**	**0.973**	**0.991**	0.986	**0.983**	**0.894**	**0.843**	**0.976**	**0.914**	**0.907**
CDM(×5) [11]	0.883	0.894	0.971	0.951	0.927	0.832	0.813	0.958	0.895	0.875
Ours (×5)	<u>0.958</u>	<u>0.942</u>	<u>0.988</u>	<u>0.970</u>	<u>0.964</u>	<u>0.905</u>	<u>0.852</u>	<u>0.981</u>	<u>0.919</u>	<u>0.914</u>

FGF2 + BMP2, and control (no growth factor). For each condition, four image sequences were captured, each consists of 780 frames (12 frames-per-hour, 1392×1040 pixels). The GT annotation includes a list of coordinates of the cell instances' centers along with the corresponding indices of the depicted biological cells. There is a complete cell annotation for only one sequence, where three cells and their lineage trees were marked in the others.

Experimental Setting and Evaluated Metrics. To quantify the tracking performance for the C2C12 dataset, we used the association accuracy (AA) and the target effectiveness (TE) measures as suggested by [12,26]. All comparisons were made with respect to human-annotated target cells and their frame-to-frame associations, which are considered the GT. The AA measure is defined by the ratio between the number of true positive associations and the total number of GT associations. The TE measure [6] considers the number of cell instances correctly associated within a track with respect to the total number of GT cell instances associated within that track. Training-test split was as in [12] where the test sequences of the C2C12 were annotated by the nnU-Net network [16].

C2C12 Tracking Results. The AA and the TE scores are reported in Table 1 for the four different types of C2C12 growth factor conditions. The top part of the table compares the performances of our framework with respect to the reported scores of four other methods: AGC [4], ST-GDA [5], BFP [26], MPM [12], where the latter two were considered the best performing methods for this dataset [12,26]. The bottom part of the table presents tracking scores for subsampled C2C12 sequences, simulating ×5 lower frame rate. Our scores are compared with those obtained by the CDM method [11] which was reported to have the best performance for this setup. Our tracking framework is shown to outperform all other methods, for almost all cell growth factor conditions in both setups. The advantage of our method over the classical global cell tracking approaches which are based on combinatorial algorithms such as AGC [4] and ST-GDA [5] can be explained by the fact that the optimization process is performed via training through the loss functions. Moreover, the uniquely designed message passing scheme which considers long term cell associations is

Table 2. Cell Tracking Challenge (CTC) results. Our tracking scores (TRA measure [22]) for the test sequences of five different datasets as published by the CTC organizers. The table presents the top three scores (our scores appear in bold) and our ranks with respect to the number of competing methods. The difference between our score and the best one appears in parentheses.

Dataset	Our-Rank(diff)	1st	2nd	3rd
PhC-C2DH-U373	**1**/25	**0.985**	0.982	0.981
Fluo-N2DH-SIM+	**1**/35	**0.978**	0.975	0.973
Fluo-N3DH-SIM+	**1**/11	**0.974**	0.972	0.967
Fluo-C2DL-Huh7	2/6(0.026)	0.960	**0.934**	0.865
Fluo-N2DL-HeLa	4/34(0.004)	0.993	0.991	0.991

preferable over the local, greedy and therefore sub-optimal frame-by-frame association paradigms such as MPM [12], BFP [26], and CDM [11].

3.2   Cell Tracking Challenge (CTC)

We competed in the CTC for five datasets [23,39] to further assess our method. Fig. 3 presents representative examples for each of the datasets, demonstrating their variability. The CTC is an excellent platform for objective and unbiased comparisons of the performances of different cell tracking methods when applied to a variety of 2D and 3D microscopy sequences of different acquisition methods (e.g., fluorescence or phase-contrast microscopy) and different cell types. Each CTC dataset is split into training and test sequences where the annotations of the test sets are deliberately unavailable to the competitors. The scores and ranks for all competing methods are calculated and published by the challenge organizers[2]. To quantify the tracking results the CTC organizers use the TRA measure [22]. The TRA measure is based on a comparison of the nodes and edges of acyclic oriented graphs representing cells and their associations in both the GT and the evaluated method. It computes the weighted sum of the required operations to transform the predicted cell lineage tree into the GT cell lineage tree and captures all the required information for tracking assessment, including mitotic events. For the CTC experiments, we used a publicly available segmentation method [3] to predict segmentation maps for the test datasets.

Results.   Table 2 presents the TRA scores for five CTC datasets: PhC-C2DH-U373, Fluo-N2DH-SIM+, Fluo-N3DH-SIM+, Fluo-C2DL-Huh7 and Fluo-N2DL-HeLa. Our scores are compared to the scores of the three-best performing methods. Our ranks are presented with respect to the number of competing methods that took part in the CTC for each dataset. Our method was ranked **first** for three out of five datasets and achieved the second- and the fourth-best ranks for the two other datasets. Our tracking results appear under BGU-IL (5).

---

[2] The Cell Tracking Benchmark web page is available at `http://celltrackingchallenge.net/latest-ctb-results/`.

Table 3. Quantitative ablation study. ST and DML stand for spatio-temporal and deep-metric-learning feature extraction, respectively. EE stands for the proposed edge encoder (Fig. 2(c)). MPNN is the chosen message passing scheme. The symbols ✓ and ✗ indicate included or excluded, respectively. AA and TE stand for association accuracy and target effectiveness scores.

Components				Results	
ST	DML	EE	MPNN	AA	TE
✓	✓	✓	GAT	0.805	0.353
✓	✓	✗	GCN	0.775	0.271
✓	✓	✓	GCN	0.871	0.359
✓	✓	✗	PDN	0.205	0.109
✗	✓	✓	EP-MPNN	0.989	0.960
✓	✗	✓	EP-MPNN	0.987	0.978
✓	✓	✓	EP-MPNN	**0.994**	**0.989**

3.3   Ablation Studies

We conduct ablation studies using the C2C12 dataset [9] to demonstrate the contribution of each component of our framework to the overall cell tracking performance. The first three rows in Table 3 present a comparison of the proposed EP-MPNN to two other commonly used message passing schemes, namely, GCN [19] and GAT [40]. The PDN method is similar to ours but does not have the edge encoder in the GNN block. The comparisons to the PDN and to the GCN without edge encoder assessed its importance. To demonstrate the significance of both the DML and the ST features we conducted two more experiments that are presented in the fifth and the sixth rows in the table. The DML and ST features both contribute to our tracking results. Further ablation study experiments are presented in Section B.4 of the Appendix.

4   Conclusions

We introduced an end-to-end deep learning framework for the simultaneous detection of complete cell trajectories in high-throughput microscopy sequences. This was accomplished by representing cell instances and their potential associations by nodes and edges (respectively) in a direct, acyclic graph; using deep metric learning for the extraction of distinguishing node features; and by using GNN and a uniquely designed message passing scheme to apply long range interactions between nodes and edges. The GNN's edge feature outputs were exploited to detect active edges that form paths in the graph, where each such path represents a cell trajectory.

Ablation study experiments presented in Section 3.3 assess the contribution of each of the proposed method components to the tracking accuracy. State-of-the-art tracking results are shown for a variety of 2D and 3D publicly available microscopy sequences. The code is released for comparisons and future study.

References

1. Amat, F., Lemon, W., Mossing, D.P., McDole, K., Wan, Y., Branson, K., Myers, E.W., Keller, P.J.: Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nature methods **11**(9), 951–958 (2014)
2. Arbelle, A., Reyes, J., Chen, J.Y., Lahav, G., Riklin Raviv, T.: A probabilistic approach to joint cell tracking and segmentation in high-throughput microscopy videos. Medical Image Analysis **47**, 140–152 (2018)
3. Arbelle, A., Riklin Raviv, T.: Microscopy cell segmentation via convolutional lstm networks. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1008–1012. IEEE (2019)
4. 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. IEEE (2015)
5. Bise, R., Yin, Z., Kanade, T.: Reliable cell tracking by global data association. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1004–1010. IEEE (2011)
6. Blackman, S.S.: Multiple-target tracking with radar applications. Dedham (1986)
7. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6247–6257 (2020)
8. Chen, Y., Song, Y., Zhang, C., Zhang, F., O'Donnell, L., Chrzanowski, W., Cai, W.: CellTrack R-CNN: A novel end-to-end deep neural network for cell segmentation and tracking in microscopy images. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 779–782. IEEE (2021)
9. Eom, S., Sanami, S., Bise, R., Pascale, C., Yin, Z., Huh, S., Osuna-Highley, E., Junkers, S.N., Helfrich, C.J., Liang, P.Y., et al.: Phase contrast time-lapse microscopy datasets with automated and manual cell tracking annotations. Scientific Data **5**(1), 1–12 (2018)
10. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning (ICML). pp. 1263–1272. PMLR (2017)
11. Hayashida, J., Bise, R.: Cell tracking with deep learning for cell detection and motion estimation in low-frame-rate. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 397–405. Springer (2019)
12. Hayashida, J., Nishimura, K., Bise, R.: MPM: Joint representation of motion and position map for cell tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
14. He, T., Mao, H., Guo, J., Yi, Z.: Cell tracking using deep neural networks with multi-task learning. Image and Vision Computing **60**, 142–153 (2017)
15. Hernandez, D.E., Chen, S.W., Hunter, E.E., Steager, E.B., Kumar, V.: Cell tracking with deep learning and the Viterbi algorithm. In: International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS). pp. 1–6. IEEE (2018)
16. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-net: A self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)

17. Jug, F., Levinkov, E., Blasse, C., Myers, E.W., Andres, B.: Moral lineage tracing. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5926–5935 (2016)
18. Kausler, B.X., Schiegg, M., Andres, B., Lindner, M., Koethe, U., Leitte, H., Wittbrodt, J., Hufnagel, L., Hamprecht, F.A.: A discrete chain graph model for 3D+ t cell tracking with high misdetection robustness. In: European Conference on Computer Vision (ECCV). pp. 144–157. Springer (2012)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
20. Magnusson, K.E., Jaldén, J., Gilbert, P.M., Blau, H.M.: Global linking of cell tracks using the Viterbi algorithm. IEEE Transactions on Medical Imaging **34**(4), 911–929 (2014)
21. Maška, M., Daněk, O., Garasa, S., Rouzaut, A., Munoz-Barrutia, A., Ortiz-de Solorzano, C.: Segmentation and shape tracking of whole fluorescent cells based on the Chan–Vese model. IEEE Transactions on Medical Imaging **32**(6), 995–1006 (2013)
22. Matula, P., Maška, M., Sorokin, D.V., Matula, P., Ortiz-de Solórzano, C., Kozubek, M.: Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. PloS one **10**(12), e0144959 (2015)
23. Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D.M., Karas, P., Bolcková, T., Štreitová, M., Carthel, C., Coraluppi, S., Harder, N., Rohr, K., Magnusson, K.E.G., Jaldén, J., Blau, H.M., Dzyubachyk, O., Křížek, P., Hagen, G.M., Pastor-Escuredo, D., Jimenez-Carretero, D., Ledesma-Carbayo, M.J., Muñoz-Barrutia, A., Meijering, E., Kozubek, M., Ortiz-de Solorzano, C.: A benchmark for comparison of cell tracking algorithms. Bioinformatics **30**(11), 1609–1617 (2014)
24. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check. In: European Conference on Computer Vision (ECCV). pp. 681–699. Springer (2020)
25. Neumann, B., Walter, T., Hériché, J.K., Bulkescher, J., Erfle, H., Conrad, C., Rogers, P., Poser, I., Held, M., Liebel, U., et al.: Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature **464**(7289), 721–727 (2010)
26. Nishimura, K., Hayashida, J., Wang, C., Bise, R.: Weakly-supervised cell tracking via backward-and-forward propagation. In: European Conference on Computer Vision (ECCV). pp. 104–121. Springer (2020)
27. Padfield, D., Rittscher, J., Roysam, B.: Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. Medical Image Analysis **15**(4), 650–668 (2011)
28. Panteli, A., Gupta, D.K., Bruijn, N., Gavves, E.: Siamese tracking of cell behaviour patterns. In: Medical Imaging with Deep Learning. pp. 570–587. PMLR (2020)
29. Payer, C., Štern, D., Feiner, M., Bischof, H., Urschler, M.: Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks. Medical Image Analysis **57**, 106–119 (2019)
30. Rempfler, M., Lange, J., Jug, F., Blasse, C., Myers, E.W., Menze, B.H., Andres, B.: Efficient algorithms for moral lineage tracing. In: IEEE International Conference on Computer Vision (ICCV). pp. 4695–4704 (2017)
31. Rempfler, M., Stierle, V., Ditzel, K., Kumar, S., Paulitschke, P., Andres, B., Menze, B.H.: Tracing cell lineages in videos of lens-free microscopy. Medical image analysis **48**, 147–161 (2018)

32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241. Springer (2015)
33. Rozemberczki, B., Englert, P., Kapoor, A., Blais, M., Perozzi, B.: Pathfinder discovery networks for neural message passing. In: Proceedings of the Web Conference. pp. 2547–2558 (2021)
34. Scherr, T., Löffler, K., Böhland, M., Mikut, R.: Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. Plos One **15**(12), e0243219 (2020)
35. Schiegg, M., Hanslovsky, P., Haubold, C., Koethe, U., Hufnagel, L., Hamprecht, F.A.: Graphical model for joint segmentation and tracking of multiple dividing cells. Bioinformatics **31**(6), 948–956 (2015)
36. Schiegg, M., Hanslovsky, P., Kausler, B.X., Hufnagel, L., Hamprecht, F.A.: Conservation tracking. In: IEEE International Conference on Computer Vision (ICCV). pp. 2928–2935 (2013)
37. Sixta, T., Cao, J., Seebach, J., Schnittler, H., Flach, B.: Coupling cell detection and tracking by temporal feedback. Machine Vision and Applications **31**(4), 1–18 (2020)
38. Spilger, R., Imle, A., Lee, J.Y., Mueller, B., Fackler, O.T., Bartenschlager, R., Rohr, K.: A recurrent neural network for particle tracking in microscopy images using future information, track hypotheses, and multiple detections. IEEE Transactions on Image Processing **29**, 3681–3694 (2020)
39. Ulman, V., Maška, M., Magnusson, K.E., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., et al.: An objective comparison of cell-tracking algorithms. Nature methods **14**(12), 1141–1152 (2017)
40. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (ICLR) (2018)
41. Wang, J., Su, X., Zhao, L., Zhang, J.: Deep reinforcement learning for data association in cell tracking. Frontiers in Bioengineering and Biotechnology **8**, 298 (2020)
42. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5022–5030 (2019)
43. Weng, X., Wang, Y., Man, Y., Kitani, K.M.: Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6499–6508 (2020)