THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques Appliquées
Unité de recherche : Centre de recherche Inria de l'Université Grenoble Alpes

Traitement numérique par graphes pour l'analyse d'images microscopiques

Digital processing by graphs for the analysis of microscopic images

Présentée par :

Ronny TONATO ZAMBRANO

Direction de thèse :

Sophie ACHARD DIRECTRICE DE RECHERCHE, CNRS DELEGATION ALPES	Directrice de thèse
Lionel HERVE DIRECTEUR DE RECHERCHE, CEA CENTRE DE GRENOBLE	Co-directeur de thèse
Guillaume GODEFROY INGENIEUR DOCTEUR, CEA CENTRE DE GRENOBLE	Co-encadrant de thèse

Rapporteurs :

DAVID ROUSSEAU
PROFESSEUR DES UNIVERSITES, UNIVERSITE D'ANGERS
VINCENT GRIPON
DIRECTEUR DE RECHERCHE, IMT ATLANTIQUE BREST

Thèse soutenue publiquement le **4 novembre 2025**, devant le jury composé de :

VALERIE PERRIER, PROFESSEURE DES UNIVERSITES, GRENOBLE INP - UGA	Présidente
SOPHIE ACHARD, DIRECTRICE DE RECHERCHE, CNRS DELEGATION ALPES	Directrice de thèse
DAVID ROUSSEAU, PROFESSEUR DES UNIVERSITES, UNIVERSITE D'ANGERS	Rapporteur
VINCENT GRIPON, DIRECTEUR DE RECHERCHE, IMT ATLANTIQUE BREST	Rapporteur
PIERRE BORGNAT, DIRECTEUR DE RECHERCHE, CNRS DELEGATION RHONE AUVERGNE	Examinateur

Invités :

LIONEL HERVE
DIRECTEUR DE RECHERCHE, CEA CENTRE DE GRENOBLE
GUILLAUME GODEFROY
INGENIEUR DOCTEUR, CEA CENTRE DE GRENOBLE

Contents

List of Figures **5**

List of Tables **9**

Abstract **13**

Resumé **14**

0 Overview **15**

1 Introduction **20**
1.1 Microscopy-based image analysis 20
1.1.1 Ongoing techniques 21
1.1.1.1 Bright-field microscopy 21
1.1.1.2 Fluorescence microscopy 22
1.1.1.3 Phase-contrast microscopy 23
1.1.1.4 Lens-free microscopy 24
1.1.2 Image reconstruction 24
1.1.3 Image-based applications 24
1.2 Graphs for biology 25
1.2.1 Research problem 25
1.2.2 Biological context 27
1.2.3 Imaging context 28
1.2.4 Objectives 29
1.3 Graph-based methodology 29
1.3.1 Graph notions 29
1.3.1.1 Graph definition 30
1.3.1.2 Complete graph 31
1.3.1.3 Subgraph 31
1.3.1.4 Clique 32
1.3.1.5 Features on graph 32
1.3.1.6 Edges orientation 33
1.3.1.7 Edges weight 33
1.3.1.8 Paths 33
1.3.2 Graph measures, construction and representation 34
1.3.2.1 Node measures 34
1.3.2.2 Edge measures 38
1.3.2.3 Graph measures 38
1.3.2.4 Graph construction 43
1.3.2.5 Graph representation 46
1.3.3 Learning on graphs 48

1.3.3.1 Graph embedding . . . 48
1.3.3.2 Inductive and Transductive learning . . . 49
1.3.3.3 Prediction tasks . . . 49
1.3.4 GNN methods . . . 50
1.3.4.1 Graph convolution . . . 51
1.3.4.2 Message passing . . . 52
1.3.4.3 Graph attention . . . 53
1.3.5 GNN-based applications . . . 55
1.3.5.1 Dynamics and interactions in network topology . . . 55
1.3.5.2 Cell image analysis . . . 57

2 Characterization of cellular tissues 58
2.1 Research context . . . 62
2.2 Literature review . . . 64
2.3 Methodology . . . 65
2.3.1 Dataset . . . 67
2.3.2 Segmentation process . . . 72
2.3.3 Graph conversion . . . 73
2.3.4 Graph metrics . . . 74
2.4 Results and Discussion . . . 76
2.4.1 ECMstiff data . . . 76
2.4.2 ROCKi data . . . 87
2.4.3 Actomyosin data . . . 90

3 Graph-based clustering 96
3.1 Research context . . . 98
3.2 Literature review . . . 99
3.3 Methodology . . . 101
3.3.1 Prediction tasks . . . 103
3.3.2 Datasets . . . 103
3.3.2.1 Simulated data . . . 103
3.3.2.2 Experimental data . . . 103
3.3.3 Graph 3D . . . 108
3.3.3.1 Unsupervised classification . . . 109
3.3.3.2 Supervised classification . . . 110
3.4 Results and Discussion . . . 112
3.4.1 HeLa and fibroblast cells . . . 112
3.4.2 PerKO cells . . . 120
3.4.3 Simulations . . . 122
3.4.3.1 Two distinct populations . . . 122
3.4.3.2 Two mixed populations . . . 124
3.4.4 HDF-HUVECs . . . 128

4 Dynamic prediction of dry mass 132
4.1 Research context . . . 132
4.2 Literature review . . . 133
4.3 Methodology . . . 135
4.3.1 Datasets . . . 137
4.3.1.1 Simulated data . . . 137

4.3.1.2 Experimental data . . . 138
4.3.2 Prediction task . . . 138
4.3.3 Data analysis . . . 139
4.3.3.1 Spatial autocorrelation . . . 139
4.3.3.2 Correlation analysis . . . 140
4.3.4 Graph construction . . . 143
4.3.5 Models . . . 144
4.3.5.1 Loss function . . . 145
4.3.5.2 Performance metrics . . . 145
4.4 Results and Discussion . . . 146
4.4.0.1 Simulated data . . . 146
4.4.0.2 Experimental data . . . 153

5 Conclusions and Perspectives **159**

6 Annexes **162**
6.1 Segmentation process on cellular tissues . . . 162
6.1.1 Initial segmentation . . . 162
6.1.2 Training-validation data split . . . 164
6.1.3 Model architecture . . . 164
6.1.4 Model training . . . 164
6.2 Graph clustering on simulated data . . . 166
6.2.1 Analysis of two distinct simulations . . . 166
6.2.2 Analysis of two mixed cell types . . . 169
6.2.2.1 Results on GMM and SVM methods . . . 169
6.2.2.2 Results on GNN-based supervised classification . . . 171
6.3 Graph clustering on experimental data . . . 173
6.3.1 Histograms of node features for two distinct populations . . . 173
6.3.2 Results on GNN-based supervised classification . . . 180
6.4 Spatial autocorrelation analysis . . . 182
6.5 GNN architectures . . . 186
6.5.1 Supervised classification model . . . 186
6.5.1.1 GNN encoder . . . 186
6.5.1.2 MLP Classifier . . . 187
6.5.2 Dynamic prediction . . . 188
6.5.2.1 GNN-based model . . . 188

References **189**

List of Figures

1 Image Data Processing pipeline . . . 15
2 Lens-free microscope design . . . 16
3 Examples of image acquisitions . . . 17
4 Reconstructed images . . . 17
5 Image-based analysis . . . 18

1.1 Cell cultures used in this work . . . 21
1.2 Bright-field image of Diploid yeast cells . . . 22
1.3 Fluorescence image of endoplasmic reticulum (ER) staining in HeLa cells . . . 23
1.4 Phase-contrast image of pancreatic stem cells . . . 23
1.5 Lens-free imaging of HeLa cells . . . 24
1.6 Examples of graphs in some fields of science and industry . . . 26
1.7 Examples of biological networks . . . 27
1.8 Microscopic image-based graph representation for a HeLa culture . . . 29
1.9 Microscopic image-based graph representation for pancreatic epithelial cells . . . 30
1.10 Example of a graph G . . . 30
1.11 Example of complete graph . . . 31
1.12 Example of subgraph . . . 32
1.13 Clique examples on a graph . . . 32
1.14 Types of graphs based on edges orientation . . . 33
1.15 Examples of graphs with weighted edges . . . 33
1.16 Paths on graphs . . . 34
1.17 Shortest paths on weighted graphs . . . 34
1.18 Node degree on graphs . . . 35
1.19 Undirected graph G_0 . . . 36
1.20 Local clustering coefficient on a node . . . 37
1.21 Core number on a graph . . . 38
1.22 Closed and open triplets . . . 39
1.23 Examples of assortativity and disassortativity graphs . . . 40
1.24 Undirected graph H . . . 41
1.25 Subgraph induced by the neighbors of each node in H . . . 41
1.26 Example of bridges on a graph . . . 43
1.27 Delaunay triangulations . . . 44
1.28 Gabriel node pairs . . . 44
1.29 Example of Urquhart graph . . . 45
1.30 Example of k-NN graph ($k = 1$) . . . 45
1.31 Undirected graph G_1 . . . 46
1.32 Directed graph G_2 . . . 46

1.33 Graph embedding . . . 49
1.34 Some examples of graph prediction tasks . . . 50
1.35 Example dataflows in three mainly types of GNNs . . . 51
1.36 Graphical representation of the convolution mechanism . . . 51
1.37 Graphical representation of the message passing mechanism . . . 53
1.38 Graphical representation of the attention mechanism . . . 55

2.1 Bright-field image of Human Pancreatic Duct Epithelial Cells . . . 58
2.2 Illustration depicting extracellular matrix . . . 59
2.3 Comparison between ERnet and Otsu methods . . . 61
2.4 A diagram explaining cells spreading based on the stiffness of their substrate . . . 63
2.5 Representation of a cellular tissue network as a graph . . . 63
2.6 Two-step proposed workflow . . . 66
2.7 Behavior of cells mediated by inhibitors acting on the actomyosin . . . 67
2.8 Evolving cellular network representation for all conditions . . . 69
2.9 Evolving cellular network representation for conditions corresponding to ECMstiff series . . . 70
2.10 Evolving cellular network representation for conditions corresponding to Actomyosin 1st set . . . 71
2.11 Evolving cellular network representation for conditions corresponding to Actomyosin 2nd set . . . 72
2.12 Segmentation process . . . 73
2.13 Graph conversion process . . . 74
2.14 Graph metrics obtained from the initial segmentation of the ECMstiff data . . . 78
2.15 Graph metrics obtained from the automatic segmentation of the Lensless6X data . . . 79
2.16 Evolving cellular network for condition 3 at 5 frames of the time-lapse video with their corresponding initial and final segmentation . . . 81
2.17 Evolving cellular network for condition 3 at 5 frames of the time-lapse video with their corresponding initial and final output graph . . . 82
2.18 Quantitative analysis for networks on ECMstiff data-Part 1 . . . 84
2.19 Quantitative analysis for networks on ECMstiff data-Part 2 . . . 85
2.20 Quantitative analysis for networks on ECMstiff data-Part 3 . . . 86
2.21 Evolving cellular network for condition 2 . . . 87
2.22 Quantitative analysis for networks on ROCKi data . . . 89
2.23 Characterization of epithelial network self-assembly . . . 92
2.24 Analysis of actomyosin regulation of epithelial network . . . 94

3.1 Proposed workflow for graph-based clustering . . . 102
3.2 Frame representation of 4 simulations from the first simulated dataset 104
3.3 Frame representation of 4 simulations from the second simulated dataset with two cell types . . . 105
3.4 Lens-free image of HeLa cells . . . 106
3.5 Lens-free image of Fibroblast cells . . . 106
3.6 Lens-free image of PerKO cells . . . 107
3.7 Lens-free image of HUVECs and HDF . . . 108
3.8 An example of Graph 3D over 5 frames . . . 109

3.9 GNN-based workflow for supervised classification . . . 111
3.10 Clusters on HeLa cells over frame 25 . . . 115
3.11 Clusters on HeLa cells over frame 125 . . . 115
3.12 Clusters on HeLa cells over frame 150 . . . 116
3.13 Clusters on Fibroblast cells over frame 31 . . . 116
3.14 Clusters on Fibroblast cells over frame 125 . . . 117
3.15 Clusters on Fibroblast cells over frame 150 . . . 117
3.16 Radar plot for cell morpho-dynamic features per cluster on mixing HeLa and fibroblast cells . . . 118
3.17 Radar plot for Graph 3D-based node features per cluster on mixing HeLa and fibroblast cells . . . 119
3.18 Radar plot for cell morpho-dynamic features per cluster on PerKO cells 121
3.19 Radar plot for Graph 3D-based node features per cluster on PerKO cells 122
3.20 Histogram of spatial degree distribution between two cell populations on Gabriel graph . . . 125
3.21 Histogram of spatial degree distribution between two cell populations on Delaunay graph . . . 125
3.22 Histogram of spatial betweenness centrality distribution between two cell populations on Urquhart graph . . . 125
3.23 GNN-based spatial model GNN_{sp} . . . 129
3.24 GNN-based temporal model GNN_{temp} . . . 129
3.25 Region partition for images on experimental data . . . 130
3.26 GNN-based global model GNN_{sp_temp} . . . 131

4.1 Proposed workflow for dynamic prediction of dry mass . . . 136
4.2 Frame representation of 4 simulations from the simulated data . . . 137
4.3 Lens-free image of HeLa cells . . . 138
4.4 Moran's Index on a simulation . . . 141
4.5 *P*-value from Moran's Index on a simulation . . . 141
4.6 Moran's Index on HeLa data . . . 142
4.7 *P*-value from Moran's Index on HeLa data . . . 142
4.8 Correlation analysis on HeLa data . . . 143
4.9 Outcomes obtained for two cells on 1st simulation . . . 150
4.10 Comparison between ground truth and predicted dry mass values for Cell 1 from simulated data . . . 151
4.11 Comparison between ground truth and predicted dry mass values for Cell 2 from simulated data . . . 152
4.12 Outcomes obtained for two cells on HeLa dataset . . . 155
4.13 Comparison between ground truth and predicted dry mass values for Cell 1 . . . 156
4.14 Comparison between ground truth and predicted dry mass values for Cell 2 . . . 157

6.1 U-Net model architecture . . . 165
6.2 Loss and accuracy over epochs . . . 172
6.3 Histograms for spatial and temporal degree . . . 173
6.4 Histograms for spatial and temporal betweenness centrality . . . 174
6.5 Histograms for spatial and temporal closeness centrality . . . 175
6.6 Histograms for spatial and temporal mean neighbor degree . . . 176

6.7 Histograms for spatial and temporal mean shortest path 177
6.8 Histograms for spatial clustering coefficient 178
6.9 Histograms for spatial and temporal core number 179
6.10 Histograms for spatial and temporal harmonic centrality 180
6.11 Moran's Index for the 2nd simulation of simulated data 182
6.12 *P*-value from Moran's Index for the 2nd simulation of simulated data 182
6.13 Moran's Index for the 3rd simulation of simulated data 183
6.14 *P*-value from Moran's Index for the 3rd simulation of simulated data 183
6.15 Moran's Index for the 4th simulation of simulated data 184
6.16 *P*-value from Moran's Index for the 4th simulation of simulated data 184
6.17 Moran's Index for the 5th simulation of simulated data 185
6.18 *P*-value from Moran's Index for the 5th simulation of simulated data 185
6.19 GNN encoder 186
6.20 MLP Classifier 187
6.21 GNN-based model 188

List of Tables

1.1 Shortest path distances on graph G_0 . . . 36
1.2 Node centrality measures on graph G_0 . . . 36
1.3 Shortest path distances on graph H . . . 41
1.4 Reciprocal shortest path distances on graph H . . . 41
1.5 Shortest path distances on subgraph H_1 . . . 42
1.6 Reciprocal shortest path distances on subgraph H_1 . . . 42
1.7 Shortest path distances on subgraph H_2 . . . 42
1.8 Reciprocal shortest path distances on subgraph H_2 . . . 42

3.1 Summary of the analysis conducted . . . 97
3.2 Comparison between GMM and He2CL algorithm with different feature combinations . . . 114
3.3 Best mean accuracy score on GMM for all possible node features combinations . . . 123
3.4 Best mean accuracy score on SVM for all possible combinations of features . . . 124
3.5 Mean accuracy score on GMM for the simulated data containing two cell types . . . 126
3.6 Mean accuracy score on a supervised method (SVM) for the simulated data containing two cell types . . . 126
3.7 Comparison of model performance by node features . . . 128
3.8 Comparison of model performance per node features between GNN_{sp} and GNN_{temp} . . . 129
3.9 Model performance per node features for experimental data . . . 130

4.1 Comparison of models' performance per features and graphs from 1st simulation . . . 148
4.2 Comparison of models' performance per features and graphs from 2nd simulation . . . 148
4.3 Comparison of models' performance per features and graphs from 3rd simulation . . . 149
4.4 Comparison of models' performance per features and graphs from HeLa test data . . . 154

6.1 Mean accuracy score (10 times) on GMM for all possible combinations of two distinct simulations with regard to the graph structures . . . 167
6.2 Best mean accuracy score (10 times) on GMM for all possible combinations of two distinct simulations with regard to all possible node features combinations . . . 168

6.3 Best mean accuracy score (10 times) on a supervised method (SVM) for all possible combinations of two distinct simulations with regard to the possible combinations of features 169
6.4 Kolmogorov-Smirnov test for simulated data with regard to the graph structures 170
6.5 Mean accuracy score (10 times) on GMM for the simulated data containing two cell types over all possible node features combinations 171
6.6 Mean accuracy score (10 times) on a supervised method (SVM) for the simulated data, containing two cell types, regarding to all graph structures and the possible combinations of features 171
6.7 Model performance per GNN layers 172
6.8 Comparison of model performance per node features and GNN layers for experimental data 181
6.9 Comparison of model performance per node features for experimental data 181

Acknowledgements

First and foremost, I would like to express my gratitude to God for guiding my life and providing me with the strength to pursue my goals steadfastly.

I am deeply grateful to my parents, Dayse, Marco, and Sulay, and my brothers, Diego and Jordan, for their unwavering support and encouragement throughout my PhD studies. Their belief in me has been a constant source of motivation and inspiration.

I would like to extend my sincere thanks to my PhD supervisor, Sophia Achard, for her exceptional availability, understanding, and insightful, constructive advice. I am deeply grateful for her guidance and mentorship throughout this journey.

I would also like to thank my PhD co-supervisors at CEA, Guillaume Godefroy and Lionel Hervé, for their accessibility and wise counsel. Their expertise and insights have greatly enriched my PhD experience, and I feel fortunate to have had the opportunity to learn from them.

I am grateful to the L4IV team for their warm welcome and support throughout my PhD. Their collaborative spirit and availability have made my time at the lab both productive and enjoyable.

I would also like to thank everyone I had the opportunity to meet during these three years of my PhD. Whether through collaborations, conferences, or casual interactions, each person I encountered has contributed to my growth and learning in unique ways. I am thankful for the diverse perspectives and experiences that have enriched my PhD journey.

Finally, I am grateful to CEA for providing me with the opportunity to pursue my PhD at this renowned research institute, which has enabled me to gain valuable knowledge and skills. The resources, facilities, and stimulating intellectual environment at CEA have been instrumental in the success of my research.

Dedication

To my parents, for their unconditional love and support.

Abstract

Image-based analyses have currently been utilized to extract relevant information from cell culture microscopic videos, enabling the study of single-cells and cell-cell interactions. However, these analyses have some limitations that could reduce the quality of these biological studies. Graphs, a particular type of data structure, can be proposed as a means to improve these analyses.

The main work is summarized in utilizing graph-based methods for three applications: the characterization of cellular tissues, a graph-based cell clustering method and the dynamic prediction of cell dry mass. For the first application, a segmentation method was implemented to segment cellular tissues images based on ground truth data manually segmented. This model was able to generalize over unseen data. Cellular tissues naturally develop mesh-like network structures, making it pertinent to represent them as graphs. This approach allowed for the identification of graph metrics to distinguish tissues that grow under different conditions and to analyze their formation.

For the second application, 2D microscopic videos of cell cultures were used. A spatio-temporal network-based approach was developed, in which individual cells are represented as nodes and their positional proximity determines the links (or edges) between them. This proposed graph structure integrates both positional and chronological data to capture distinctive node-level attributes. Two learning-based prediction tasks are designed: an unsupervised technique leveraging these extracted attributes and a supervised model utilizing a Graph Neural Network (GNN)-based model. Results show that cell types can be predicted without the need to track individual cells.

For the last application, the research focused on analyzing cell dry mass growth because it has been observed that a cell's growth is influenced by its neighbors. State-of-the-art GNN models are proposed to study the cell dry mass growth based on the characteristics of neighboring cells using a dynamic prediction approach.

Keywords: single-cells, cell-cell interactions, microscopic videos, graphs, cellular tissues, spatio-temporal network, dry mass, Graph Neural Network.

Resumé

Les analyses basées sur les images ont actuellement été utilisée pour extraire des informations pertinentes à partir de vidéos microscopiques de cultures cellulaires, permettant l'étude des cellules individuelles et des interactions cellule-cellule. Cependant, ces analyses présentent certaines limitations qui pourraient réduire la qualité de ces études biologiques. Les graphes, un type particulier de structure de données, peuvent être proposé comme un moyen d'améliorer ces analyses.

Le principal travail est résumé dans l'utilisation de méthodes basées sur les graphes pour trois applications : la caractérisation des tissus cellulaires, une méthode de regroupement cellulaire basée sur les graphes et la prédiction dynamique de la masse sèche cellulaire. Pour la première application, une méthode de segmentation a été mise en œuvre pour segmenter les images de tissus cellulaires en fonction de données de vérité terrain segmentées manuellement. Ce modèle a pu généraliser à des données non vues. Les tissus cellulaires développent naturellement des structures de réseau en forme de maillage, ce qui rend pertinent de les représenter sous forme de graphes. Cette approche a permis d'identifier des métriques de graphe pour distinguer les tissus qui poussent dans différentes conditions et d'analyser leur formation.

Pour la deuxième application, des vidéos microscopiques 2D de cultures cellulaires ont été utilisées. Une approche basée sur un réseau spatio-temporel a été développée, dans laquelle les cellules individuelles sont représentées par des nœuds et leur proximité positionnelle détermine les liens (ou arêtes) entre eux. Cette structure de graphe proposée intègre à la fois les données positionnelles et chronologiques pour capturer des attributs distinctifs au niveau des nœuds. Deux tâches de prédiction basées sur l'apprentissage sont conçues : une technique non supervisée exploitant ces attributs extraits et un modèle supervisé utilisant un modèle basé sur un réseau de neurones en graphes (GNN). Les résultats montrent que les types de cellules peuvent être prédits sans avoir besoin de suivre les cellules individuelles.

Pour la dernière application, la recherche s'est concentrée sur l'analyse de la croissance de la masse sèche cellulaire car il a été observé que la croissance d'une cellule est influencée par ses voisines. Des modèles GNN de l'état de l'art sont proposés pour étudier la croissance de la masse sèche cellulaire en fonction des caractéristiques des cellules voisines en utilisant une approche de prédiction dynamique.

Mots-clés: cellules individuelles, interactions cellule-cellule, vidéos microscopiques, graphes, tissus cellulaires, réseau spatio-temporel, masse sèche, réseau de neurones en graphes.

Chapter 0

Overview

This document represents the main results obtained during the PhD at the Département des Technologies pour l'Innovation en Santé (DTIS)-Division of Innovative Technologies for Health from CEA. I worked at the Laboratoire Instrumentation optique et traitement de l'Information pour l'Imagerie In Vitro (L4IV)-Optical Instrumentation and Data-Processing for In-Vitro Imaging Research Laboratory. My PhD supervisor was Sophie Achard, a CNRS Senior Research Scientist and Scientific Director of Multidisciplinary Institute in Artificial intelligence (MIAI); and my PhD co-supervisors were Lionel Hervé, a Senior Research Scientist at L4IV; and Guillaume Godefroy, a Research Scientist at L4IV.

L4IV is a research team whose work focus on developing non-conventional optical systems for in-vitro health applications. These applications primarily include in-vitro diagnostic devices, tools for the pharmaceutical and biotechnology industries, and tools for environmental and agricultural monitoring. The L4IV expertise encompasses optics tools, data science skills, knowledge of physics and biology, and system architecture. My work was focused on the in-vitro diagnostic devices axis.

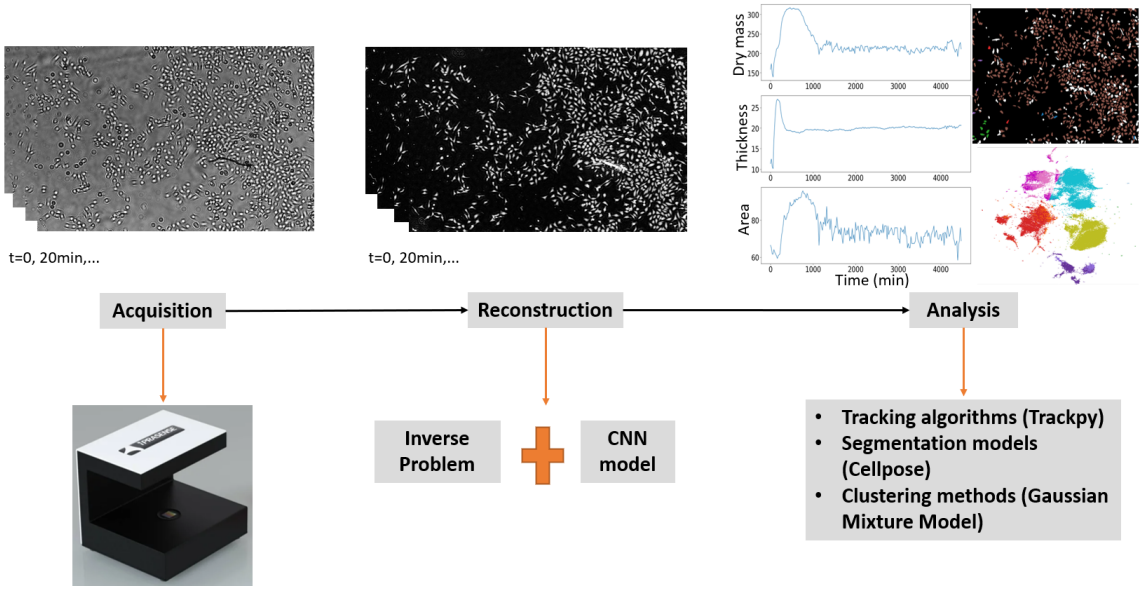


Figure 1: Image Data Processing pipeline. Samples are acquired from live cells using a lens-free microscope. The images of these samples are reconstructed using a combination of inverse problem solving and deep learning-based corrections. Then, relevant characteristics are extracted from images through deep learning methods to perform different analyses.

For particular projects undertaken by the L4IV team, a general workflow is applied (Figure 1), which begins with the acquisition of image of sample to analyze (cells, bacterias, organoids, tissues, spermatozoids, parasites) through a dedicated optical system. The image of the sample is retrieved via a reconstruction process that combines inverse problem solving and corrections by a deep learning convolutional neural network (CNN). This reconstruction approach is one of many available options, but it is typically used by the L4IV team for particular projects. Finally, reconstructed images are processed using specialized data processing techniques, such as tracking algorithms, segmentation models or clustering algorithms, to address specific needs. These needs may include the detection of heterogeneity in a cell culture, tracking of cell division, the study of cellular heredity, and other similar requirements.

Among all the optical tools developed, the L4IV team has specialized in the lens-free microscopy technique [11, 10, 13, 29, 119, 117]. The system is capable of visualizing thin, and transparent objects. It is also an ideal system for observing cell culture-type samples. Lens-free microscopy is indeed simple (Figure 2). It consists of a semi-coherent light source that illuminates an object (sample) located on an image sensor. This results in a system that is inexpensive, compact, and robust, enabling its placement in an incubator. This setup allows for observing samples over a wide field, time-lapse measurements and thus the tracking of a culture's evolution over time. The principle of the lens-free system is to measure the intensity of diffraction of the sample with a CMOS (Complementary Metal-Oxide-Semiconductor) detector at a distance of about $1\,mm$ [119].

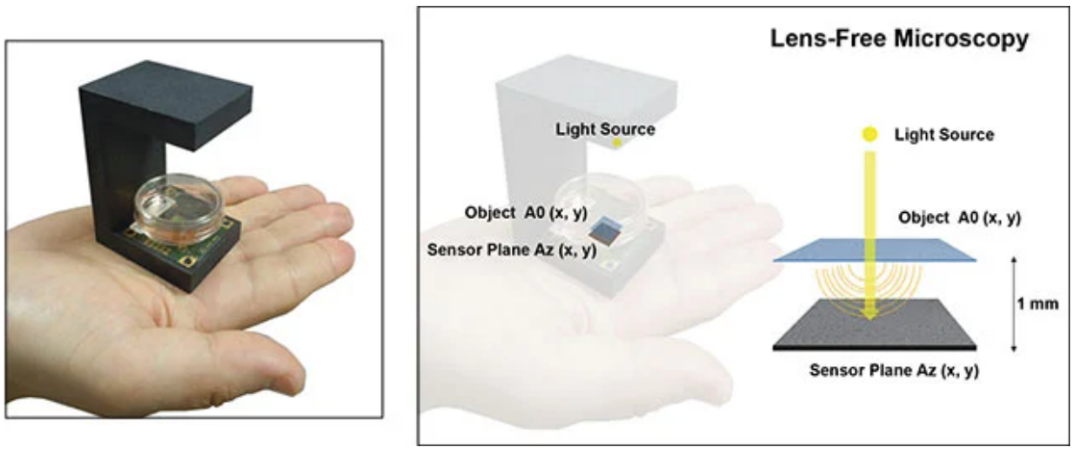


Figure 2: Lens-free microscope design. Source: https://www.photonics.com/Articles/Lens-Free_Microscopy_Focuses_on_Commercialization/a65007

Lens-free enables the acquisition of a rich variety of images based on intensity measurements in the sensor plane (Figure 3). The image of the sample is obtained through computation [118]. As previously described, the reconstruction of the sample's image is achieved through an iterative process that alternates between solving an inverse problem and applying an AI (artificial intelligence) step, which involves CNNs trained on simulated data.

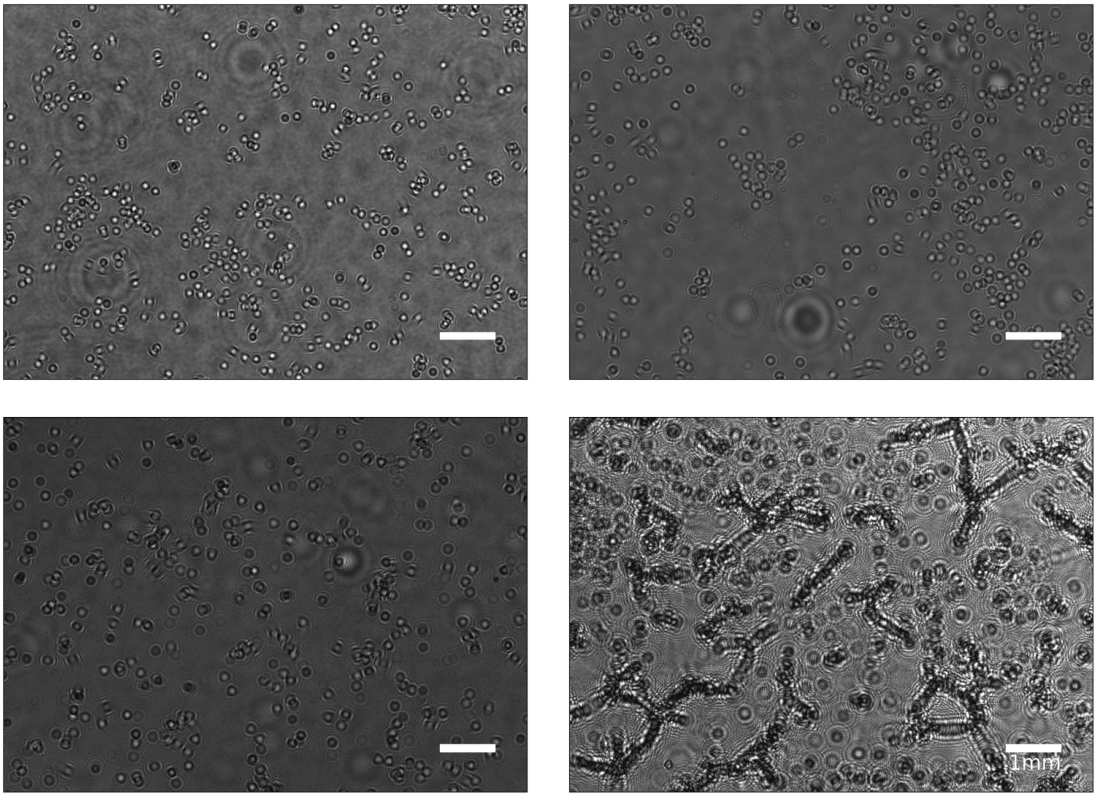


Figure 3: Examples of image acquisitions. The top left panel displays HeLa cells, while the top right panel shows fibroblast cells. The bottom left panel presents a mixture of HeLa and fibroblast cells, and the bottom right panel illustrates epithelial cells.

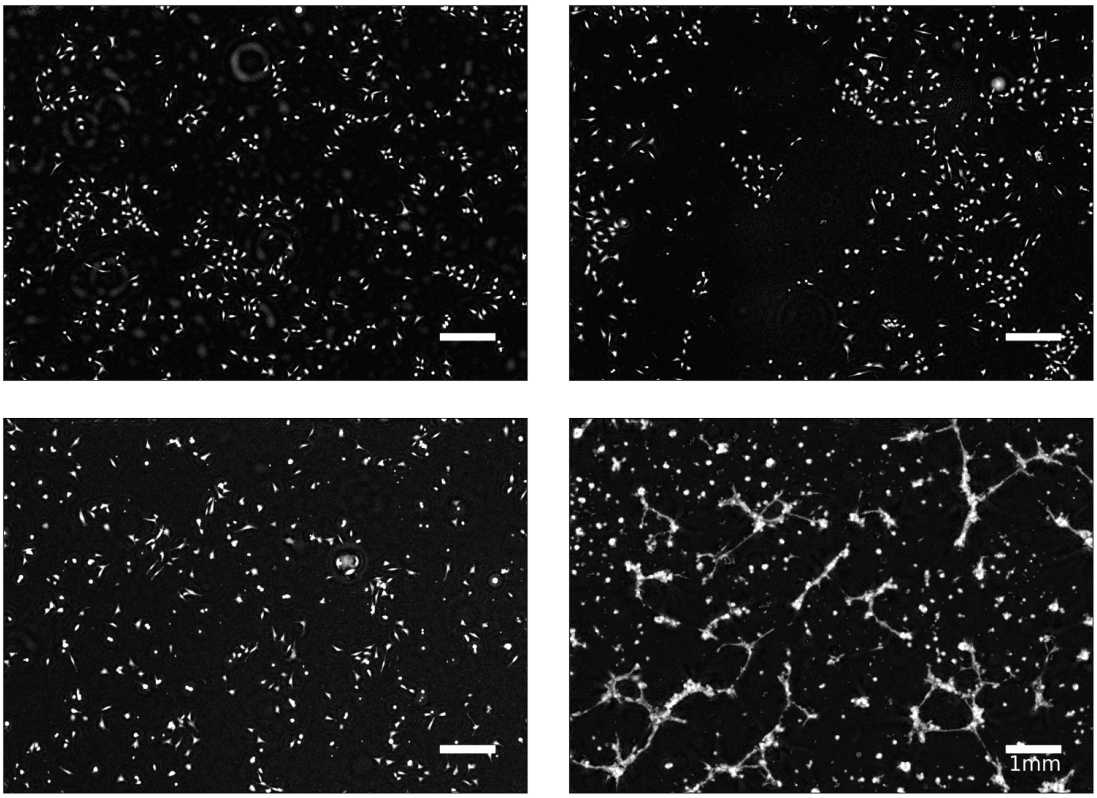


Figure 4: Reconstructed images from Figure 3

The inverse problem approach used to retrieve the image sample from intensity measurements is formulated as a regularized optimization problem [118]. The measurement is modeled as a function of the optical path difference (OPD), which is proportional to the dry mass. A cost function that integrates a data fidelity term and regularization terms is defined. This cost function is minimized through an optimization method that updates OPD.

The resolution of the reconstructed image is defined by the size of the measurement pixels (Figure 4).

The L4IV team also works on the analysis of videos of cellular cultures. These analyses involve mainly segmentation, tracking, feature extraction, and classification tasks (Figure 5). Among recent projects that tackle these analyses, a notable example is the project REVEAL[1], which focused on tracking and classifying (using a clustering approach) cells. However, there are still some challenges and limitations linked to all these analyses: difficulty in performing tracking (and therefore identifying cellular division) and the fact that cells are analyzed individually without considering their neighborhood, which could be potentially interesting.

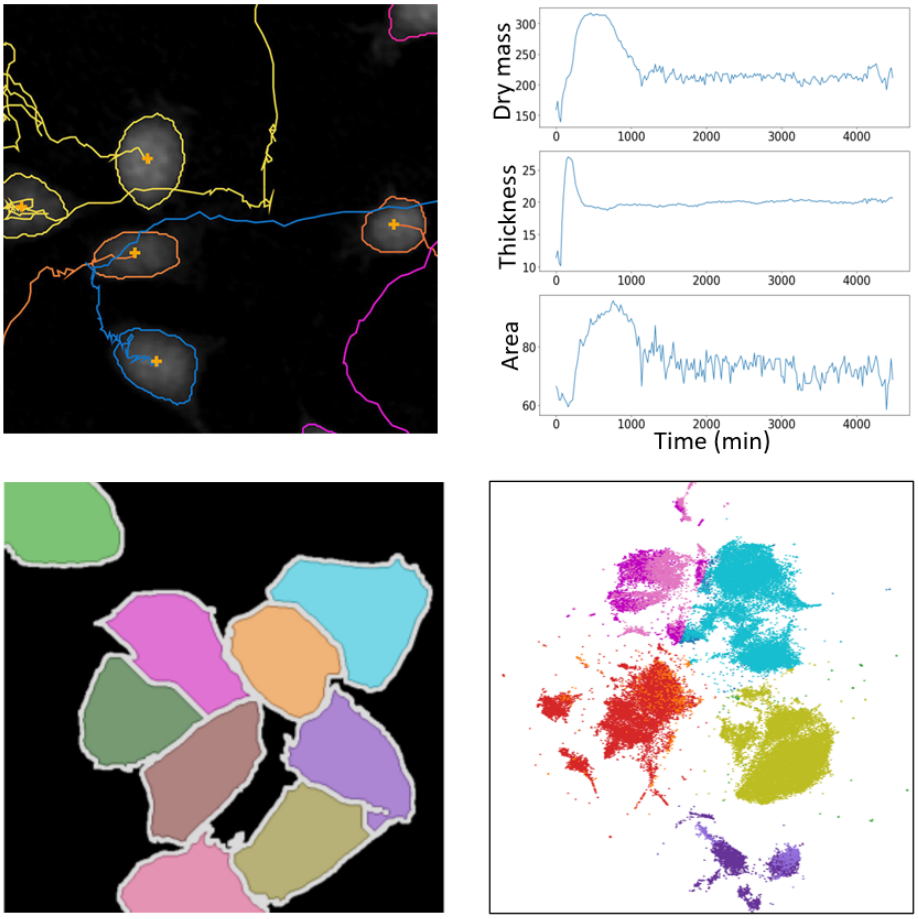


Figure 5: Image-based analysis. The top left panel displays the tracking of mouse cells [130], while the top right panel shows quantitative morphological features (dry mass, thickness, area) from mouse cells over time. The bottom left panel presents the segmentation of HeLa cells [225], and the bottom right panel illustrates cell clusters on a 2D plane representation.

My PhD research aims to explore the utilization of graphs on 2D cell microscopic

[1] https://reveal-h2020.eu/

videos to improve the current analyses because this allows to take into account cellular interactions, and potentially to identify cells without tracking them due to a new architecture that has been developed. My main contributions consist of the development of a graph-based spatio-temporal method (2D+time), learning on graphs (including this type of spatio-temporal structure), and the exploration and identification of graph metrics.

The thesis manuscript is structured as follows. The first chapter introduces the research problem and provides practical definitions and notions about graphs, which also includes some pertinent graph metrics at node level and globally utilized for the posterior analyses. The provided foundational graph information will enhance understanding of this data structure before introducing Graph Neural Network (GNN) models and discussing how learning occurs in graphs. The following three chapters present three different applications derived from my main research, each tackled with its respective context, methodology, results and discussion. GNN-based models are applied for two of these applications. Finally, conclusions are drawn and perspectives are presented, including recommendations and directions for future research in this area.

Chapter 1

Introduction

Cells are the primary membrane-bound unit that contains the fundamental molecules of life and of which all living things are composed [249]. A cell consists of three parts: the cell membrane, the nucleus, and, between the two, the cytoplasm. Within the cytoplasm lie intricate arrangements of fine fibers and hundreds or even thousands of miniscule but distinct structures called organelles [114]. Cells are divided into two main classes, initially defined by whether they contain a nucleus: prokaryotic and eukaryotic cells. Prokaryotic cells lack a nuclear envelope; eukaryotic cells have a nucleus in which the genetic material is separated from the cytoplasm [66].

Cells can acquire specific functions and carry out various tasks, such as replication, DNA repair, protein synthesis, migration and motility [7]. These tasks often involve interactions between cells. Cell-cell interactions come in a variety of forms, most of which fall into one of two types: direct physical contact and non-contact communication [128]. These interactions are key to the proper function of biological systems, as they are the source of the signals for precision changes in molecular processes [128].

This work focused on analyzing cell cultures of HeLa, fibroblasts, perKO, and pancreatic epithelial cells (Figure 1.1) that tend to form tissue networks by altering their morphogenesis. To analyze these types of cells, it is necessary to first visualize them using imaging tools. Microscopes make this possible. Furthermore, to account for possible interactions and behaviors within cells, it is appropriate to represent them using a suitable data structure that maintains this information and could facilitate its understanding. Graphs could be used for this purpose.

In the following sections, some traditional and recent microscopy-based techniques used to observe cells are presented. Interest in the use of graphs, the key data structure throughout this work, is then described from a biological perspective. This introduces the general research problem tackled in this PhD and the objectives to address it. Finally, practical information about graphs, from useful definitions to deep learning models based on this data structure, is provided.

1.1 Microscopy-based image analysis

The discovery of the cell was made possible through the invention of the microscope [253]. Robert Hooke first introduced the concept of cells as biological structures in his book Micrographia [123]. He used a simple microscope with two convex lenses to observe specimens under reflected light [253]. This marked an important

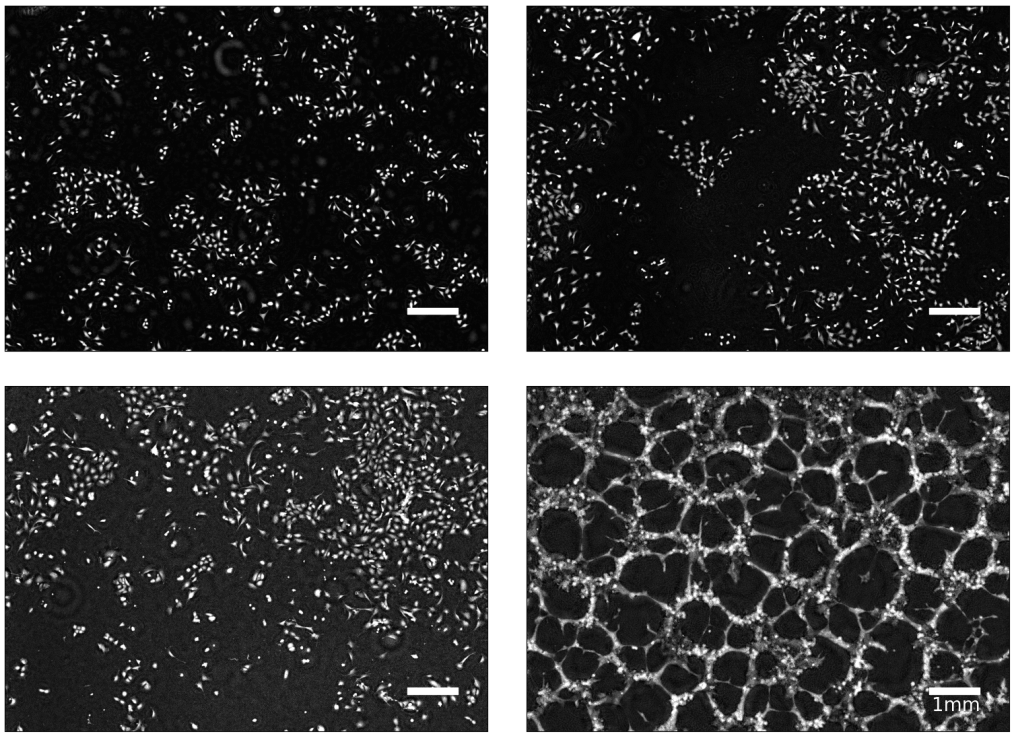


Figure 1.1: Cell cultures used in this work: HeLa cells (top left), fibroblast cells (top right), perKO cells (bottom left), and pancreatic epithelial cells-based tissue network (bottom right).

starting point for analyzing cells through microscopy and conducting subsequent studies derived from the observation of these objects. Microscopic techniques have been continuously developed to improve the visualization of biological objects (in particular, cells) to the greatest extent possible.

1.1.1 Ongoing techniques

Microscopy is a fundamental tool in cellular biology, enabling scientists to analyze single cells with remarkable precision. Traditional microscopy techniques (such as bright-field, fluorescence and phase-contrast) provide static images of cells, offering valuable insights into their structure and composition. However, live or video microscopy extends this capability by capturing dynamic processes in real-time, enabling the study of cellular movements, interactions, and responses over time. This capability is crucial for understanding the complex and often rapid changes that occur within and between cells.

1.1.1.1 Bright-field microscopy

Bright-field microscopy is one of the most basic and widely used techniques in optical microscopy. In this method, the object is illuminated from one side, and the resulting image is formed by the light that is transmitted through the sample. The contrast in bright-field images is generated by the absorption of light by the sample, which appears dark against a bright background (see Figure 1.2). This technique is simple and cost-effective, making it suitable for a wide range of applications, including the examination of stained tissue sections, cell cultures, and other biological specimens. However, it may not provide sufficient contrast for transparent or thin samples [131].

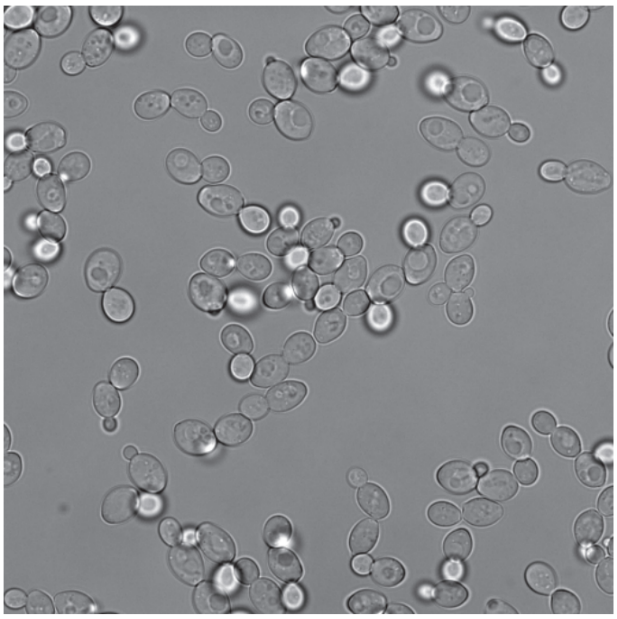


Figure 1.2: Bright-field image of Diploid yeast cells [296]

1.1.1.2 Fluorescence microscopy

Fluorescence microscopy is a powerful technique that allows for the visualization of specific components within a specimen by tagging them with fluorescent dyes or proteins [164]. In this method, the object is illuminated with light of a specific wavelength, which excites the fluorescent molecules, causing them to emit light at a longer wavelength. The emitted light is then detected to form an image [164] (see Figure 1.3). Fluorescence microscopy is extensively used in cell biology, neuroscience, and molecular biology. It offers high sensitivity and specificity, enabling the detection of low-abundance molecules and the study of dynamic processes within living cells. However, this technique is limited by several factors, including *photobleaching*, where a fluorophore permanently loses its ability to fluoresce upon illumination, *phototoxicity*, which involves the damaging effects of light on living cells, and interference from autofluorescence in biological specimens [136].

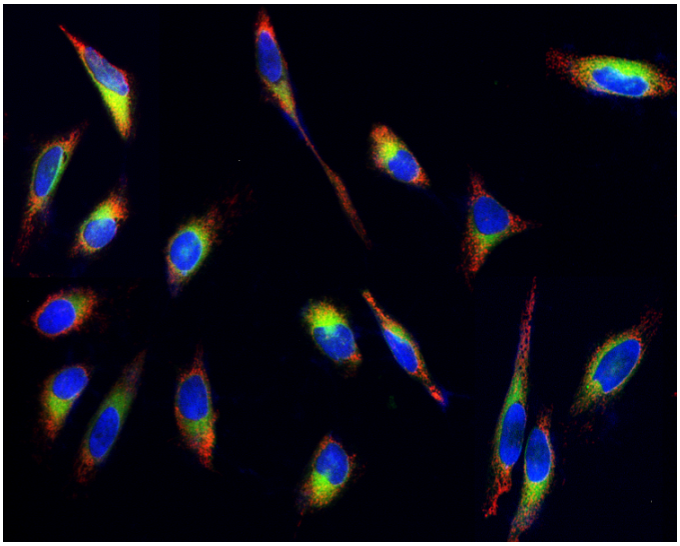


Figure 1.3: Fluorescence image of endoplasmic reticulum (ER) staining in HeLa cells cultured in a 96-well black wall/clear bottom plate. Source: https://www.aatbio.com/resources/assaywise/2016-5-1/fluorescence-live-cell-imaging

1.1.1.3 Phase-contrast microscopy

Phase-contrast microscopy is an imaging technique that converts phase shifts—differences between the phases of two periodic signals—in light passing through a transparent object into changes in brightness [295]. This method is particularly useful for observing living cells and tissues without the need for staining, as it enhances the contrast of transparent and semi-transparent specimens [295]. Phase-contrast microscopy works by exploiting the differences in the refractive index of the object, which causes light to be retarded or advanced in phase (see Figure 1.4). Specifically, the phase shift measured using this technique is directly proportional to the dry mass of the biological sample [199].

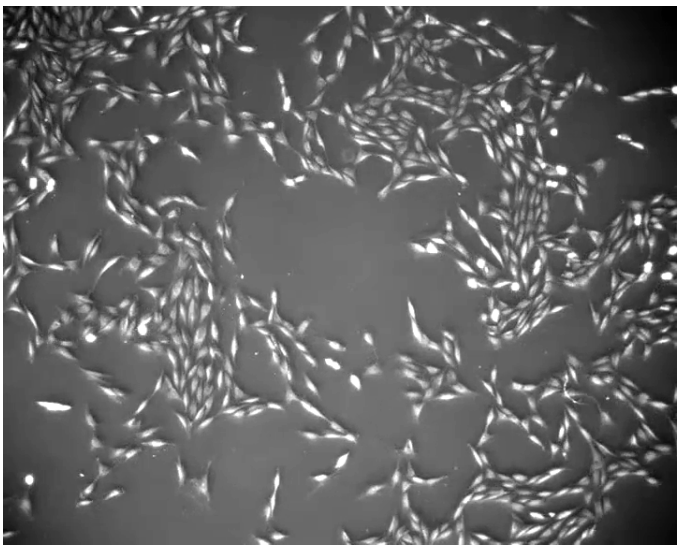


Figure 1.4: Phase-contrast image of pancreatic stem cells on a polystyrene substrate. Source: https://celltrackingchallenge.net/2d-datasets/

1.1.1.4 Lens-free microscopy

Lens-free microscopy [10, 13, 29, 222] is an imaging technique that uses digital sensors and computational algorithms to reconstruct images from the captured holograms (see Figure 1.5). The system is simple, compact, robust, and free of any optical components. It includes a semi-coherent light source that illuminates a sample object situated approximately $1\,mm$ away from an image sensor. This approach offers several benefits, including high resolution, a large field of view that allows for the simultaneous imaging of multiple cells or even entire tissue sections, the ability to perform real-time monitoring over extended time periods (time-lapse acquisitions), and label-free imaging, which enables the observation of biological processes without perturbing them.

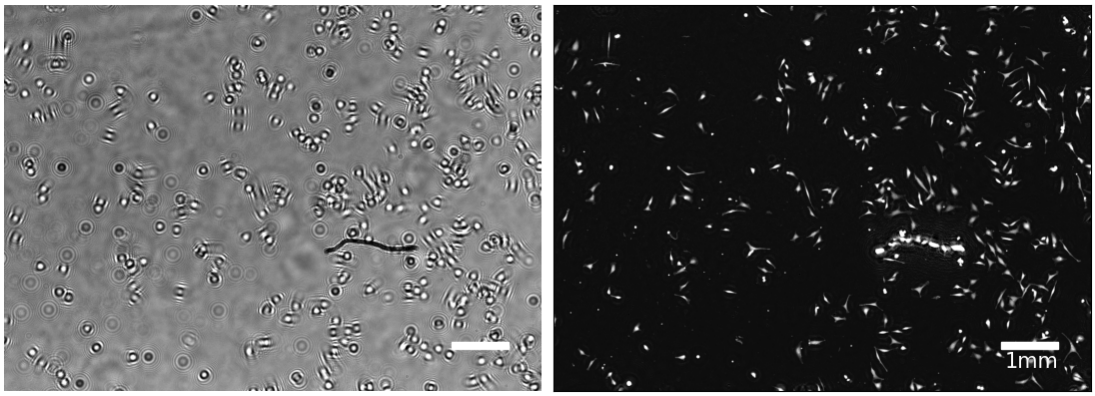


Figure 1.5: Lens-free imaging of HeLa cells. Left: Acquisition of live HeLa cells. Right: Reconstructed image obtained from the acquisition data through a process that combines inverse problem solving with corrections applied by a deep learning convolutional neural network.

1.1.2 Image reconstruction

Image acquisitions are performed from live cells through a lens-free microscope. The system provides an optical path difference image of each acquisition through a reconstruction process that combines inverse problem solving and corrections by a convolutional neural network [118]. The wide field of view imaged is 29.4 mm^2.

This reconstruction method was applied to the acquisition of all experimental data derived from cell cultures (HeLa, fibroblasts, perKO, HDF, HUVECs, and pancreatic epithelial cells) used in this work.

1.1.3 Image-based applications

Images and their analysis have a wide range of potential applications and tasks across various fields, driven by advancements in computer vision and image processing technologies. These applications include autonomous vehicles, virtual reality, the medical field, and others. Autonomous vehicles utilize image processing for critical tasks, such as lane detection, obstacle avoidance, and recognizing traffic signs to ensure safe navigation [179]. For augmented reality and virtual reality uses, images play a vital role in environment mapping and real-time object tracking, significantly enhancing the immersive experience [176]. In medical imaging, images are used for diagnostic purposes, tumor detection, and guiding surgical procedures with precision

[221]. Additionally, there are numerous methods to obtain images in medicine, including magnetic resonance imaging (MRI), radiography, ultrasound, computed tomography scans, and microscopy.

Microscopic imaging enables the observation of cells, the discovery of their structures, and the monitoring of biological phenomena, among other studies. To delve deeper into these analyses, automate them, and make them quantitative, this type of image can be processed using image data processing techniques such as segmentation, classification, and tracking. Traditional methods for image segmentation include thresholding algorithms [204, 280], watershed methods [30, 242], active contours [86], and deep neural networks [99, 109, 110, 174, 227, 251]. Image classification relies on various techniques, such as deep learning models, CNN, support vector machines, k-Nearest Neighbors, decision trees and random forests [33, 34, 148, 227, 234].

Different methods have been proposed for solving the problem of tracking single particles or cells [60]. These methods involve the development of algorithms and deep learning technologies [54, 111, 178, 262, 289], which are applied depending on the type of microscopic images acquired, such as bright-field, fluorescence, phase-contrast, and lens-free imaging. Some of these algorithms include LineageTracker [77], Pathfinder [56], SpotTracking [61], CellTracker [126], TrackMate [258], Laptrack [91], Trackpy [9]. In this work, tracking information for cells was obtained from the available microscopic images using the Trackpy and Laptrack algorithms.

However, some of these image data processing techniques still present several challenges and limitations. These include the difficulty in defining a unique segmentation method for the entire time-lapse acquisition and/or unseen data; the performance of cell tracking due to high cell density, overlapping cells, the quality of reconstructed images, irregular cell movement patterns, and cell division dynamics; and the high computational cost associated with processing high-definition images.

1.2 Graphs for biology

In this section, the interest of graphs, the key data structure throughout this work, is described from both biological and imaging perspectives. It also includes the introduction of the general research problem addressed in this PhD, as well as the objectives proposed to achieve it.

1.2.1 Research problem

Current biological studies of cells rely on image-based analysis, as described previously. Most of the time, these analyses remain centered on elementary cellular measurements, where cells are considered as isolated objects without considering their interactions with neighboring cells. If time-lapse videos from 2D cell microscopic images (taken at certain time intervals) are considered, it will increase the difficulty of their analysis and understanding, since not only spatial interactions are involved but also temporal interactions evolving over time.

Deep learning models have shown success in various tasks such as speech recognition [55, 65, 299], machine translation [73, 156, 203, 219, 267], image reconstruction [76, 106, 162, 300], object recognition [109, 125, 247, 254], segmentation [99, 109, 110, 174, 227, 251], etc. This success is due to their ability to exploit and use the local statistical properties that exist within data, which can be represented in a grid

form (such as images, known as Euclidean data) or an ordered sequence (such as musical note sequences or text sequences).

However, not all real-world data can be represented in the format required for classic deep learning models, that is, within a grid or as an ordered structure. Data that cannot be represented as grids or ordered structures is known as non-Euclidean data, and some examples include meshes, molecules, and 3D complex geometric shapes, etc. Furthermore, non-Euclidean data can be represented using graphs or manifolds, where it is important to note that graphs are ubiquitous. Examples of data that can be represented by graphs (see Figure 1.6) include social networks [282], academic paper citation networks [80], communication networks [155], recommendation systems [276], knowledge graphs [104], molecules [144], MRI [112], and road maps [292].

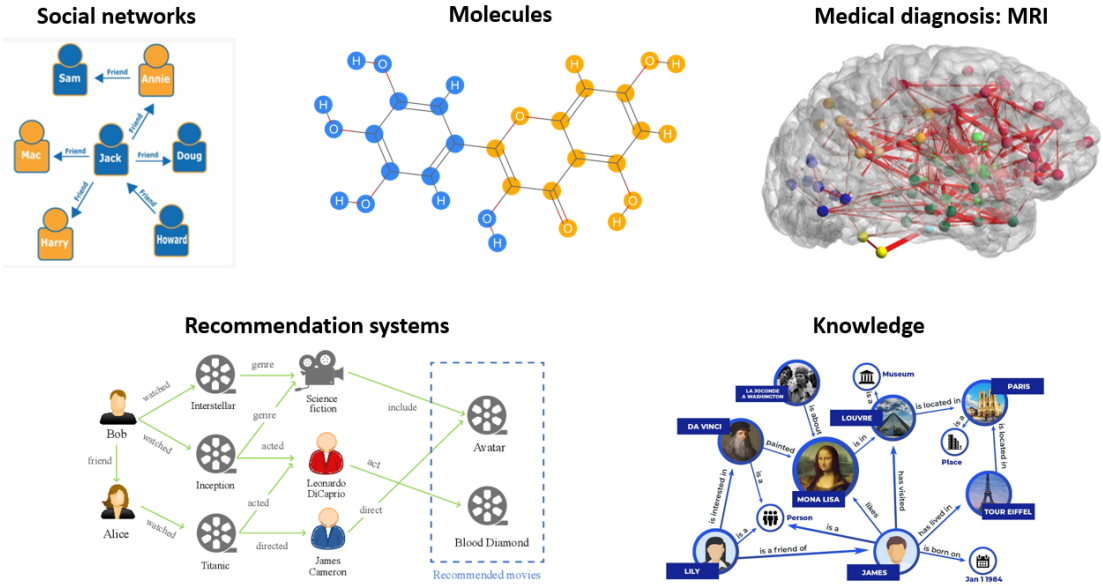


Figure 1.6: Examples of graphs in some fields of science and industry

Classic deep learning architectures are limited in their application to Euclidean data. This has motivated the study of deep learning methods for this type of data structure, which was introduced by Bronstein et al. in [46] through the concept of Geometric deep learning. This aims to study and develop new deep learning methods for data represented by graphs or manifolds.

In general, networks or graphs are used to capture relationships between entities or objects [147]. Besides, as graphs are also useful to represent non-Euclidean data, an instinctive idea could be to characterize 2D images through this representation to build a cell graph with its interactions between cells. However, this immediately raises several questions: How should a graph be constructed, and how should its interactions be defined? How should the graph be interpreted? What types of applications could be analyzed with graphs generated from available data? What types of models could be developed? Among these considerations, it is evident that Graph Neural Networks (GNNs) represent the current state-of-the-art in deep learning models, utilizing graphs as input data [101]. Consequently, the focus will be on globally understanding and analyzing 2D cell microscopic images through graphs and their potential applications/models derived from this type of structure.

1.2.2 Biological context

Networks are one of the most common ways to represent biological systems as complex sets of binary interactions or relations between different bioentities [147]. In general, graphs (a mathematical way of representing a network), are used to capture these relationships between entities or objects. In a typical representation, a graph consists of a set of vertices, nodes, or points (in this case, cells) connected by edges, lines, links, arrows, or arcs [147].

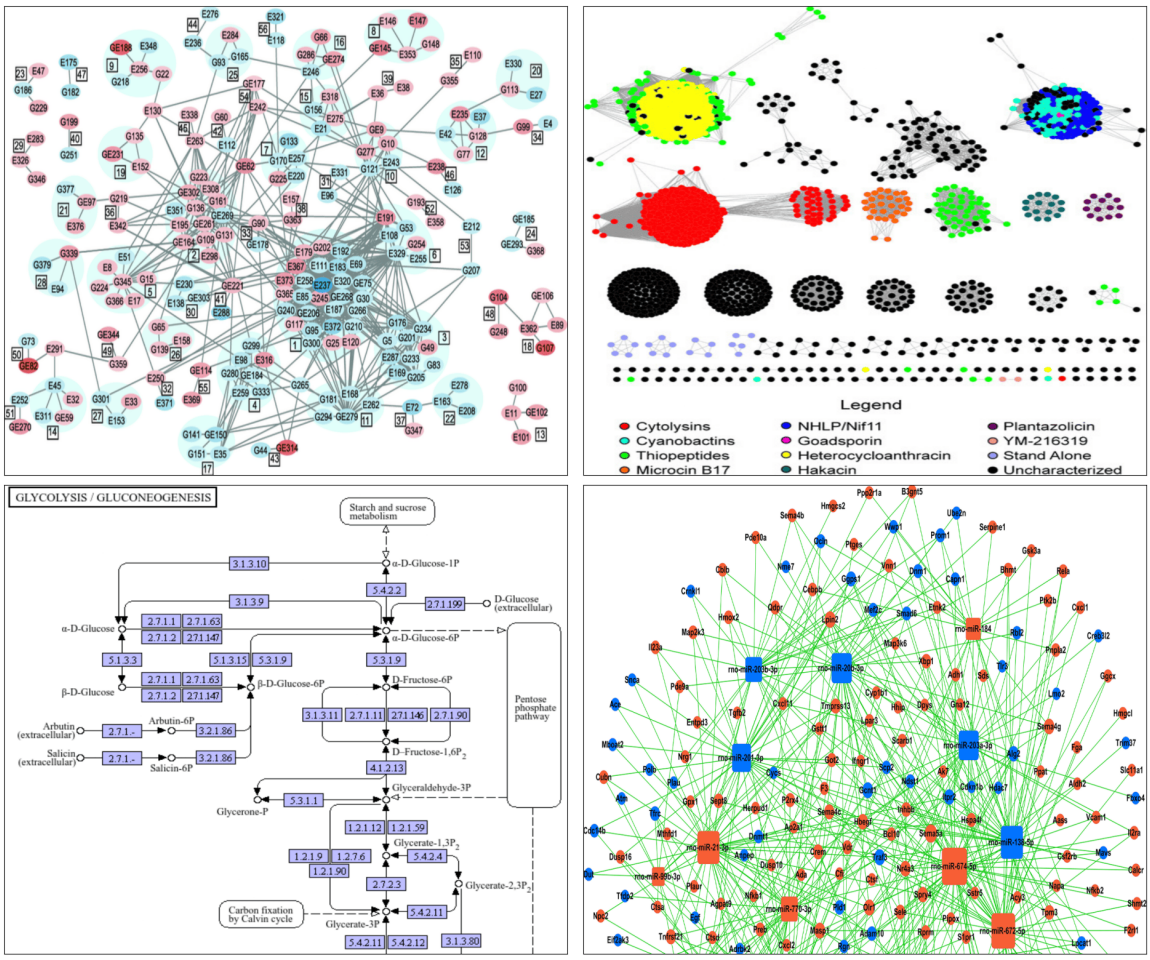


Figure 1.7: Examples of biological networks. Top left: A protein–protein interaction network shown in Cytoscape [256]. Top right: A sequence similarity network of TOMM (thiazole/oxazole-modified microcin) D-proteins [67]. Bottom left: A KEGG (Kyoto Encyclopedia of Genes and Genomes) metabolic pathway (Source: https://www.kegg.jp/pathway/ko00010). Bottom right: A miRNA-gene network where box nodes represent miRNAs (microRNAs) and circular nodes represent mRNAs (messenger RNAs) [182].

Biological networks model the functions of cell and tissue-specific molecular interactions at various organizational levels, from individual cells to entire organs [147]. Some of these biological representations (see Figure 1.7) include Protein-Protein Interaction Networks, which detail how proteins interact to facilitate biological processes within cells; Sequence Similarity Networks, which capture sequence similarities between proteins or genes; Gene Regulatory Networks, which map regulatory relationships between transcription factors and their binding sites or genes; Signal Transduction Networks, which illustrate the transmission of molecular signals within

or into cells [84]; Metabolic Networks, which consist of metabolites and their interactions; Gene Co-expression Networks, which connect genes based on significant co-expression data from technologies like Microarrays, RNA-Seq (RNA sequencing), or scRNA-seq (Single-cell RNA sequencing); and lncRNA–Protein Interaction Networks, which reveal the functions of long non-coding RNAs through their interactions with proteins [294].

In biomedical research, graphs are highly valuable for both researchers and clinicians [294] because they can capture the associations between any type of biological entity such as proteins, genes, small molecules, metabolites, ligands, diseases, or drugs [147]. Some examples of these graphs include disease networks, which link diseases sharing at least one causative gene; drug-disease associations, which provide information on known and/or predicted drug-disease connections; and disease-symptom graphs, which map diseases to their symptoms and illustrate potential disease progression. This visualization aids clinicians in administering more efficient medical treatments swiftly [248].

1.2.3 Imaging context

Graph-based techniques for representing and manipulating data have been extensively explored in the domains of image processing and image analysis [233]. Among all possible graph-based imaging applications, researchers have been focused on analyzing the following ones: segmentation, object recognition, and image restoration. These applications rely on classic algorithms, properties and tools derived from graph theory and spectral graph theory [48, 85, 208, 244, 255, 263, 270, 283].

The examples of graph utilization in both biological and imaging contexts (as seen previously) highlight the importance of this data structure for application in this research. Regarding the reconstructed 2D microscopic images available for this work, the question remains: How can these images be represented to extract relevant information for their subsequent analysis? This remains unclear, as the proper way to construct graphs from tissue images is still not well established [257]. The answer to the question lies in the construction of a graph from the image where nodes represent cells or cellular components, and edges represent their various types of interactions [51]. It is important to note that the interpretation and usage of nodes and edges can vary depending on the biological structures observed in the images. Figure 1.8 illustrates a HeLa culture for a particular instant. In this case, it is clearly possible to observe individual cells as distinct objects in the image. Then, the graph extracted from this image consists of nodes (red points) representing cells and edges (green lines) representing their spatial proximity. This graph representation allows for the extraction of certain characteristics (or features) related to the network topology for each cell, which can be used for the analysis of its spatial behavior. Additionally, it is possible to link each cell over time by constructing edges that represent its temporal proximity between successive frames in a time-lapse video.

Figure 1.9 depicts a tissue network formed from pancreatic epithelial cells. In this network, individual cells and their interactions can not be visually distinguished because the cells are mostly assembled together. The graph associated with this image consists of nodes (red points) representing cell assemblies and edges (green lines) corresponding to the cellular branches of the tissue network. Due to the nature of the biological object considered, it is not prudent to extract features for individual cell assemblies from this graph representation. Instead, it will be more

convenient to extract features from the entire graph (globally), which could enable the characterization of the evolution of this network over time.

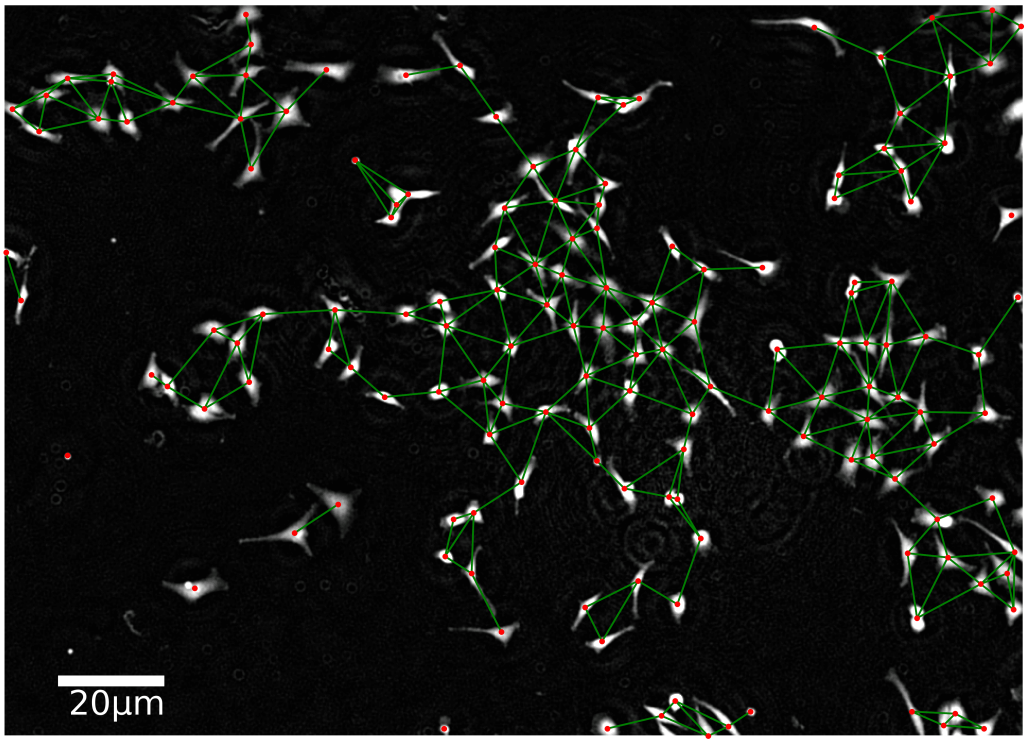


Figure 1.8: Microscopic image-based graph representation for a HeLa culture. Red points denote nodes in the graph and green lines correspond to their spatial proximity (edges).

1.2.4 Objectives

As previously introduced, the **general research objective** is:

- Analyze 2D live-cell microscopic videos through graphs.

Thus, the aim is to explore the utilization of graphs on these microscopic videos and their relevance to various applications. Based on the information provided above, the following **specific objectives** are proposed:

- Construct appropriate graph representations from the provided datasets in accordance with the analyzed task.
- Study the interpretability of generated graphs and evolving networks in biological terms over time.
- Develop graph-based deep learning models (GNNs) for the prediction tasks addressed, including cell classification and cell dry mass forecasting.

1.3 Graph-based methodology

1.3.1 Graph notions

In this section, key notions, definitions, and useful tools are introduced to facilitate familiarity with graphs and the models that utilize this type of data structure.

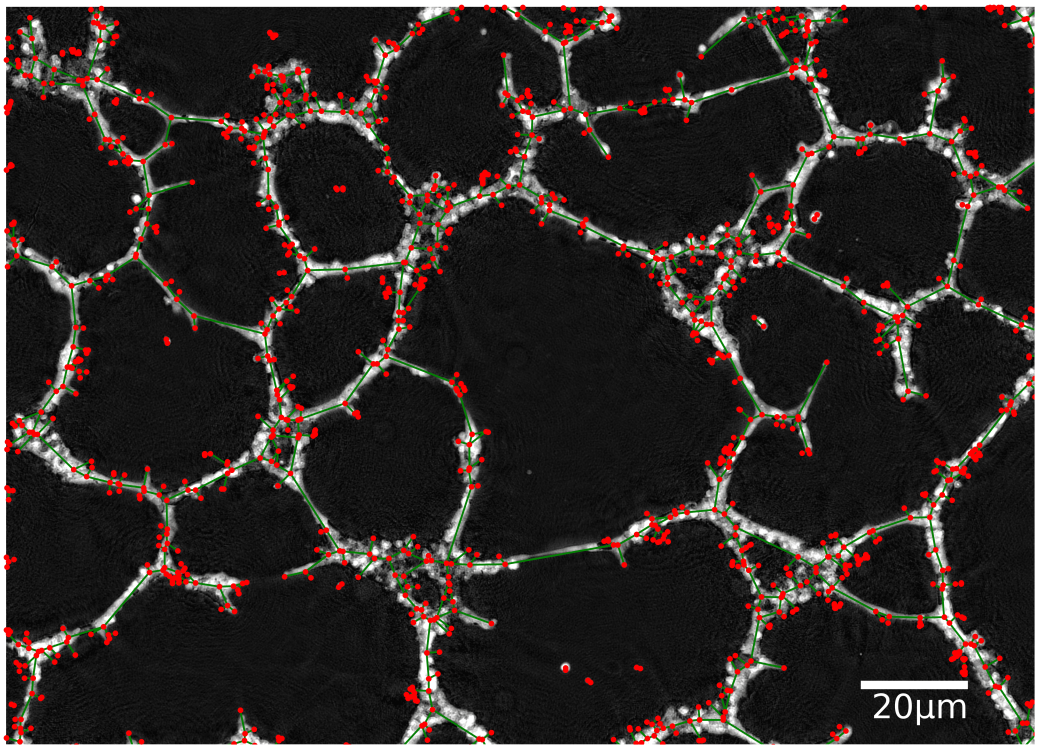


Figure 1.9: Microscopic image-based graph representation for pancreatic epithelial cells. Red points denote nodes (cell assemblies) in the graph and green lines correspond to their cellular branches (edges).

1.3.1.1 Graph definition

First, the definition of a graph will be introduced.

Definition 1.1 (Graph)**.** A **graph** G is a pair $G = (V, E)$ where:

- V is a set of nodes (vertices): $\{v_i\}_{i \in V}$
- E is a set of edges (paired vertices): $\{e_i\}_{i \in E}$.

In the following figure, an example of a graph G is shown:

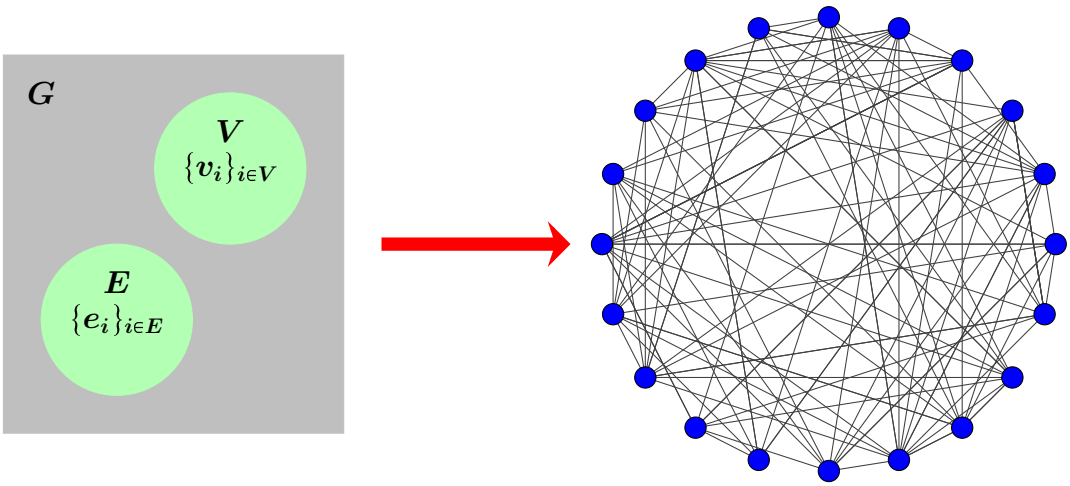


Figure 1.10: Example of a graph G

So, the information about these two sets is all that is needed to retrieve the graph structure.

As some examples of graphs were described previously (see Section 1.2.1), their respective nodes and edges representations can now be illustrated as follows:

- In social networks, a node is represented by a person, and his/her relationships with others represent the edges of the graph.
- For a molecule, its nodes are the atoms and the edges are determined by the bonds between atoms.
- In recommendation systems, a node is represented by a customer and his/her connections with others represent the edges of the graph.
- For knowledge graphs, a node is represented by a concept and the edges can be seen as statements.
- For MRI, nodes represent brain regions, and the interactions between brain regions represent the edges.

Graphs can have thousands or even millions of nodes and edges, making them complex. Social networks are examples of such complex graphs, typically representing data from millions of people, where an individual can have thousands of connections with others.

1.3.1.2 Complete graph

Let $G = (V, E)$ be a graph in which each node is connected to every other node. Then, G is known as a **complete graph**. Figure 1.11 shows a complete graph with 5 nodes where all nodes are connected to each other.

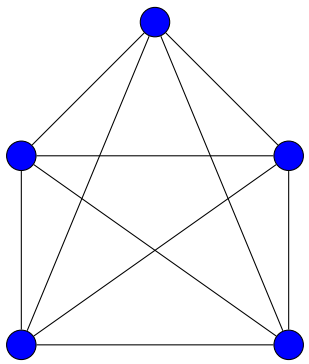
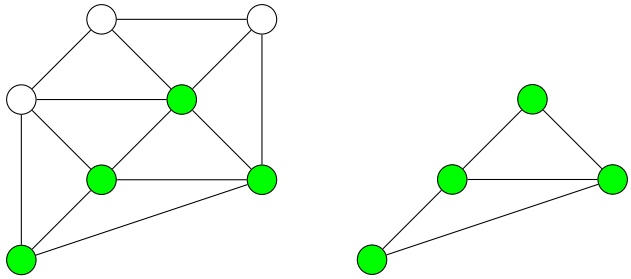


Figure 1.11: Example of complete graph

1.3.1.3 Subgraph

Let $G = (V, E)$ be a graph with V, E as its sets of nodes and edges, respectively. Then, a **subgraph** H of the graph G is simply another graph formed from a subset of the nodes and edges of G: $H = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$.

Figure 1.12 displays an example of subgraph construction. In this example, a graph G with 7 nodes is considered, as shown in Figure 1.12a. From G, a subgraph H is extracted, depicted in Figure 1.12b. This subgraph H consists of 4 nodes (highlighted in green) and their associated edges.

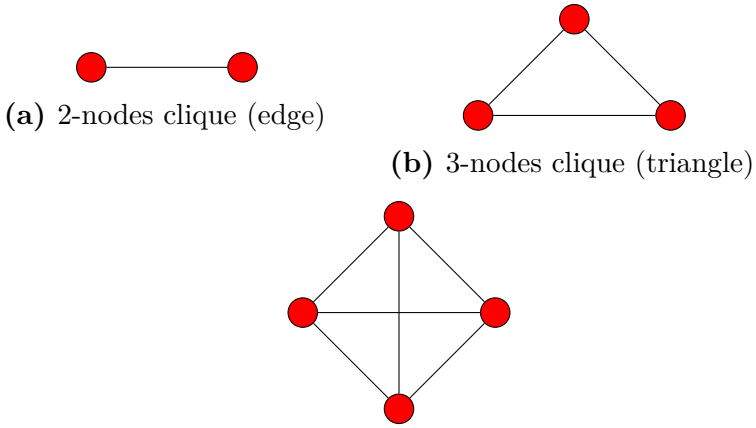


(a) Graph G **(b)** Subgraph H

Figure 1.12: Example of subgraph

1.3.1.4 Clique

For a graph $G = (V, E)$, a **clique** C is a subset of the nodes such that every two distinct nodes are connected (adjacent). Equivalently, a clique C in G can be defined as a complete induced subgraph. Figure 1.13 shows three examples of cliques for 2, 3 and 4 nodes.



(a) 2-nodes clique (edge)

(b) 3-nodes clique (triangle)

(c) 4-nodes clique

Figure 1.13: Clique examples on a graph

1.3.1.5 Features on graph

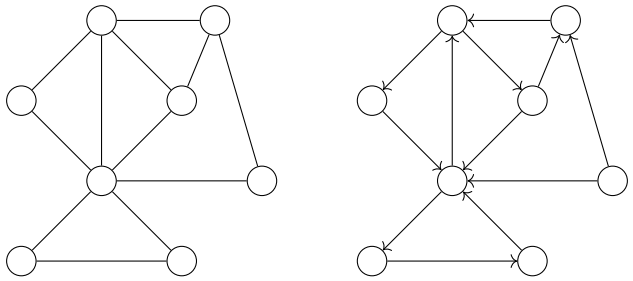
Graphs are capable of representing information at the node, edge, and global levels, which may consist of numerical values, categorical attributes, or other characteristics. This information could be defined as **features** for nodes, edges or graph, respectively. Examples of features for graph representations include:

- In social networks, node features can include name, age, and job; edge features can provide information about friendship (whether the individual is a friend), following (whether the individual follows you), and family. Additionally, the group of interest can be annotated as a global feature for the graph.
- In a molecule, atomic numbers can be used as features for atoms; the bond order serves as an edge feature, and a global feature can be assigned to determine whether the molecule is a drug.

1.3.1.6 Edges orientation

Another important remark is that the edges of a graph can define a symmetric (or non-symmetric) relation on nodes, that is, it is necessary to differentiate between two types of graphs based on edge orientation: undirected and directed graphs.

An **undirected graph** is a graph whose edges have no direction. Each edge represents a two-way relationship between two nodes and can be traversed in both directions; in other words, all edges are bidirectional. Formally, an undirected graph $G = (V, E)$ has its set of edges E described by unordered pairs of elements of V.



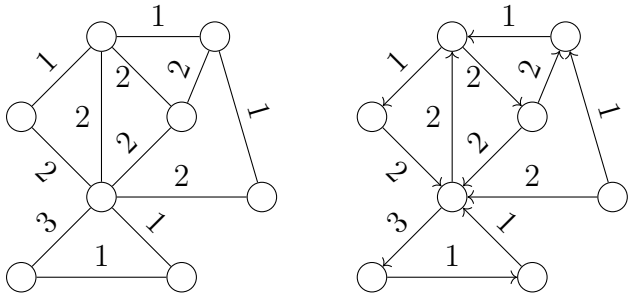
(a) Undirected graph **(b)** Directed graph

Figure 1.14: Types of graphs based on edges orientation

On the other hand, a **directed graph** (or digraph) is a graph with directed edges, that is, edges indicate a one-way relationship, where each edge, which exists between two nodes, can be traversed in a single direction. Formally, a directed graph $G = (V, E)$ has its set of edges E composed by ordered pairs of elements of V.

1.3.1.7 Edges weight

In a graph, edges can carry more information, that is, they can be assigned a **weight**, which can be generally a continuous value (even zero or negative value). The edge weights may play an important role in the graph structure, depending on the problem being addressed. In Figures 1.15a and 1.15b, examples of graphs with weighted edges are shown for an undirected graph and a directed graph, respectively.



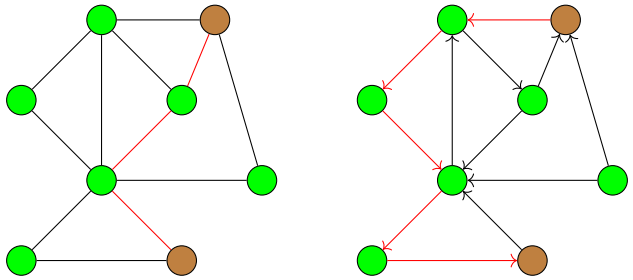
(a) Undirected graph **(b)** Directed graph

Figure 1.15: Examples of graphs with weighted edges

1.3.1.8 Paths

A **path** is a sequence (or list) of edges connecting two nodes. In Figures 1.16a and 1.16b, examples of paths (red edges) that connect two nodes (colored brown) are shown for an undirected graph and a directed graph. Thus, for an undirected graph,

it can be observed that there will be numerous paths for any two nodes due to the graph structure. However, this is not the case for a directed graph, where the number of paths for any two nodes is limited due to the orientation of the edges presented on the graph.

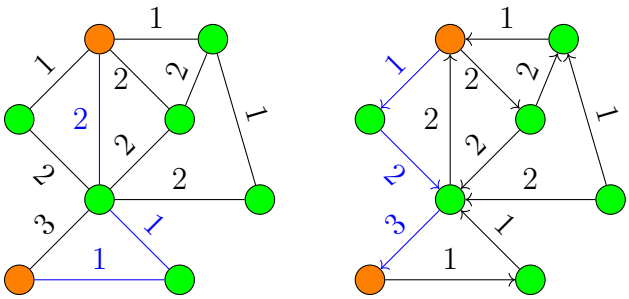


(a) Undirected graph **(b)** Directed graph

Figure 1.16: Paths on graphs

For two nodes, a **shortest path** is a path that connect these two nodes and the sum of the weights of its constituent edges is the minimum possible. In Figures 1.17a and 1.17b, examples of shortest paths (blue edges) that connect two nodes (colored orange) are shown for an undirected graph and a directed graph.

The notion of the shortest path is widely used in the resolution of many combinatorial optimization problems on graphs, where the goal is to find the paths between all pairs of nodes that involve the shortest distances to reach each of them. Classical combinatorial optimization problems on graphs include the minimum spanning tree problem [150, 216], the shortest path problem (a well-known problem in graph theory) [27, 75, 89], the traveling salesman problem [69, 154, 226, 240], and the vehicle routing problem [70, 191].



(a) Undirected graph **(b)** Directed graph

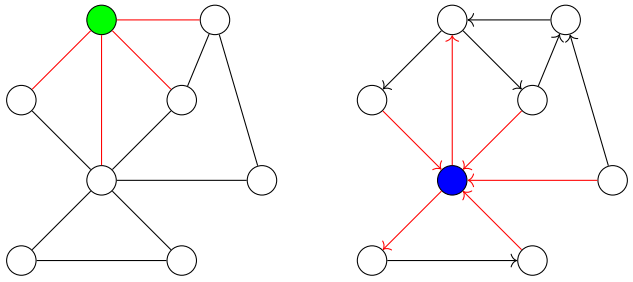
Figure 1.17: Shortest paths on weighted graphs

1.3.2 Graph measures, construction and representation

1.3.2.1 Node measures

Some node-based measures on a graph $G = (V, E)$ are introduced below. The **degree** of a node is determined by the number of edges connected to it. It must be emphasized that for a directed graph, each node has two degrees: in-degree and out-degree. The out-degree is the number of outgoing edges emanating from a node and the in-degree is the number of incoming edges onto a node. So, the total degree of a node is given by the sum of its in-degree and out-degree.

Figures 1.18a and 1.18b provide an idea of how to compute the node degree for undirected and directed graphs. For the undirected graph, the green node has degree 4 since there are 4 neighbours connected to it. Instead, the blue node from the directed graph has degree 6 because it has 4 incoming edges and 2 outgoing edges.



(a) Undirected graph **(b)** Directed graph

Figure 1.18: Node degree on graphs

The **node mean neighbor degree** measures the average degree of the node's neighborhood. This metric can reveal insights into the connectivity of the node's immediate neighborhood [23].

The **node mean shortest path length** measures the average minimum number of steps required for a node to reach all other nodes in the network. This metric can indicate how well a node is connected to other nodes within the graph [6].

Centrality measures help determine the importance of nodes within a graph. Below, the following measures are introduced: betweenness centrality, closeness centrality, and harmonic centrality.

The **node betweenness centrality** is defined as a measure of how often a node lies on the shortest path between all pairs of nodes in a network. Betweenness centrality [42] of a node u is the sum of the fraction of all-pairs shortest paths that pass through u:

$$\mathrm{BC}(u) = \sum_{s,t \in V} \frac{\sigma(s,t|u)}{\sigma(s,t)},$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest paths, and $\sigma(s,t|u)$ is the number of those paths passing through some node other than u. If $s = t, \sigma(s,t) = 1$, and if $u \in s,t$, $\sigma(s,t|u) = 0$ [41].

To normalize the betweenness values obtained, they are multiplied by $2/((|V|-1)(|V|-2))$ for undirected graphs, and $1/((|V|-1)(|V|-2))$ for directed graphs where $|V|$ is the number of nodes in G.

The **node closeness centrality** is a global metric that shows how quickly a node can communicate with other ones in the network. In a connected graph, closeness centrality [90] of a node v is the reciprocal of the average shortest path distance to v over all $|V|-1$ reachable nodes

$$\mathrm{CC}(v) = \frac{|V|-1}{\sum_{u=1}^{|V|-1} d(u,v)},$$

where $d(u,v)$ is the shortest path distance between u and v, and $|V|-1$ is the number of nodes reachable from v.

In a (not necessarily connected) graph, the **harmonic centrality** [39, 187] of a node v is the sum of the reciprocal of the shortest path distances from all other

nodes to v

$$\mathrm{HC}(v) = \sum_{u \neq v} \frac{1}{d(u,v)},$$

where $d(u,v)$ is the shortest path distance between u and v.

The following illustrates how to compute these centrality measures using a small example. The undirected graph G_0, which has 7 nodes and is represented in Figure 1.19, is considered. It is also assumed that each edge in G_0 has a weight equal to 1.

Firstly, the shortest path distances between all pairs of nodes on G_0 are computed. These calculations can be visualized in Table 1.1. To compute betweenness centrality, every pair of nodes in the graph is considered, and the number of times a node can interrupt the shortest paths between the two nodes of the pair is counted. Then, the betweenness values are multiplied by $2/((7-1)(7-2) = 1/15$ (since G_0 is undirected) to normalize them.

To calculate closeness centrality and harmonic centrality, the shortest path distances previously calculated are used, and the respective formulas are applied.

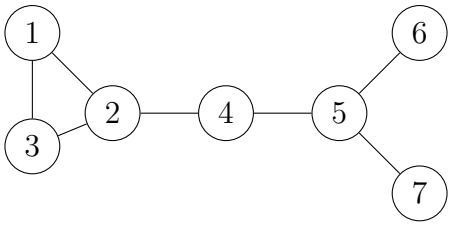


Figure 1.19: Undirected graph G_0

Table 1.1: Shortest path distances on graph G_0

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	-	1	1	2	3	4	4
2	1	-	1	1	2	3	3
3	1	1	-	2	3	4	4
4	2	1	2	-	1	2	2
5	3	2	3	1	-	1	1
6	4	4	3	2	1	-	2
7	4	3	4	2	1	2	-

Table 1.2: Node centrality measures on graph G_0

Node	**Betweenness centrality (normalized)**	**Closeness centrality**	**Harmonic centrality**
1	0	6/15	10/3
2	13/45	6/11	25/6
3	0	6/15	10/3
4	13/45	6/10	4
5	16/45	6/11	25/6
6	0	6/16	17/6
7	0	6/16	17/6

The **local clustering coefficient** of a node quantifies how close its neighbours are to being a clique (complete graph). Let $G = (V, E)$ a graph, and e_{ij} is an edge

that connects nodes i and j. Then, the neighborhood N_i of node i is defined as

$$N_i = \{j : e_{i,j} \in E \vee e_{j,i} \in E\},$$

and k_i as the number of nodes in the neighbourhood of node i. Then, the local clustering coefficient $LCc(i)$ for a node i is given by a proportion of the number of connections between the nodes within its neighborhood divided by the total number of connections that could exist between them. For directed graphs [279], the local clustering coefficient is defined as

$$LCc(i) = \frac{|\{e_{j,k} : j, k \in N_i, e_{j,k} \in E\}|}{k_i(k_i - 1)},$$

and the local clustering coefficient for undirected graphs [279] is defined as

$$LCc(i) = \frac{2|\{e_{j,k} : j, k \in N_i, e_{j,k} \in E\}|}{k_i(k_i - 1)},$$

where $|\cdot|$ represents the cardinality of the set.

An example is provided to illustrate the computation of the local clustering coefficient for a given node in three scenarios. An undirected graph composed of four nodes is considered, where the node (green) i_g has three neighbors. Different ways of representing the links between the neighbors of the green node are explored.

Firstly, the total number of connections that could exist between its neighbors is 3. In Figure 1.20a, it is assumed that all its neighbors are connected to each other. The local clustering coefficient is then calculated as follows

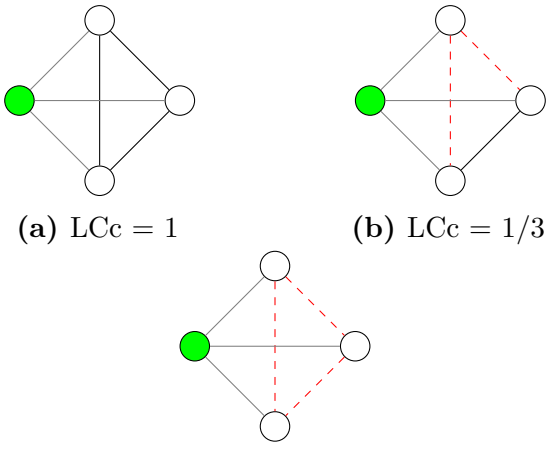
$$LCc(i_g) = \frac{3}{3} = 1.$$

Figure 1.20b illustrates the scenario when there is only one link within the neighborhood of the green node. Therefore, the local clustering coefficient is

$$LCc(i_g) = \frac{1}{3}.$$

Finally, it is assumed that there are no links within the neighborhood of the green node, which is shown in Figure 1.20c. Then, the local clustering coefficient is

$$LCc(i_g) = \frac{0}{3} = 0.$$



(a) LCc = 1 **(b)** LCc = 1/3

(c) LCc = 0

Figure 1.20: Local clustering coefficient on a node

A k-core is a maximal subgraph that contains nodes of degree k or more. The **core number** [25] of a node is the largest value k of a k-core containing that node. Figure 1.21 provides an example of an undirected graph used to compute the core numbers of its nodes. In this figure, it is noted that orange nodes have a core number equal to 1 since their maximal subgraph is composed of themselves and their node neighbor (degree 1). Blue nodes have a core number of 2 because their possible maximal subgraphs consist of themselves with two node neighbours (degree 2). Finally, red nodes have a core number equals to 3 since their maximal subgraph corresponds to the complete graph formed by themselves (where they have degree 3).

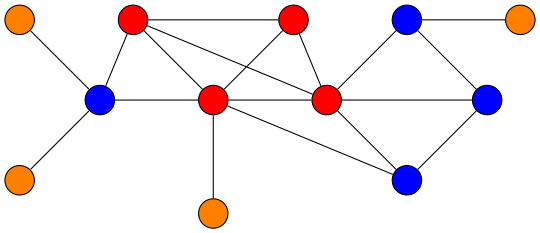


Figure 1.21: Core number on a graph

1.3.2.2 Edge measures

As for the nodes, betweenness centrality can be also computed on edges. Betweenness centrality [42] of an edge e is the sum of the fraction of all-pairs shortest paths that pass through e

$$\mathrm{BC}(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)},$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest paths, and $\sigma(s,t|e)$ is the number of those paths passing through edge e [41].

In addition, measures related to the distance or similarity between pairs of nodes can be calculated, such as: Euclidean distance, relative displacement, cosine similarity, etc.

Cosine similarity is a measure of how different two points (or vectors) are based on their orientation rather than their magnitude. Given two nodes u, v with their x-y positions $\boldsymbol{u_{x,y}} = (u_x, u_y)$, $\boldsymbol{v_{x,y}} = (v_x, v_y)$, its cosine similarity ($\cos(\theta_{u,v})$) is defined by

$$\cos(\theta_{u,v}) = \frac{\langle \boldsymbol{u_{x,y}}, \boldsymbol{v_{x,y}} \rangle}{\|\boldsymbol{u_{x,y}}\| \|\boldsymbol{v_{x,y}}\|} = \frac{u_x v_x + u_y v_y}{\sqrt{u_x^2 + u_y^2}\sqrt{v_x^2 + v_y^2}}.$$

Besides, the **euclidean distance** between u and v (denoted by $d_{u,v}$) is given by

$$d_{u,v} = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}.$$

Finally, the **relative displacement** between u and v (denoted by $r_{u,v}$) is defined by

$$r_{u,v} = \boldsymbol{u_{x,y}} - \boldsymbol{v_{x,y}} = (u_x - v_x, u_y - v_y).$$

1.3.2.3 Graph measures

The **number of connected components** in a graph can also be computed to discern global arrangements within the network. In this context, a connected component is defined as a subset of vertices in a graph that are interconnected by paths.

Density: It is the ratio of the number of edges with respect to the maximum possible edges. The density D for an undirected graph $G = (V, E)$ is defined as

$$D = \frac{2|E|}{|V|(|V|-1)},$$

and for directed graphs is

$$D = \frac{|E|}{|V|(|V|-1)},$$

where $|V|$ is the number of nodes and $|E|$ is the number of edges in G.

The structural characteristics of the networks can be quantified using specific network connectivity metrics, including the global clustering coefficient and network assortativity. The **global clustering coefficient** (1.3.2.3) describes the tendency of the network to build triangles by relating triplets of nodes to each other [180]. A *closed triplet* (see Figure 1.22a) is defined as three nodes that are connected by three edges, while a *open triplet* (see Figure 1.22b) represents three nodes that are connected by two edges [195].



(a) Closed triplet **(b)** Open triplet

Figure 1.22: Closed and open triplets

Then, the global clustering coefficient (GClc) for an undirected graph is defined as

$$\text{GClc} = \frac{\text{number of closed triplets}}{\text{total number of triplets}}$$

Network assortativity (1.3.2.3) describes the preference of nodes to be connected with others of similar properties, here specifically, in terms of the node degree (see Figure 1.23a). Conversely, nodes that tend to connect with others having dissimilar properties are known as disassortativity mixing (see Figure 1.23b). The **assortativity coefficient** for an undirected graph is described as

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i},$$

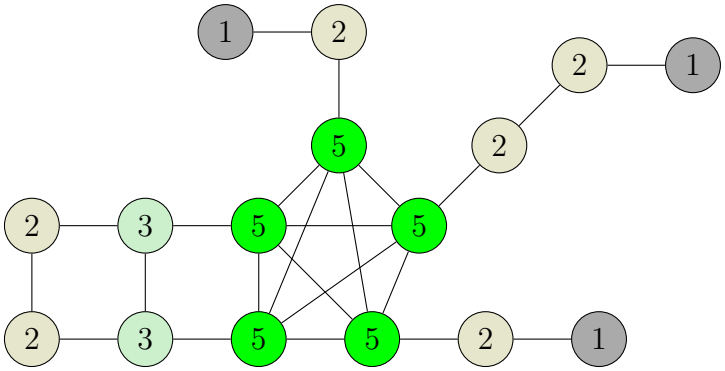
where e_{ij} is the fraction of edges connecting nodes of type i and j, a_i is the sum over e_{ij} for all j, and b_i is the sum over e_{ij} for all i [180]. No mixing preference is shown for an assortativity coefficient $r = 0$, while positive values mean assortative tendencies and negative values indicate disassortative tendencies [180].

Additionally, the following measures are introduced to represent global patterns in the network structure:

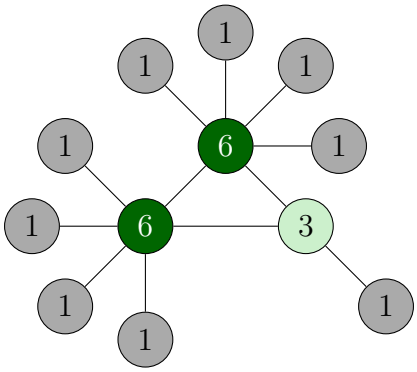
Global efficiency: The efficiency of a pair of nodes in a graph is the multiplicative inverse of the shortest path distance between the nodes. The average global efficiency of a graph is the average efficiency of all pairs of nodes [153]. Global efficiency examines how efficiently information is exchanged over the network [153, 231].

For an undirected graph $G = (V, E)$, its global efficiency E_{glob} is given by

$$E_{glob} = E(G) = \frac{1}{|V|(|V|-1)} \sum_{i \neq j} \frac{1}{d(i,j)}$$



(a) Assortativity network



(b) Disassortativity network

Figure 1.23: Examples of assortativity and disassortativity graphs

where $d(i, j)$ is the shortest path distance between nodes i and j, and $|V|$ is the total number of nodes in G.

Local efficiency: The local efficiency of a node in the graph is the average global efficiency of the subgraph induced by the neighbors of the node. The average local efficiency is the average of the local efficiencies of each node [153]. Local efficiency reveals how the network is fault tolerant, that is, how efficiently is information transmitted locally when a node is removed [153, 231].

For an undirected graph $G = (V, E)$, its local efficiency E_{loc} is given by

$$E_{loc} = \frac{1}{|V|} \sum_{i \in V} E(G_i)$$

where $E(G_i)$ is the subgraph induced by the neighbors of node i, and $|V|$ is the number of nodes in G.

To facilitate a better understanding of these metrics, an example of their computation is provided below. Let $H = (V, E)$ be an undirected graph with 5 nodes and 6 edges as shown in Figure 1.24. It is also assumed that each edge in H has a weight equal to 1.

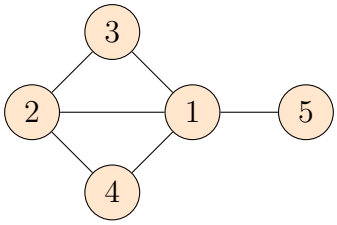


Figure 1.24: Undirected graph H

To compute the global efficiency of H, the shortest path distances between all pairs of nodes are first calculated. These distances can be observed in Table 1.3, and their reciprocal values are illustrated in Table 1.4. Then, by summing all reciprocal shortest path distances, the global efficiency of H is determined as follows

$$E(H) = \frac{1}{20} \sum_{i \neq j} \frac{1}{d(i,j)} = \frac{16}{20} = 0.80.$$

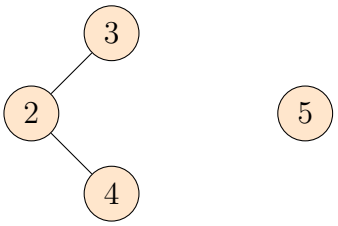
Table 1.3: Shortest path distances on graph H

	1	**2**	**3**	**4**	**5**
1	-	1	1	1	1
2	1	-	1	1	2
3	1	1	-	2	2
4	1	1	2	-	2
5	1	2	2	2	-

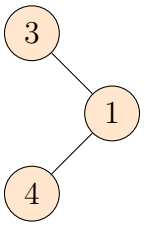
Table 1.4: Reciprocal shortest path distances on graph H

	1	**2**	**3**	**4**	**5**
1	-	1	1	1	1
2	1	-	1	1	1/2
3	1	1	-	1/2	1/2
4	1	1	1/2	-	1/2
5	1	1/2	1/2	1/2	-

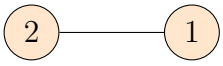
To compute the local efficiency of H, one must determine the subgraph induced by the neighbors of each node and then calculate the global efficiency of the resulting subgraphs. Figure 1.25 illustrates the subgraph induced by the neighbors of each node in H.



(a) Subgraph H_1



(b) Subgraph H_2



(c) Subgraphs H_3, H_4



(d) Subgraph H_5

Figure 1.25: Subgraph induced by the neighbors of each node in H

For the subgraph induced by the neighbors of node 1, denoted as H_1 (Figure 1.25a), the shortest path distances and their reciprocal values are shown in Tables 1.5 and 1.6, respectively. The global efficiency of H_1 is

$$E(H_1) = \frac{5}{12}.$$

Table 1.5: Shortest path distances on subgraph H_1

	2	**3**	**4**	**5**
2	-	1	1	0
3	1	-	2	0
4	1	2	-	0
5	0	0	0	-

Table 1.6: Reciprocal shortest path distances on subgraph H_1

	2	**3**	**4**	**5**
2	-	1	1	0
3	1	-	1/2	0
4	1	1/2	-	0
5	0	0	0	-

Tables 1.7 and 1.8 show the shortest path distances and their reciprocal values, respectively for the subgraph H_2 (Figure 1.25b). The global efficiency of H_2 is

$$E(H_2) = \frac{5}{6}.$$

Table 1.7: Shortest path distances on subgraph H_2

	1	**3**	**4**
1	-	1	1
3	1	-	2
4	1	2	-

Table 1.8: Reciprocal shortest path distances on subgraph H_2

	1	**3**	**4**
1	-	1	1
3	1	-	1/2
4	1	1/2	-

In this example, the subgraphs induced by the neighbors of nodes 3 and 4 (Figure 1.25c) are the same. These subgraphs, denoted as H_3 and H_4, consist of nodes 1 and 2, which are connected by an edge. Therefore, the shortest path distance between these nodes is 1. The global efficiency for these subgraphs is

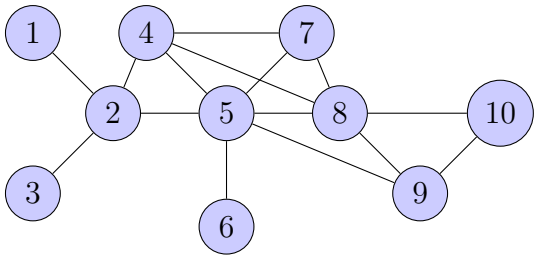
$$E(H_3) = E(H_4) = \frac{2}{2} = 1.$$

The subgraph induced by the neighbors of node 5 (Figure 1.25d), denoted as H_5, consists solely of node 1, since node 5 has only node 1 as its neighbor. Consequently, its global efficiency is zero.

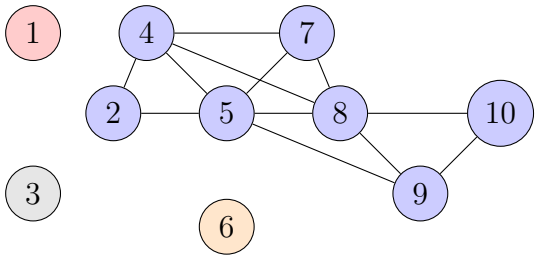
Finally, the local efficiency of H is given by

$$E_{loc} = \frac{1}{5}\left(\frac{5}{12} + \frac{5}{6} + 1 + 1 + 0\right) = \frac{13}{20} = 0.65.$$

Number of bridges: a bridge in a graph is defined as an edge whose removal increases the number of connected components in the graph. This metric is quantified as the total number of bridges present in the graph. To illustrate this concept, consider the example presented in Figure 1.26a, which depicts an undirected graph with 10 nodes and a single connected component. If the edges (1,2),(2,3), and (5,6) of this graph are removed iteratively, the number of connected components starts to increase until it becomes 4 (the maximum possible). Figure 1.26b shows the resulting graph after cutting the edges mentioned above. Therefore, the number of bridges for this graph is 3 because only 3 edges need to be removed to increase the total number of connected components.



(a) Original graph



(b) Graph after cutting edges (bridges)

Figure 1.26: Example of bridges on a graph

1.3.2.4 Graph construction

This section aims to introduce the various graph construction methods employed to address the applications examined in this thesis.

Most of the graphs utilized in other research areas, such as social networks and recommendation systems, already have their topological structures predefined; that is, their edges are known. However, for the majority of the datasets employed in this work, which consist of 2D microscopic images of cell cultures, graphs cannot be easily inferred. In other words, the construction of the edges to define the topology of the graph remains unclear. It is important to note an exception for the rest of datasets handled in this work, corresponding to biological networks that form cellular tissues. In this case, the graph can be derived from the skeleton obtained from the binarized image, as will be illustrated in Chapter 2.

For datasets consisting of 2D microscopic images of cell cultures, the idea is to build a graph by relating the spatial proximity between nodes (cells) in an image. This approach can also be used to infer a notion of temporal proximity between two distinct and ordered frame images.

The notion of spatial proximity involves constructing graph edges by fixing a connectivity radius (or threshold) τ. Specifically, if the distance between two nodes is less than or equal to τ, an edge is created between them; otherwise, no edge is formed. This type of construction is referred to as a **Distance graph**.

Below, some graph structures used in the different applications described in the following chapters are presented:

Delaunay graph is based on Delaunay triangulation, which subdivides the convex hull of a set of points in the plane into triangles whose circumcircles contain no other points [72]. For a given set of points, the Delaunay graph is defined by the edges corresponding to the sides of the triangles that satisfy the triangulation condition (see Figure 1.27). Furthermore, a threshold is applied to remove some distant edges if necessary.

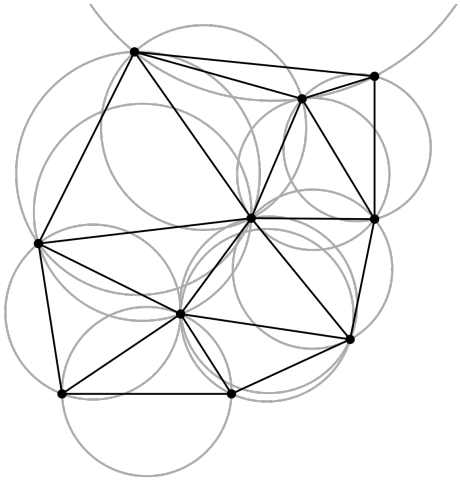


Figure 1.27: Delaunay triangulations. Source: Wikipedia[1]

Gabriel graph is a subgraph of the Delaunay triangulation. Given a node set S of the Gabriel graph G, if for any two distinct points $a \in S$ and $b \in S$, the closed disc having ab as a diameter contains no other points, then, a and b are adjacent (see Figure 1.28); that is, (a, b) is an edge on G [92]. A threshold is also applied to remove some distant edges if necessary.

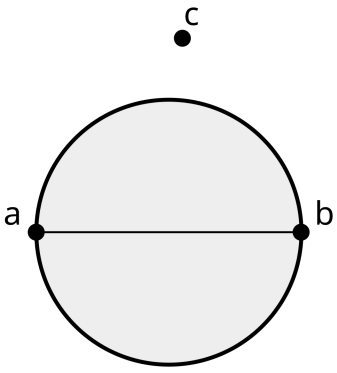


Figure 1.28: Gabriel node pairs. Source: Wikipedia[2]

[1] https://en.wikipedia.org/wiki/Delaunay_triangulation

Urquhart graph is also a subgraph of the Delaunay triangulation. For a given set of points, the Urquhart graph is obtained by removing the longest edge from each of triangle in the Delaunay triangulation [266]. As before, a threshold is applied to remove some distant edges if necessary.

Figure 1.29 shows an example of an Urquhart graph. The longest edges, which are thin and cyan, are removed from each Delaunay triangle.

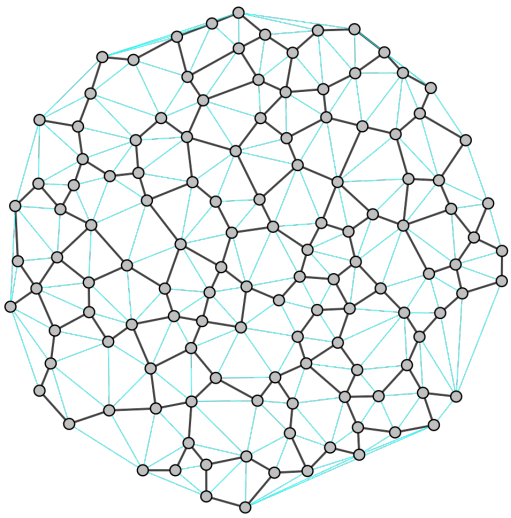


Figure 1.29: Example of Urquhart graph. Source: Wikipedia[3]

The ***k*-Nearest Neighbors (*k*-NN) graph** [18] is defined as a graph $G = (V, E)$ where an edge connects two nodes, u and v, if the distance between them is one of the k smallest distances from u to any other node in V. In Figure 1.30, an example of a k-NN graph (k=1) of 100 points in the Euclidean space is observed.

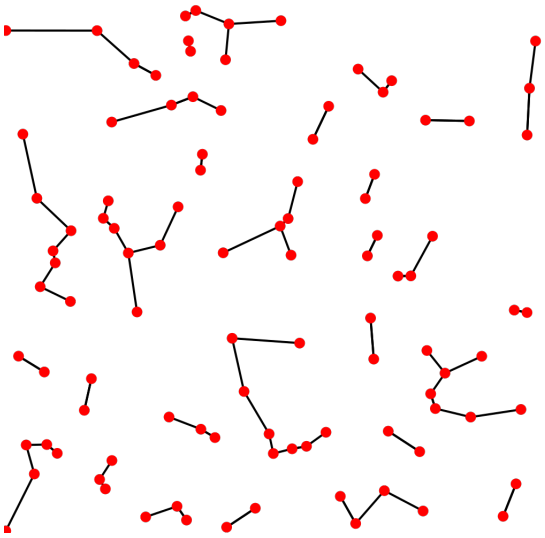


Figure 1.30: Example of k-NN graph ($k = 1$). Source: Wikipedia[4]

[2] https://en.wikipedia.org/wiki/Gabriel_graph
[3] https://en.wikipedia.org/wiki/Urquhart_graph
[4] https://en.wikipedia.org/wiki/Nearest_neighbor_graph

1.3.2.5 Graph representation

Graphs can be stored and their structure retrieved (at any time) through three main representations: an adjacency matrix, an adjacency list, and a sparse matrix. These representations also allow for working efficiently on graphs, that is, performing calculations on them.

Each of these representations is described below.

Definition 1.2 (Adjacency matrix)**.** Let be $G = (V, E)$ a graph. The adjacency matrix of G is defined as a matrix $A \in \mathcal{M}_{|V|\times|V|}$, where each entry of A is given by

$$a_{(i,j)} = \begin{cases} 1 & \text{if there is an edge between nodes } i \text{ and } j. \\ 0 & \text{if not} \end{cases}$$

for $i, j = 1, 2, \ldots, V$.

Then, for the next undirected graph G_1

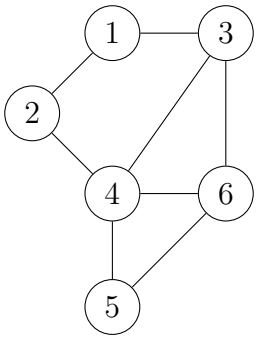


Figure 1.31: Undirected graph G_1

Its adjacency matrix is

$$A_{G_1} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Alternatively, consider the following directed graph G_2

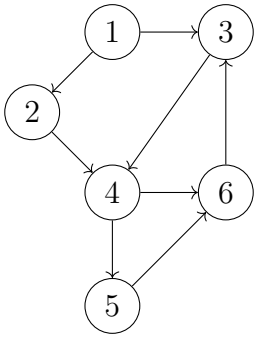


Figure 1.32: Directed graph G_2

So, its adjacency matrix is

$$A_{G_2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

It is important to note that A_{G_1} is a symmetric matrix, whereas $A_{G_2}$is a non-symmetric matrix.

Definition 1.3 (Adjacency list)**.** Let be $G = (V, E)$ a graph. The adjacency list L_G of G is a collection of unordered lists, where each unordered list within L_G describes the set of neighbors of a particular node in G.

Consider the directed graph G_2 depicted in Figure 1.32. The adjacency list L_{G_2} for this graph can be represented as follows

$$L_{G_2} = \begin{cases} \boxed{1} \longrightarrow \boxed{2} \longrightarrow \boxed{3} \\ \boxed{2} \longrightarrow \boxed{4} \\ \boxed{3} \longrightarrow \boxed{4} \\ \boxed{4} \longrightarrow \boxed{5} \longrightarrow \boxed{6} \\ \boxed{5} \longrightarrow \boxed{6} \\ \boxed{6} \longrightarrow \boxed{3} \end{cases}.$$

Alternatively, L_{G_2} can be described using the following matrix representation

$$L_{G_2} = \begin{bmatrix} [1,2], & [1,3] \\ [2,4] & \\ [3,4] & \\ [4,5], & [4,6] \\ [5,6] & \\ [6,3] & \end{bmatrix}.$$

Nodes and edge weights, if available, can also be stored as lists. In this example, it can be assumed that the list of nodes L_V is

$$L_V = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0],$$

and the list of edge weights L_E is

$$L_E = [0.3, 0.4, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0].$$

Both introduced representations have some advantages and disadvantages. The adjacency matrix has a scale of $|V|^2$ (where $|V|$ is the number of nodes), which means it occupies a lot of space in memory. However, it can be sparse if there are not many edges in the graph. Furthermore, it requires $N!$ permutations to represent the same graph, and it is easy to find an edge since one only needs to access the position (row and column) on the matrix. On the other hand, the adjacency list has a scale of $|E|$

(where $|E|$ is the number of edges), which means it occupies less space in memory. However, it might be difficult to find an edge since, in the extreme case, one has to visit at most $|V|-1$ entries of a node to find the edge, involving $|V|-1$ operations. In contrast, for the adjacency matrix, only 1 operation is needed to find the edge.

An alternative to the adjacency list, is the use of a sparse matrix data structure. In such case only the non-zero elements are kept along with their coordinates and everything else is discarded as non-informative [147]. This data structure is defined below.

Definition 1.4 (Sparse matrix)**.** Let be $G = (V, E)$ a graph and $A \in \mathcal{M}_{|V|\times|V|}$ its adjacency matrix. The sparse matrix of G is defined as a matrix $Sp_G \in \mathcal{M}_{2\times|E|}$ (or $\mathcal{M}_{3\times|E|}$ for weighted graphs), where the first row keeps the i coordinate for each element in $A(i, j)$, and the second row the j coordinate in $A(i, j)$. For weighted graphs, the third row keeps the weight associated to each edge.

For the directed graph G_2, given in Figure 1.32, its sparse matrix is

$$Sp_{G_2} = \begin{pmatrix} 1 & 1 & 2 & 3 & 4 & 5 & 4 & 6 \\ 2 & 3 & 4 & 4 & 5 & 6 & 6 & 3 \end{pmatrix}$$

Generally, the sparse matrix is the primary data structure used in most GNN-based models due to its efficient storage of information.

Finally, the notation for some useful matrices is introduced:

- $\boldsymbol{A}$: the adjacency matrix, which could include edge weights.
- $\boldsymbol{D}$: the node degree matrix, which is a diagonal matrix with the number of edges for each node.
- $\boldsymbol{L}$: the Laplacian matrix, which is used on spectral analysis and is defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$.
- $\boldsymbol{X}$: the node features matrix, which contains information stored on nodes.
- $\boldsymbol{X_E}$: the edge features matrix, which contains information stored on edges.

1.3.3 Learning on graphs

After introducing the definition of a graph and some notions about this type of data structure, it will be shown below how learning on graphs is done and what types of predictive tasks can be performed on them.

1.3.3.1 Graph embedding

Figure 1.33 illustrates the procedure for performing graph learning, depending on the specific prediction task. First, the node features matrix $\boldsymbol{X}$, adjacency matrix $\boldsymbol{A}$ (and even the edge features matrix $\boldsymbol{X_E}$) are input into an encoder network, which is generally a neural network architecture. This network transforms (or encodes) their information into a latent space denoted by Z, which is generally a matrix of dimension lower than the input data. This process is known as graph embedding, as it encodes the structure of the graph, given by $\boldsymbol{X}$, $\boldsymbol{X_E}$ and $\boldsymbol{A}$, into a better representation of lower dimension. Subsequently, the embedding matrix $\boldsymbol{Z}$ is passed

through a decoder network, which is generally a neural network architecture, to obtain the estimated outputs depending on the predictive task to be performed. In this case, estimated labels for nodes, edges, or graphs can be obtained to perform labeling, classification, or regression tasks on nodes; link prediction on edges; and graph classification. Besides, an estimate of the adjacency matrix can be retrieved, which is used to optimize the latent space in encoder-decoder architectures for graphs, such as Graph Auto-Encoders or Variational Graph Auto-Encoders [145].

The embedding matrix $\boldsymbol{Z}$ has to be suited for the task, that is, it has to be learnable in the sense that certain properties or characteristics of the graph structure should be identifiable in this new space; otherwise, the model will not be able to learn the dynamics and information necessary to perform well the assigned task.

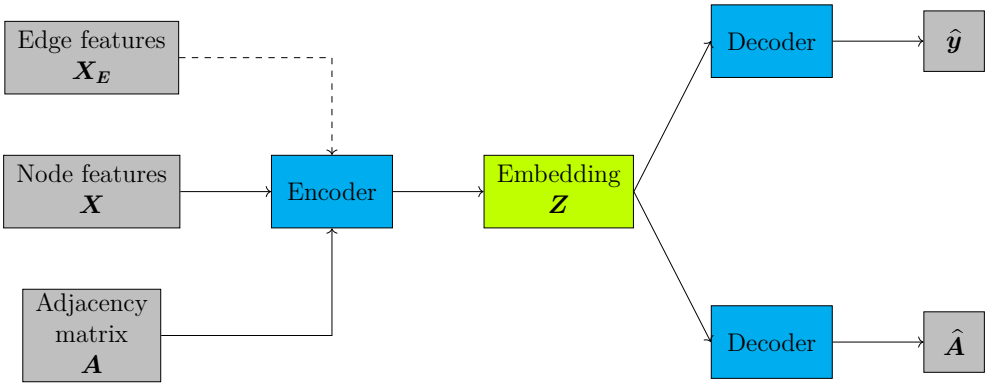


Figure 1.33: Graph embedding

1.3.3.2 Inductive and Transductive learning

It is necessary to distinguish between two types of learning: inductive and transductive learning. On the one hand, the general principle of **inductive learning** is that a subset of graphs from the dataset is used to train the model, while the remaining data is reserved for evaluating its performance. This allows the generalization to new graphs since the model is evaluated by using unseen data. When treating the dataset solely as a graph, inductive learning assumes that the model has access only to a subset of the graph (the training set), while allowing the addition of new nodes.

On the other hand, **transductive learning** assumes that the model has access to the entire dataset, which means it cannot generalize to new data, as training is performed using all available samples. In case of considering the dataset as just a graph, the transductive learning refers to the model for its access to the complete graphs, and it is not possible to add new nodes. In this context, it is possible to label nodes, which were masked during the training phase or find new edges on the graph.

1.3.3.3 Prediction tasks

The following presents some applications that can be performed on nodes, edges, or entire graphs (see Figure 1.34).

For **nodes**, prediction tasks include labeling nodes in a graph (clustering); identifying the topic of a research paper (CORA[5] dataset); detecting bots in a social network or identifying different communities in a social graph (community detection); labeling new nodes (classification); and performing regression.

[5] https://graphsandnetworks.com/the-cora-dataset/

For **edges**, relationships can be inferred (link prediction), such as the contact map of amino acids (AlphaFold[6]); contact suggestions in social networks; directions between two locations to optimize estimated time of arrival (ETA); citations between articles; and relationships between segments in images.

For **graphs**, similarities between entire graphs can be assessed, or properties such as chemical characteristics (e.g., solubility, carcinogenicity, potential drug candidates) can be predicted. Graph-level classification tasks also include identifying research fields within an ego network.

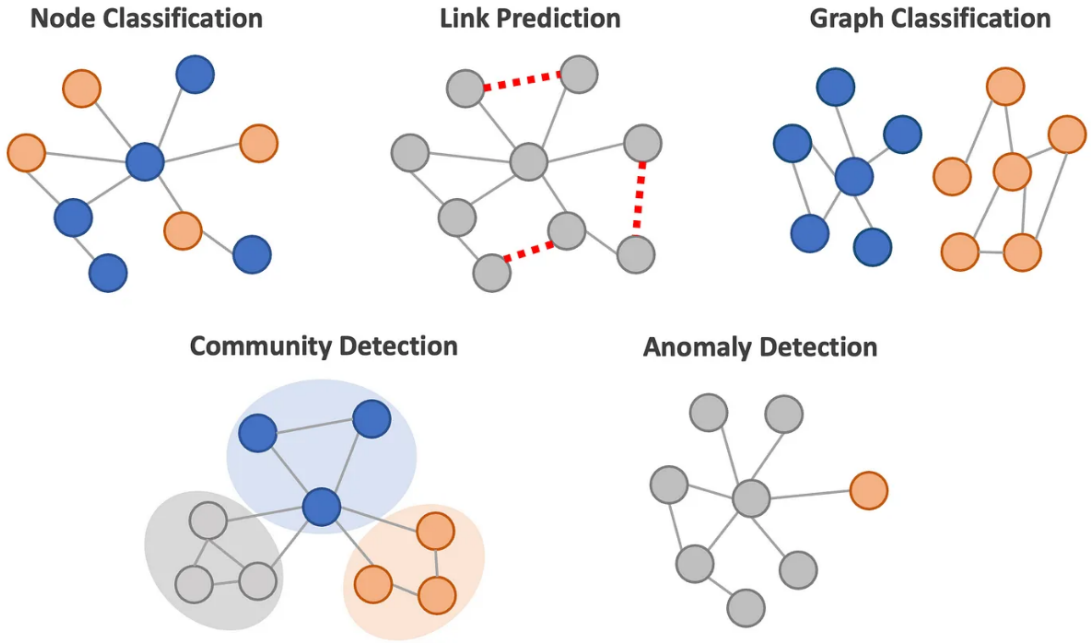


Figure 1.34: Some examples of graph prediction tasks. Source: https://short-link.me/medium-gnn-applications

1.3.4 GNN methods

This section explains how a Graph Neural Network (GNN)-based model works. Specifically, the principal GNN layers used are presented and described based on the manner in which propagation is performed. These layers form the basis for developing both new and existing state-of-the-art GNN models.

As observed or described so far, Graph Neural Networks (GNNs) [238] generally take as input a graph endowed with node and edge features, along with its adjacency matrix, and compute a function that depends on both the features and the graph structure. In this context, an architecture is introduced below that allows the transmission of information between nodes through the notion of messages, which is known as Message Passing Neural Networks (MPNN) [96]. This type of structure propagates node features by exchanging information between adjacent nodes. A classic MPNN architecture has several propagation layers, where each node is updated based on the aggregation of its neighbors' features. These aggregation functions are usually parametric, and generally come in 3 different types of graph neural network: graph convolution, message passing and graph attention (see Figure 1.35).

In the following, each of these aggregation functions will be described.

[6] https://www.deepmind.com/research/highlighted-research/alphafold

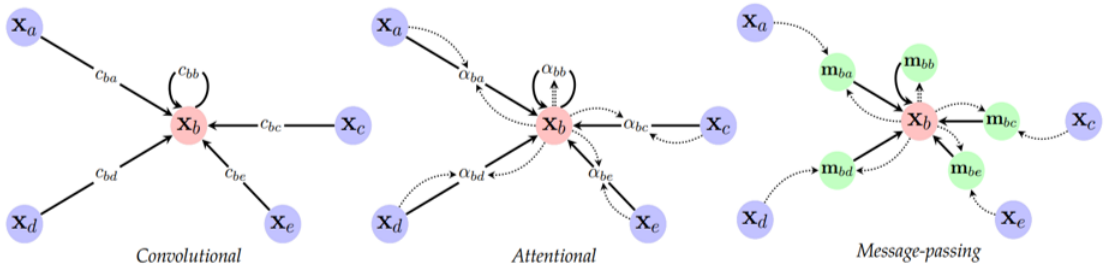


Figure 1.35: Example dataflows in three mainly types of GNN: graph convolution (left), graph attention (middle) and message passing (right). Source: https://blogs.nvidia.com/blog/what-are-graph-neural-networks/

1.3.4.1 Graph convolution

This method enables the extension of the convolution operator by introducing the neighborhood aggregation, where the information of a given node is updated from its neighbors' information (see Figure 1.36). It is possible to make use of graph convolutions known from spectral graph theory ([47, 116]) to define parameterized filters that are used in a multi-layer neural network model, in order to replicate a "classical" convolutional neural network (CNN).

This new convolution operator must be permutation invariant[7] since this property is a key inductive bias for graph representation learning [194]. This condition could become challenging since the number of neighbors is unlikely to be constant. This operator is defined as follows.



Figure 1.36: Graphical representation of the convolution mechanism. Node features from neighboring nodes (blue) and the target node (red) are averaged together (in practice, the node features are normalized using the inverse of the degree matrix instead of simple averaging). The resulting value is then combined with a weight vector (W), updating the target node's features with the output. Source: https://short-link.me/medium-gnn-applications

Let $G = (V, E)$ be a graph with N nodes: $\{v_i\}_{i \in \{1,\dots,N\}}$, a node features matrix $\boldsymbol{X} \in \mathbb{R}^{N \times F}$ with F the number of input features and its adjacency matrix $\boldsymbol{A}$. Then, as previously stated, for a MPNN model, the goal is to learn a function of features on G, which produces a node-level output $\boldsymbol{Z} \in \mathbb{R}^{N \times F}$, where F is the number of output features per node. Then, every neural network layer can then be written as a

[7] **Permutation invariant** means the permutation of the nodes of the input graph does not affect the output [127].

non-linear function

$$\boldsymbol{H}^{(l+1)} = f(\boldsymbol{H}^{(l)}, \boldsymbol{A}), \tag{1.1}$$

with $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ and $\boldsymbol{H}^{(L)} = \boldsymbol{Z}$, L being the number of layers. Then, the (basic) l-th layer graph convolution (network), given by f, is defined as

$$f(\boldsymbol{H}^{(l)}, \boldsymbol{A}) = \sigma\left(\boldsymbol{A}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \tag{1.2}$$

where $\boldsymbol{W}^{(l)}$ is a weight matrix for the l-th neural network layer and $\sigma(\cdot)$ is a non-linear activation function. However, f presents two limitations: the first one consists of multiplying by $\boldsymbol{A}$ means that, for every node, all the feature vectors of all neighboring nodes are summed, excluding the node itself unless there are self-loops in the graph. This can be addressed by enforcing self-loops in the graph, that is, by simply adding the identity matrix to $\boldsymbol{A}$.

The second limitation is that $\boldsymbol{A}$ is generally not normalized and therefore the multiplication with $\boldsymbol{A}$ will completely change the scale of the feature vectors. In practice, the matrix $\boldsymbol{A}$ is normalized by $\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{D}$ is the diagonal node degree matrix.

By combining these two solutions, the l-th layer graph convolution introduced in [146] is obtained

$$f(\boldsymbol{H}^{(l)}, \boldsymbol{A}) = \sigma\left(\hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-1/2}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \tag{1.3}$$

with $\hat{\boldsymbol{A}} = \boldsymbol{A} + I$, where I is the identity matrix and $\hat{\boldsymbol{D}}$ is the diagonal node degree matrix of $\hat{\boldsymbol{A}}$.

For a graph convolution layer, several steps are needed to retrieve information for distant nodes. In the case for large graphs, it might be useful to apply a cutoff. It is possible to use a virtual node connected to all other nodes, but in practice, this becomes quickly intractable.

1.3.4.2 Message passing

This type of structure was introduced in [96]. The idea behind the notion of message passing is to aggregate the neighbors' messages computed from a given node to update its node features. Formally, this method is described as follows for a single message passing layer.

Considering, $G = (V, E)$ a graph with N nodes: $\{v_i\}_{i\in\{1,\dots,N\}}$, a node features matrix $\boldsymbol{X} \in \mathbb{R}^{N\times F}$ and an (optional) edge features matrix $\boldsymbol{X_E} \in \mathbb{R}^{N\times F'}$ where F and F' are the number of input features for nodes and edges respectively. Then, with $\mathbf{x}_i \in \mathbb{R}^F$ denoting node features of node i and $\boldsymbol{e}_{ij} \in \mathbb{R}^{F'}$ denoting edge features from node j to node i, message passing layer can be described as

$$\mathbf{h}_i = \gamma\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} \varphi\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{e}_{ij}\right)\right) \tag{1.4}$$

where $\mathbf{h}_i$ is the node features update for node i, $\mathcal{N}_i$ is some neighborhood of node i in the graph, $\bigoplus$ denotes a differentiable, permutation invariant function, e.g. sum, average or max; and γ, φ denote differentiable functions such as Multi Layer Perceptrons (MLPs) [228].

As can be seen in Equation 1.4, messages are computed from neighbors of node i and aggregated by

$$\mathbf{msg}_i = \bigoplus_{j \in \mathcal{N}_i} \varphi\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{e}_{ij}\right). \tag{1.5}$$

Finally, the node features update of node i is computed by using its previous state and the messages aggregated

$$\mathbf{h}_i = \gamma\left(\mathbf{x}_i, \mathbf{msg}_i\right). \tag{1.6}$$

Embeddings can also be generated for each part of the graph (possibly with different vector sizes), where each one can learn from the others via a transformation.

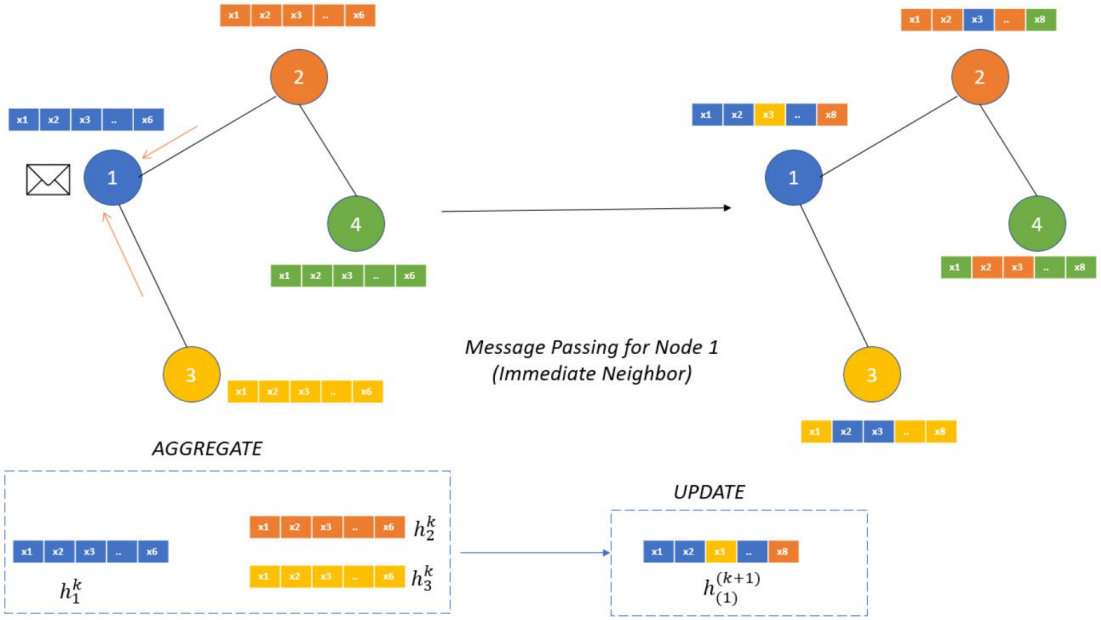


Figure 1.37: Graphical representation of the message passing mechanism. For node 1 with neighborhood nodes 2 and 3, the features $h_1^{(k+1)}$ and the features for its immediate neighbors $h_2^{(k+1)}, h_3^{(k+1)}$ are first aggregated. Once the aggregation is done then it will update its own state $h_{(1)}^{(k+2)}$. Source: https://www.aritrasen.com/graph-neural-network-message-passing-gcn-1-1/

1.3.4.3 Graph attention

The mechanism of attention on graphs was introduced in [269]. This approach employs a self-attention mechanism over the node features. The method for a single *graph attentional layer* will then be described (see Figure 1.38).

Let be $G = (V, E)$ a graph with N nodes: $\{v_i\}_{i \in \{1,\ldots,N\}}$. So, the input data to the layer is a set of node features, $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$, $\vec{h}_i \in \mathbb{R}^F$, where F is the number of features in each node. The layer produces a set of node features (of potentially different cardinality F') $\mathbf{h'} = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}$, $\vec{h}_i \in \mathbb{R}^{F'}$, as its output.

So, the input features are transformed into higher-level features (where at least one learnable linear transformation is required) to obtain sufficient expressive power. For this purpose, a shared linear transformation, parametrized by a weight matrix, $\mathbf{W} \in \mathbb{R}^{F' \times F}$, is applied to every node. Then, self-attention is performed on the nodes;

that is, a shared attentional mechanism $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \longrightarrow \mathbb{R}$ computes *attention coefficients*

$$c_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j) \tag{1.7}$$

which indicates the importance of node j's features to node i. In order to inject the graph structure into the mechanism, masked attention is performed. That is, c_{ij} is only computed for nodes $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is some neighborhood of node i in the graph. Then, to make coefficients easily comparable across different nodes, they are normalized across all choices of j using the softmax function

$$\alpha_{ij} = \text{softmax}_j(c_{ij}) = \frac{\exp\left(c_{ij}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(c_{ik}\right)}. \tag{1.8}$$

In [269], a is taken as a single-layer feedforward neural network, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, and applying a non-linear activation function, in this case, the LeakyReLU function (with negative input slope $\alpha = 0.2$). Then, the coefficients computed by the attention mechanism may be expressed as

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \parallel \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \parallel \mathbf{W}\vec{h}_k]\right)\right)} \tag{1.9}$$

where $\cdot^T$ represents transposition and $\parallel$ is the concatenation operation.

After obtained, the normalized attention coefficients are used to compute a linear combination of features corresponding to them, to serve as the final output features for every node (after potentially applying a non-linearity, σ)

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right). \tag{1.10}$$

To stabilize the learning process of self-attention, multi-head attention is employed, similarly to the method introduced by Vaswani et al. [267]. Specifically, K independent attention mechanisms execute the transformation of Equation 1.10, and then their features are concatenated, resulting in the following output feature representation

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right). \tag{1.11}$$

where $\parallel$ represents concatenation, α_{ij}^k are normalized attention coefficients computed by the k-th attention mechanism (a^k) and $\mathbf{W}^k$ is the corresponding input linear transformation's weight matrix. So, it can be noted that the final returned output, $\mathbf{h}'$, will consist of KF' features.

Specially, if multi-head attention is applied to the final (prediction) layer of the network, concatenation becomes impractical. Instead, averaging is utilized, and the application of the final non-linearity—typically a softmax or logistic sigmoid for classification tasks—is delayed until this stage.

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right). \tag{1.12}$$

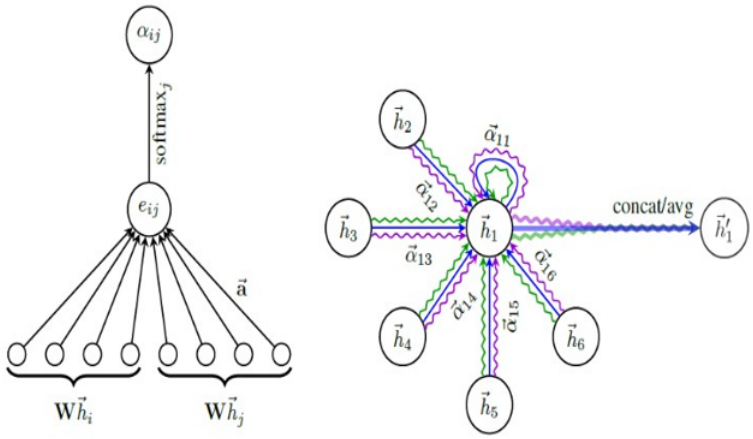


Figure 1.38: Graphical representation of attention mechanism [269]. **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by graph attention layer, defined in Equation 1.7. **Right:** An illustration of multihead attention by node 1 on its neighborhood. Different arrow styles and colors denoted independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_i$ as defined in Equation 1.12.

An illustrative diagram with some (non exhaustive) examples of GNN models depending on the graph structure, prediction task or features used as input data (node and edge features, adjacency matrix) is described in [53].

GNN models were built using the PyTorch Geometric[8](PyG) library from Python, which is based on the PyTorch[9] framework. However, these models could also be constructed using the TensorFlow[10] framework if needed. Other libraries that utilize graph-type structures or have implemented GNN models could also be explored. Additionally, the deep learning model used in Chapter 2 was also built using the PyTorch framework.

1.3.5 GNN-based applications

In this section, some applications derived in biology and using microscopic data are discussed, mainly focusing on dynamics and interactions in network topology and the analysis of microscopic images.

1.3.5.1 Dynamics and interactions in network topology

Biological networks can be effectively represented and analyzed using graph theory [93, 147], which provides valuable insights into the structural organization and relationships within biological data. This approach is particularly useful for understanding complex cell-cell interactions, a task that still remains a significant challenge. By leveraging graph-based methods, researchers can uncover patterns and dynamics that inform the storage, visualization, and analysis of intricate biological systems.

Wang et al. [277] proposed a GNN-based approach to predict the prognosis of gastric cancer patients using Cell-Graph data from multiplexed immunohistochemistry

[8] https://pytorch-geometric.readthedocs.io/en/latest/#
[9] https://pytorch.org/
[10] https://www.tensorflow.org/

images. Cell-graphs [290] are constructed by computing the Euclidean distance between any pairs of cells and fixing a connectivity radius (in pixels) proportional to the maximum effective distance between immune and tumor cells.

Bafna et al. [21] proposed CLARIFY, a tool that takes gene regulatory networks (GRNs) as input and uses them along with spatially resolved gene expression data to infer cell-cell interactions. This model employs a novel multi-level graph autoencoder, which mimics cellular networks at a higher level and cell-specific GRNs at a deeper level.

Wang et al. [274] introduced scGNN (single-cell graph neural network), a hypothesis-free deep learning framework for analyzing scRNA-Seq data. This framework models cell-cell relationships through graph neural networks and captures heterogeneous gene expression patterns using a left-truncated Gaussian mixture model [190]. scGNN combines three iterative multi-modal autoencoders, achieving superior performance in gene imputation and cell clustering on four benchmark scRNA-Seq datasets.

Acharya et al. [3] have proposed a deep neural network model that employs a CNN as the initial feature extractor, followed by a GNN for predicting cell cycle states. A different timestamp is added to the feature vectors, representing this information as a graph to capture internal interactions and predict the subsequent cell state.

Recent graph-based deep learning (DL) models have demonstrated excellent performance by leveraging spatial and/or temporal components within graphs. In this context, there exists AlphaFold [138], a DL approach that predicts the 3D structure of a protein by using its amino acid sequence as input data. In this model, the prediction task is viewed as a graph inference problem in 3D space, where the edges of the graph are defined by residues in proximity.

Wu et al. [284] have developed SPACE-GM, a geometric deep learning (DL) framework that flexibly models tumor microenvironments (TMEs) as cellular graphs. SPACE-GM employs a graph isomorphism network (GIN) [287] as its backbone and multiple multilayer perceptrons (MLPs) [228] as prediction heads. In this model, spatial cellular graphs are constructed from Voronoi polygons [49, 175, 243], where each node represents a cell, and the edges indicate adjacent cells.

Brbić et al. [44] have developed STELLAR, a geometric deep learning (DL) method for cell-type discovery and identification in spatially resolved single-cell datasets. STELLAR automatically assigns cells to cell types present in the annotated reference dataset and discovers novel cell types and cell states. In this model, the graph is constructed using the spatial proximity of cells, with molecular features of cells serving as node features.

Pineda et al. [213] have implemented MAGIK, a geometric deep learning approach that accurately estimates dynamical properties in various biologically relevant scenarios, such as cell division by considering temporal edges within a fixed time window. This deep-learning method relies on a graph neural network enhanced by attention-based components. By processing object features with geometric priors, the network is capable of performing multiple tasks, including node regression and classification, link prediction, and global dynamic properties prediction.

1.3.5.2 Cell image analysis

In cellular image analysis, two current and popular tasks are cell segmentation and tracking. However, these tasks remain challenging due to handling large datasets, tracking cells in dense and overlapping populations, and addressing motion blur or photobleaching during imaging.

Researchers have recently developed several high-performing graph-based models for these two tasks. Hajdowska et al. [103] propose a graph-based algorithm called GRABaCELL to improve the automatic segmentation and counting of living cells from fluorescence microscope images. This approach is based on a graph cuts optimization method [40].

Chowdhury et al. [62] model the problem of cell tracking over pairs of video microscopy image frames as a minimum weight matching problem in bipartite graphs [74, 20]. The bipartite matching essentially establishes one-to-one correspondences between cells in different frames.

Ben-Haim and Riklin-Raviv [28] have introduced a novel GNN approach for cell tracking in high-throughput microscopy videos for 2D and 3D datasets of different cell types. This approach models the entire time-lapse sequence as a directed graph, where cell instances are represented by its nodes and their associations by its edges. This new GNN block type enables a mutual update of node and edge feature vectors, thus facilitating the underlying message-passing process.

On the other hand, MAGIK [213] also offers solutions to the tracking problem in two distinct ways. First, this model can be used to derive the trajectories of observed objects based on the structure of their spatial and temporal connections. Second, MAGIK enables the characterization of the motion of imaged objects through their spatio-temporal connection graph, thereby bypassing the need for explicit tracking.

Zhao et al. [301] have proposed a novel deep learning (DL) approach featuring graph-based tracking for cell segmentation and tracking in microscopy images. To enable comprehensive cell detection and instance segmentation, they combine DeepLabv3+ [59] for semantic segmentation and ResNet50 [108] for enhanced feature extraction. The relative positions of graph nodes are used to track segmented cells, encompassing cell division and apoptosis.

Chapter 2

Characterization of cellular tissues

This chapter describes the first application analyzed in this thesis. The main idea of this application is to characterize network-type cellular tissues as graphs and then build features that help understand and distinguish the formation process of these structures under different initial conditions.

This work represents an interdisciplinary collaboration with Maxim Balakirev, a Research Scientist at the Large-Scale Biology Research Laboratory (BGE) from CEA. One of his current work involves the study of a particular phenomenon, which shows the transformation of cells aggregation into tubular network structures to develop organoids. The behavior of this network formation has been observed in manipulating epithelial cells (Figure 2.1), which are specialized cells type that form a physical barrier by lining the cavities of the major organs (the skin, lungs, gastrointestinal tract, and genitourinary system) separating them from one another and the external environment [129].

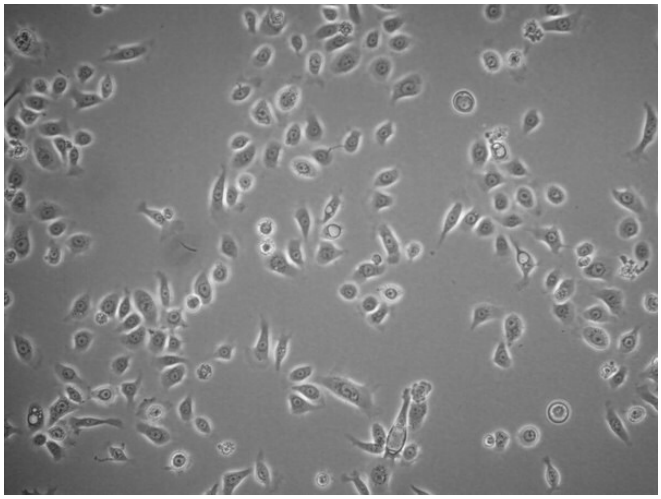


Figure 2.1: Bright-field image of Human Pancreatic Duct Epithelial Cells. Source: https://www.sigmaaldrich.com/FR/en/product/mm/scc442

Besides, this epithelial cells-based network formation immediately rises the following question: how is this process derived? Cell-cell interactions, inhibited by the utilization of drugs on cells, and cell-matrix interactions, produced on the extracellular matrix (ECM), can lead to a possible explanation to this question. The ECM (Figure 2.2) is an assembly of macromolecules (of protein and carbohydrate nature) that bind together homologous or heterologous cells and organize them into tissues [134]. There exist biological models that can explain this cellular network formation,

which are possibly based on the knowledge from epithelial-angiogenesis studies [87, 105]. Angiogenesis is the process of new blood vessels growth from pre-existing ones [236].

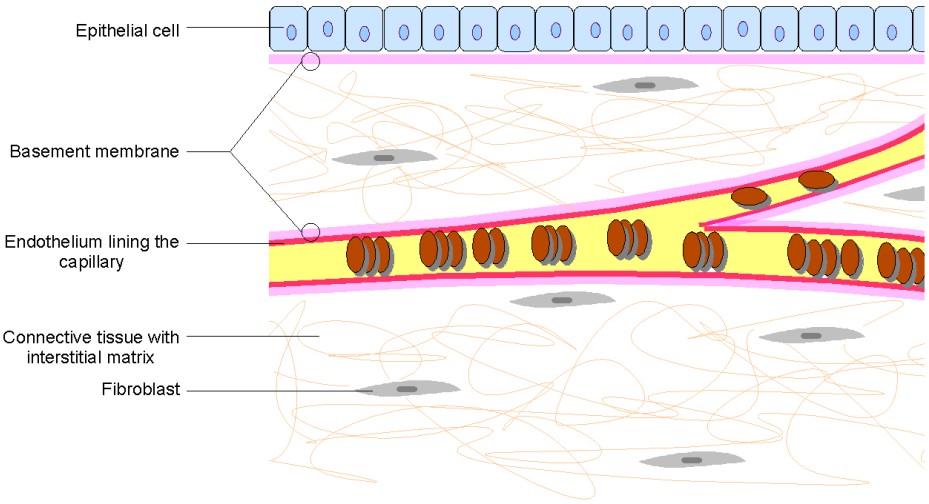


Figure 2.2: Illustration depicting extracellular matrix (basement membrane and interstitial matrix) in relation to epithelium, endothelium and connective tissue

This cellular network formation phenomenon can potentially become an important area of study in biology because its overall analysis can conduct to the study of diseases (in particular, pancreatic diseases) and the potential development of treatments. The capability of analyzing cell dynamics that lead to this process and quantifying observed structural changes are crucial for the comprehension of this type of cellular behavior. However, capturing and representing the complex connectivity of networks from two-dimensional (2D) microscopic images of cells remains a challenging task. Lens-free imaging is a useful technique that allows for the visualization of biological objects. However, it also requires a well-performing analysis of the images acquired, which is also challenging. This can be observed when trying to segment cell structures-based images through current image-processing methods.

Figure 2.3 illustrates the comparison between two segmentation methods: ERnet (a recent deep learning method for the semantic segmentation) [180], and Otsu's algorithm [204] at 3 time-steps for a video of 2D epithelial cells growth images. In this example, it is observed that both methods generate good results at $t = 17h20min$. However, ERnet performs better at $t = 43h$, whereas Otsu works better at $t = 70h40min$ (see red rectangles). Besides, it could be necessary to complement these approaches with another image techniques as for example some morphological operations like opening, closing, erosion, dilation, etc. Therefore, there is no unique segmentation method for handling these images. An intuitive idea to deal with this problematic is to define a deep learning-based model that encapsulates all these methods together, which will be seen later.

Additionally, individual cells or cell aggregates can exhibit dynamic behaviors over time, which could need to be quantified for subsequent analysis and interpretation. These cells or cell aggregates can form cellular networks, which can be represented as mathematical graphs, with nodes representing cell structures and edges corresponding to their connections [88]. These mathematical objects can allow for the quantification of metrics derived from their topological structures, potentially explaining the biological changes observed in the behavior of the cells. This process involves

transforming segmented images into graphs that represent the corresponding network structures. This can be achieved through the computation of the skeleton shape from the segmented images using image processing tools.

A workflow (Figure 2.6) is therefore proposed that enables the description of network connectivity from processed images, based on existing models and tools. Images are segmented, skeletonized, and then represented by graphs to quantify suitable and efficient metrics associated to the network connectivity. Three datasets of dynamic cell networks formed by pancreatic epithelial cells, which exhibit particular biological behaviors, are studied. Graph metrics were identified to distinguish between different experimental conditions.

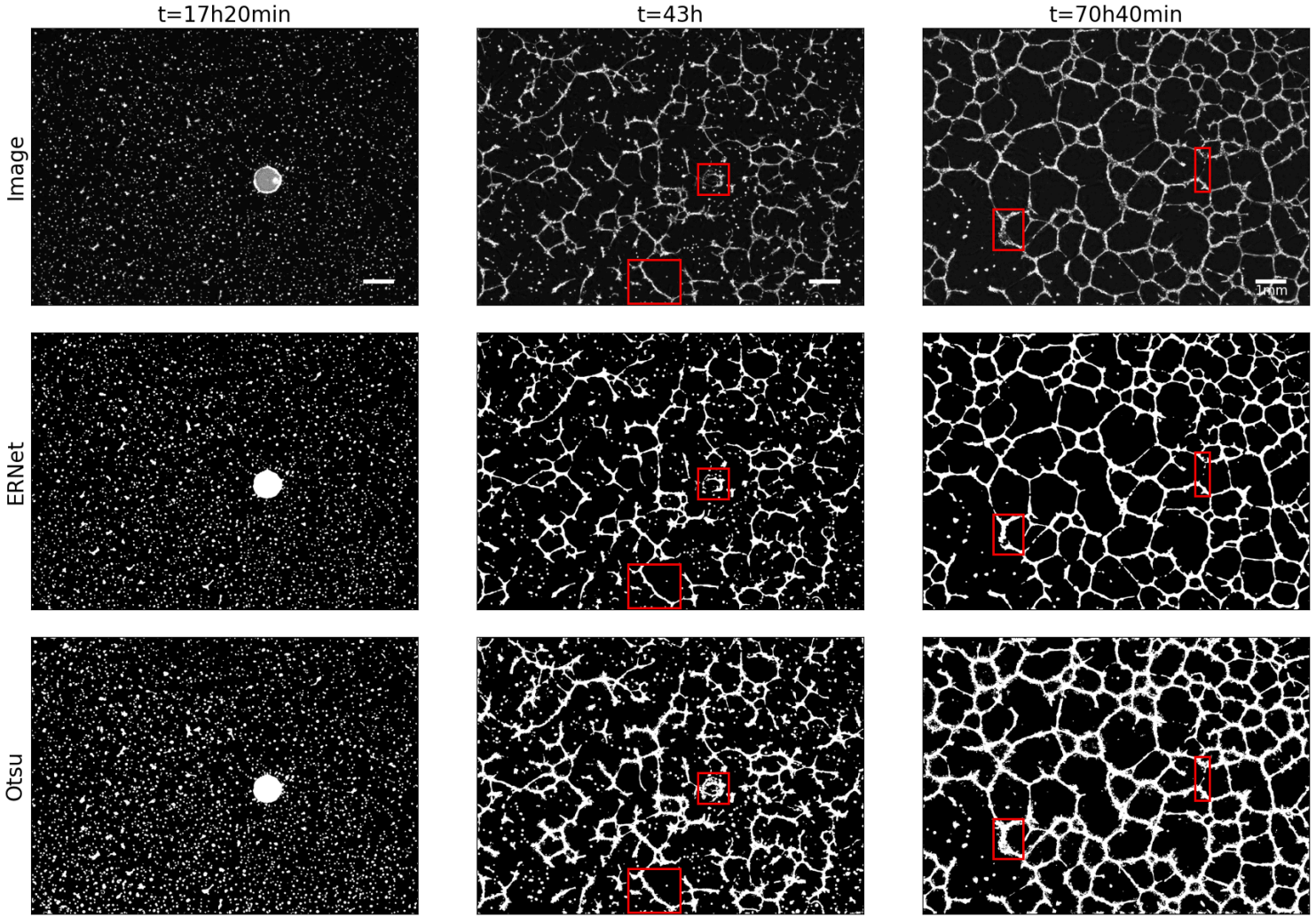


Figure 2.3: Comparison between ERnet and Otsu methods at 3 frames for a video of 2D epithelial cells growth images. Red rectangles show differences between these two methods.

2.1 Research context

Tissue formation follows a tightly regulated morphogenetic program that involves iterative interactions between cells and cell assemblies (or clusters) guiding cellular differentiation and patterning over a large scale. This poorly characterized emergent process allows single cells to self-organize into complex interconnected macro-scale structures [237].

The assembly of cells into networks plays a crucial role in tissue morphogenesis across various scales: from the initial establishment of cell-cell contacts to the final arrangement of multicellular elements within the tissue structure. For instance, in vasculogenesis, endothelial cells initially form single-cell networks that subsequently evolve into complex, branched hierarchical networks of multicellular vessels [37]. This interconnected network formation is not limited to circulatory, lymphatic, and nervous systems; it is also vital for the branching morphogenesis of glandular epithelium in organs such as the breast, lung, stomach, colon, pancreas, prostate, uterus, and cervix [107]. Dysfunctions in tissue morphogenesis are implicated in numerous pathologies, including cancer. Hence, studying cell network formation is crucial for understanding both the mechanisms of natural tissue morphogenesis and the development of human diseases.

Cell network formation is being unraveled through biological models based on knowledge from epithelial-angiogenesis studies, as mentioned previously. The process of network formation could be summarized through two types of interactions: cell-cell interactions and cell-matrix interactions. Cell-cell interactions are modulated by a drug that alters the speed of cell communication, motility, and migration, depending on the concentration administrated. Cell-matrix interactions enable cells spreading themselves. This occur when cells interact through various molecular signals and biomechanical forces mediated by the ECM stiffness[1]. Figure 2.4 illustrates an overview for uniform cell spreading on soft versus stiff substrates based in [223]. Cells (gray) extend protrusions (cyan) to test the stiffness of their substrate. On soft substrates, forces develop slowly, preventing strong adhesions, and the cell retracts its protrusion. On stiff substrates, forces build up quickly, stabilizing focal adhesions and promoting cell adhesion. This cycle repeats, enabling the cell to spread maximally based on the substrate's rigidity [223]. In summary, by controlling the parameters of drug's concentration and matrix stiffness, it is possible to speed up or slow down the network formation.

To comprehend the impact of the factors involved through cell-cell interactions and cell-matrix interactions on the dynamics of self-organization processes and the final structure outcomes, it is necessary to provide quantitative and temporal analysis of the cell network formation.

To achieve this, suitable imaging tools need to be utilized to observe these network structures to the greatest extent possible. Lens-free microscopy is a recent technique capable of observing these cellular networks over a wide field of view. Images acquired through this microscopic technique should be analyzed using segmentation methods to ensure explicit identification of all cells and their respective structures within the cellular network. As previously discussed at the beginning of this chapter, current segmentation models are not sufficiently effective to handle the dataset presented

[1] **Stiffness** is the extent to which an object resists deformation in response to an applied force [250].

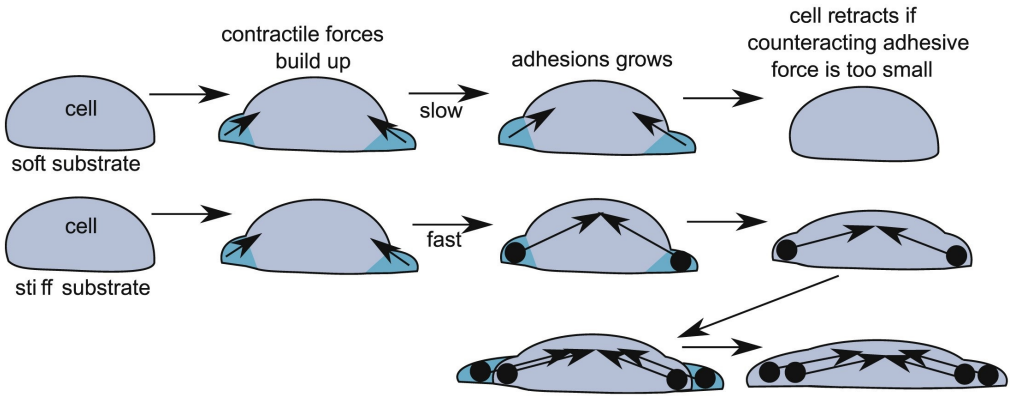


Figure 2.4: A diagram explaining cells spreading based on the stiffness of their substrate. When a cell is on a soft substrate (top), it forms protrusions that may attach to the ECM but often retract due to insufficient adhesive force. In contrast, on a stiff substrate (bottom), the cell builds up force more rapidly. This allows the cell to stick and spread further on the stiff substrate [223].

in this study, which consists of videos of pancreatic epithelial cells growth within an ECM (hydrogel). This can be complemented by the fact that the appearance of the images through the time-lapse varies considerably, making it difficult to have a pipeline that works consistently from the beginning to the end of the video. To address this issue, a new segmentation method should be proposed to manage this type of data.

Individual cells or cell aggregates form cellular networks, which can display dynamic behaviors over time, requiring quantification for further analysis and interpretation. These cellular networks can be depicted as mathematical graphs, with nodes representing cellular components and edges denoting cellular branches [51]. Consequently, graphs are utilized to efficiently capture the connectivity of the cellular network derived from segmented images (see Figure 2.5). By using graph theory, features are constructed based on the network, and metrics are quantified for network analysis.

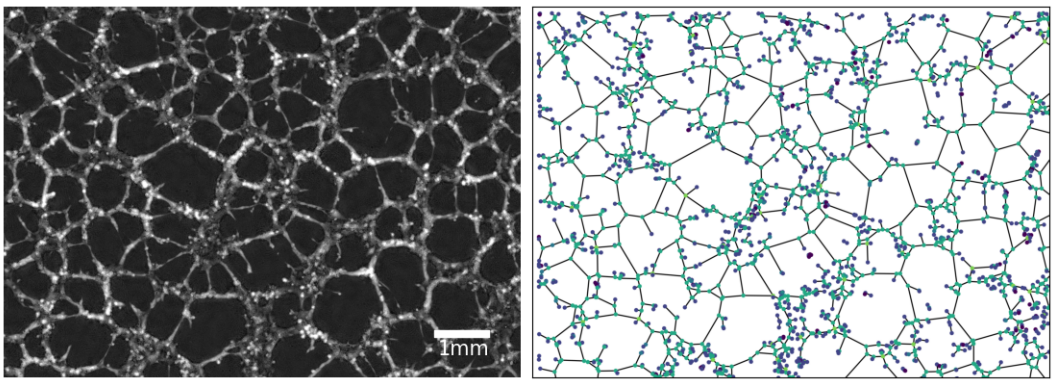


Figure 2.5: Representation of a cellular tissue network as a graph. Left: acquired input image. Right: graph representation derived from the acquired image. It is clearly observed that nodes represent cell assemblies (or clusters) and edges correspond to cellular branches.

Therefore, this work primarily focuses on two tasks: developing a new segmentation process for image data and identifying (or constructing) graph metrics to

characterize dynamics in cellular networks.

2.2 Literature review

Image processing-based methods and deep learning-based models have been developed for segmenting and analyzing cellular network structures [50, 82, 115, 180, 181, 207]. Most of these methods allows for analyzing network dynamics through graph-based metrics. GNEO (Geometric and Network Representation of Epithelial Organization) [82] was one of the first methods introduced to analyze epithelial network structures via geometric and network-based metrics. A semi-automated framework is applied on processed images to segment them, producing an epithelial network where each node corresponds to one of the cells and the links between nodes reflect the spatial adjacency between the epithelial cells [82]. This representation enables the quantification of node degree, clustering coefficient, and average degree of neighbors as network metrics, as well as the cells' area as a geometric metric. By utilizing these metrics, GNEO effectively identifies unique signatures that differentiate epithelial tissues across various organs, species, developmental stages, and genetic conditions [82].

EpiTools is an open-source image analysis toolkit designed to accurately segment membrane-labeled cells in epithelial networks and quantify their growth dynamics [115]. This method defines an image-processing pipeline to preprocess the image prior to segmentation and to segment (using a selective plan projection, an automatic seed generation, and a watershed method) the image [115]. A skeleton is generated from the binarize image, which defines a cell graph representation, where nodes represent cell membranes and edges correspond to their adjacent connections, providing a semantic description of the tissue of interest [115]. This cell graph representation allows for the calculation of geometric metrics, including area and polygon class, for the analysis of tissue dynamics. However, it does not take into account graph metrics for this analysis.

AnalyzER is a software tool designed to automatically analyze plant endoplasmic reticulum (ER) structures (tubules and cisternae) from multi-dimensional fluorescence images [207]. Segmentation of ER images identifies a single-pixel skeleton of tubular elements using hysteresis thresholding and thinning, cisternal structures through opening and active contour refinement, and enclosed polygonal regions within the ER network [207]. The single-pixel skeleton is converted to a graph representation with nodes at junctions, including puncta, and free ends, connected by edges along each tubule [165, 207]. The topological organization of the ER network is quantified using graph-theoretic metrics, including node density, cisternal density, average degree, global efficiency, meshedness[2]. betweenness, and centrality.[207].

Previous methods include the analysis of network structures, but they could struggle to handle network-type tissues that contain cell clusters or complex structures. The Angiogenesis Analyzer [50] is another tool developed to explore cell networks and their connectivity formed by endothelial cells in *in vitro* assays that mimic angiogenesis. The segmentation process proposed by this approach can effectively handle cell clusters generated in cellular networks. However, it shows poor performance when the network is not yet formed such as at the beginning of a time-lapse video. By modeling the analyzed structure at different levels, this tool can obtain basic

[2]**Meshedness coefficient** represents the fraction between the current number of loops and the maximum number of loops in a planar graph [206].

hierarchical data in the form of series of vectorial objects, such as nodes, junctions, extremities, segments, branches, or meshes [50]. This approach does not define an explicit graph representation, which would allow for the incorporation of graph-based metrics in the analysis of the angiogenesis process. This issue could be resolved by developing a pipeline that converts the vectorial objects into a graph representation to compute these metrics.

Lu et al. [181] introduced ERnet, a recent deep residual network designed for segmenting ER microscopy images. In subsequent work [180], they developed ERnet-v2, which utilizes a Swin Transformer-based model [171] to identify tubules, sheets, and sheet-based tubules (SBTs) within ER structures. The segmentation of this method identifies the entire ER structure, which is subsequently skeletonized to obtain a graph representation. In this representation, tubule junctions are represented by nodes, and tubules are represented by edges [180]. Quantitative insights into the topology of the ER network are provided through the computation of the following graph metrics: assortativity, clustering coefficient, number of connected components, node- and edge-assembly ratio, and node degree. ERnet performs well with early-stage network-type structures but struggles to segment networks based on cellular clusters. A possible solution to this problem is to combine ERnet segmentation with some morphological operations to improve the generated mask.

2.3 Methodology

Figure 2.6 shows a global overview and summary of the proposed workflow for this application. The workflow can be divided into two main steps, which are segmentation and graph conversion with metrics extraction. The segmentation process involves a two-phase method: initial and automatic segmentation. Initial segmentation of the acquired images was performed using existing image processing tools and current models. This initial segmentation was then used to train a deep-learning model, which was subsequently employed to automatically re-segment the images. A skeleton was extracted from the segmented data, and nodes and edges were identified to create a connectivity graph. Graph theory techniques were applied to extract features from the graphs, enabling the quantification of metrics that reflect dynamic network changes.

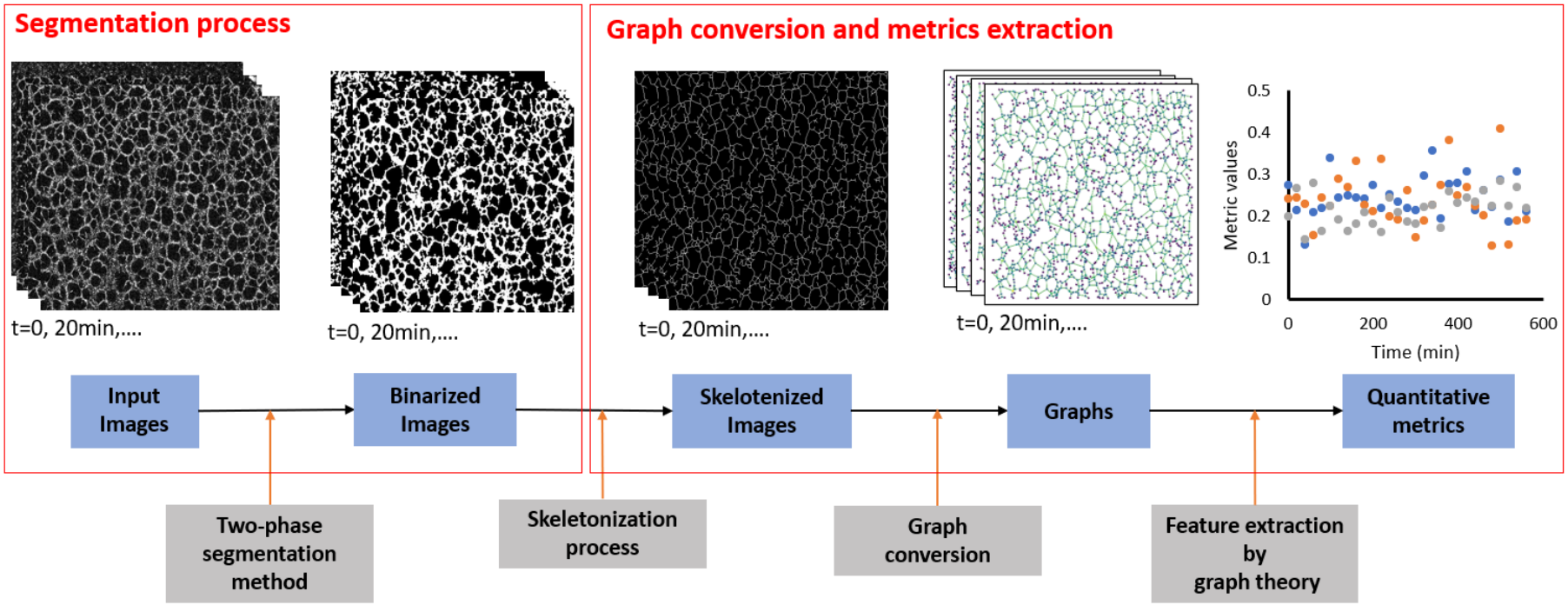


Figure 2.6: Two-step proposed workflow consists of segmentation process and graph conversion & metrics extraction. The segmentation process involves a two-phase segmentation method (initial and automatic) to binarize images. A skeleton was derived from the segmented network by identifying nodes and edges to construct the connectivity graph. Graph theory was then applied to extract features from the graphs, quantifying metrics that capture dynamic network changes.

2.3.1 Dataset

For this study, datasets comprising 2D time-lapse videos of pancreatic epithelial cell growth within an ECM (hydrogel) were used. These sequences of images capture the cell behavior under different experimental conditions known to impact cell network formation. For each video, images were captured at intervals of 20 minutes. Three distinct datasets were considered, where cell network formation was modulated by the utilization of drugs or by modifying the stiffness of the ECM. This modulation either speed up or slow down the process, depending on the concentration of the drugs or the force applied (stress) to the ECM.

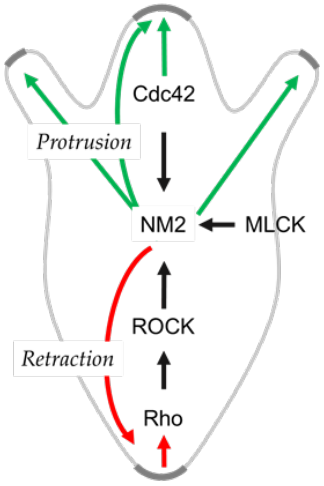


Figure 2.7: Behavior of cells mediated by inhibitors acting on the actomyosin. ROCK and MCKL promote actin-myosin contractility for cell retraction, while Cdc-42 primarily induces protrusion formation. NM2 is important for the retraction phase and collaborates with ROCK, MCKL, and Cdc-42 to regulate cell motility.

Figure 2.7 illustrates the behavior of cells mediated by inhibitors acting on the actomyosin, which refers to the actin[3]-myosin[4] complex that forms within the cells cytoskeleton[5] [1]. The inhibitors considered for this study include Rho Associated Coiled-Coil Containing Protein Kinase (ROCK), Myosin Light Chain Kinase (MLCK), Nonmuscle myosin-2 (NM2), and Cell division cycle 42 (Cdc-42). These inhibitors control morphological changes in cells, such as protrusion, a forward movement, and retraction, a movement that pulls back [245].

ROCK, an effector of the small GTPase Rho [16], promotes actin-myosin contractility, which is essential for cell movement and the retraction of the cell body during migration [4, 224, 281]. MLCK is also involved in the regulation of actin-myosin contractility [143], similar to ROCK.

Cdc-42, a GTP-binding protein of the Rho family [52], plays a central role in regulating the actin cytoskeleton and maintaining cell polarity[6] [246]. It is strongly associated with the formation of cellular protrusions [246], particularly filopodia[7],

[3]**Actin** is an abundant 43-kd protein that polymerizes to form cytoskeletal filaments [66].

[4]**Moysin** is a protein that interacts with actin as a molecular motor [66].

[5]**Cytoskeleton** is a highly dynamic system comprising a network of actin polymers and actin binding proteins [268].

[6]**Cell polarity** refers to the intrinsic asymmetry observed in cells, either in their shape, structure, or organization of cellular components [2].

[7]**Filopodia** are actin-based finger-like protrusions extending from the cell surface [52].

and lamellipodia[8] [52, 149, 151, 215].

NM2 is essential for the retraction phase of cell movement and collaborates with other proteins, such as ROCK, MLCK, and Cdc-42, to regulate the complex processes involved in cell motility [45, 124, 272, 205, 135].

For the first dataset, cell network formation was modulated by drug Y27632 that inhibits ROCK protein, denoted as ROCKi. Videos are referred to as condition 1, 2, and 3, corresponding to ROCKi concentrations of 0, 1, and 5 μM, respectively. The different ROCKi concentrations resulted in different connectivity behavior, and led to distinct evolving network structures over time. This dataset is denoted as ROCKi data. Figure 2.8 illustrates three time instants of the video for each condition in the ROCKi dataset. It is apparent that, in Condition 1, network structures do not develop as no drug is administered to the cells.

The second dataset, denoted as ECMstiff data, consists of six 2D time-lapse videos of the same epithelial cells, in which network formation was induced by changes in the stiffness of the ECM. Videos are referred to as condition 1, 2, 3, 4, 5 and 6, corresponding to stiffness values of 2500, 1070, 786.35, 713.8, 103.4, and 39.135 Pa, respectively. Figure 2.9 shows three time points from the video for each condition in this set. It is observed that the last two conditions (5 and 6), which have the lowest stiffness values, exhibit no network development, while Condition 1, with the highest stiffness value, shows accelerated network formation

The last dataset, denoted as Actomyosin data, consists of eight 2D time-lapse videos of the same epithelial cells under different initial conditions, similar to those in previous experiments. These videos are subdivided into two sets: Actomyosin 1, which consists of three 2D time-lapse videos, and Actomyosin 2, which includes the remaining five 2D time-lapse videos. For the Actomyosin 1st set, the videos are referred to as conditions 1, 2, and 3. Three distinct inhibitors were applied to the cells for these conditions: Cdc-42 at a concentration of 1 μM, ROCK at 5 μM, and at 1 μM, respectively. Figure 2.10 displays three time instants of the video for each condition in this set. At $t = 71h$, it is evident that Conditions 1 and 3 do not form a network structure, in contrast to Condition 2.

For the Actomyosin 2nd set, the videos are referred to as conditions 1, 2, 3, 4, and 5. Conditions 1 and 4 are controls, meaning no drug was used on the cells. For the remaining conditions 2, 3, and 5, three different inhibitors were applied to the cells. These inhibitors were ROCK at a concentration of 5 μM, NM2 at 1 μM, and Cdc-42 at 5 μM, respectively. Figure 2.11 presents three time instants from the video for each condition in this set. It is observed that Conditions 1 and 4 do not develop a network structure at $t = 71h$. This is also noticed in Condition 5, even though an inhibitor was used in this case.

[8] **Lamellipodia** are defined as flat broad membranous protrusions located at the leading edge of the migrating cells [239].

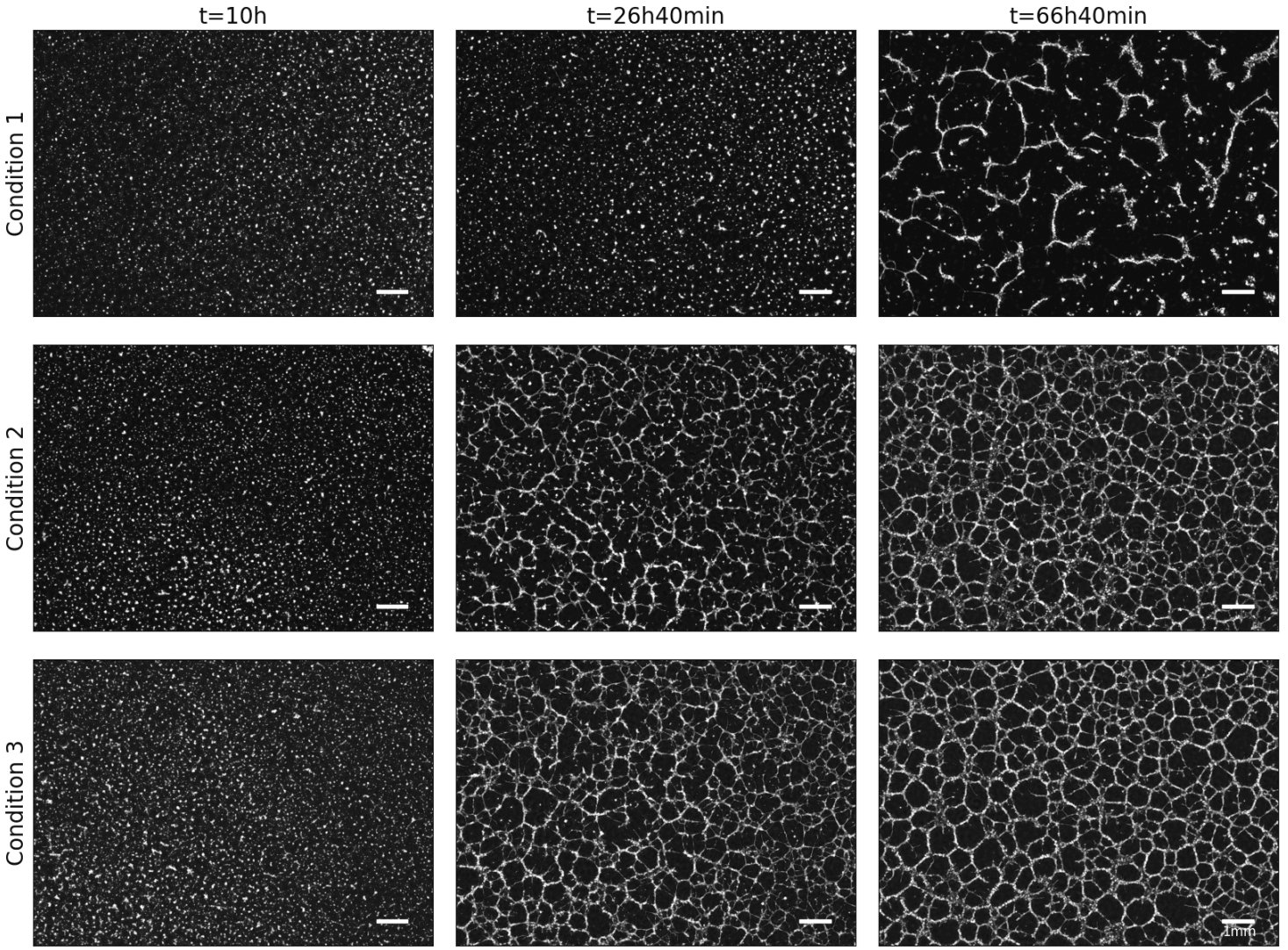


Figure 2.8: Evolving cellular network representation for all conditions at 3 frames of the time-lapse video

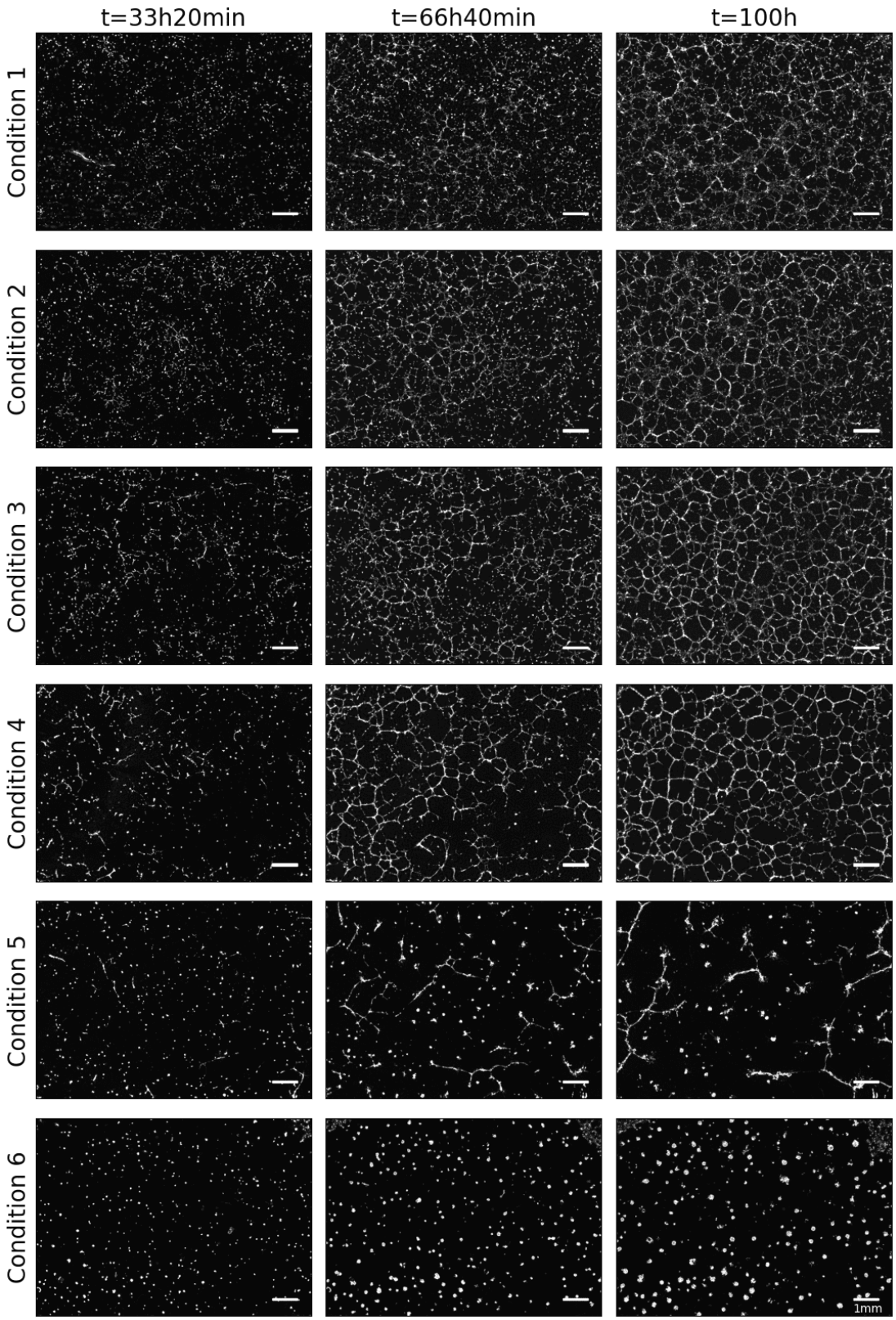


Figure 2.9: Evolving cellular network representation for corresponding to ECMstiff series at 3 frames of the time-lapse video

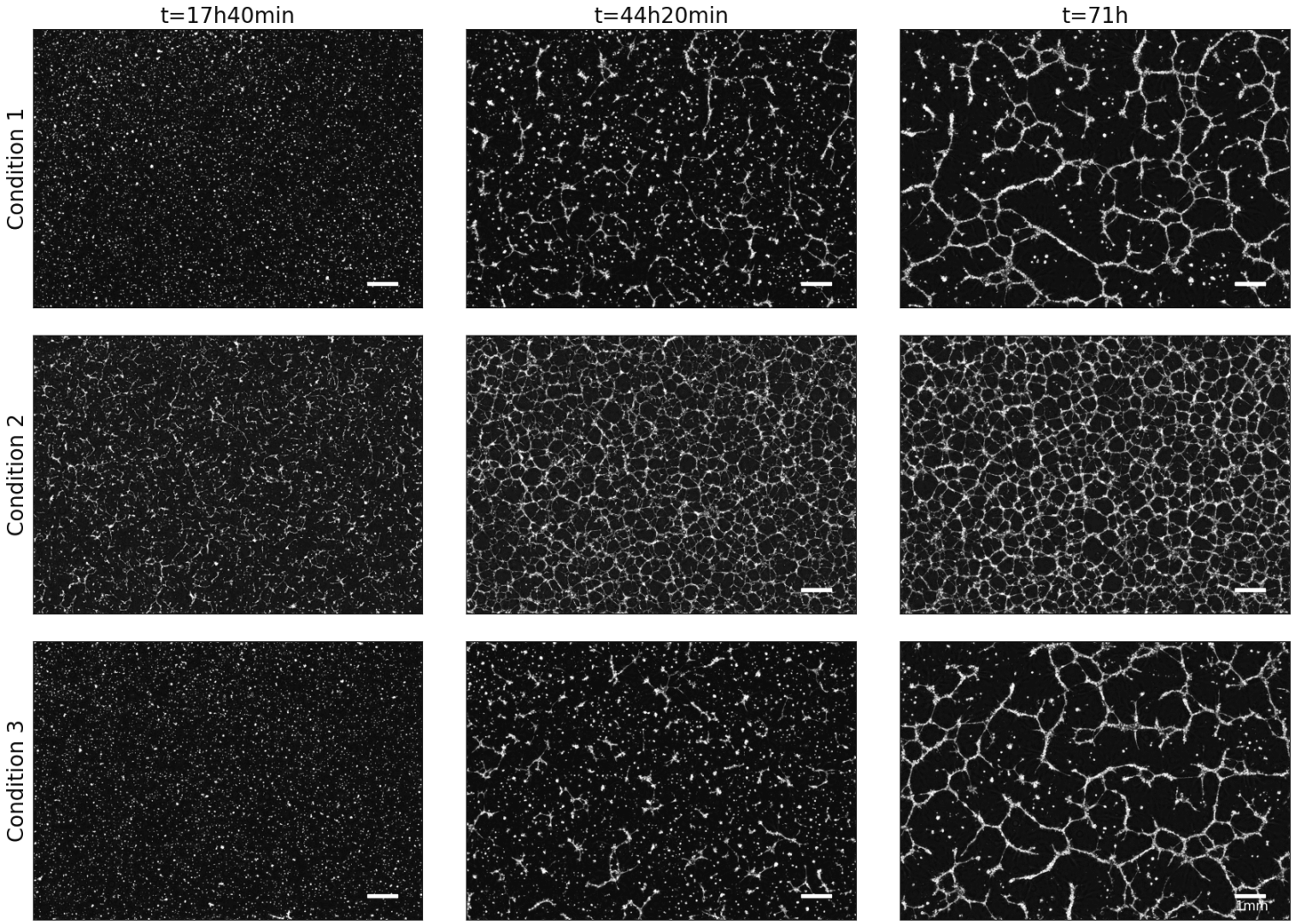


Figure 2.10: Evolving cellular network representation for corresponding to Actomyosin 1st set at 3 frames of the time-lapse video

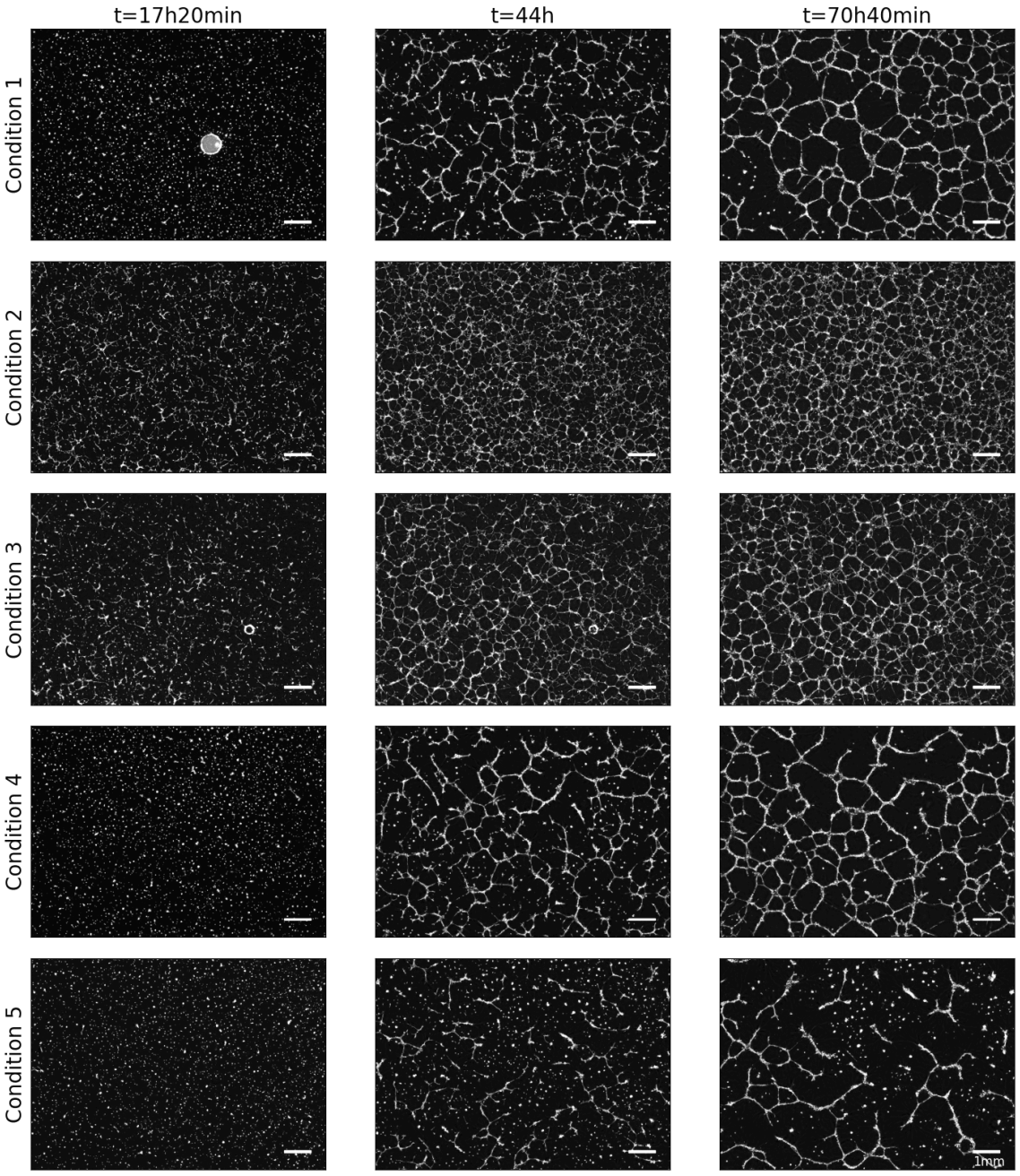


Figure 2.11: Evolving cellular network representation for corresponding to Actomyosin 2nd set at 4 frames of the time-lapse video

2.3.2 Segmentation process

Images were segmented by following the workflow proposed in Figure 2.12. The proposed segmentation method results by combining two phases: an initial and an automatic segmentation. Initial segmentation is made by combining existing tools, such as downsampling, filtering, upsampling, threshold methods, pretrained models (ERnet), and morphological operations. Additionally, the segmented images exhibited abrupt discontinuities in the trend when computing graph metrics (see Figure 2.14). Consequently, a global automatic segmentation method was developed based on the initial segmentation.

Automatic segmentation is deployed by a U-Net [227] architecture, which is trained

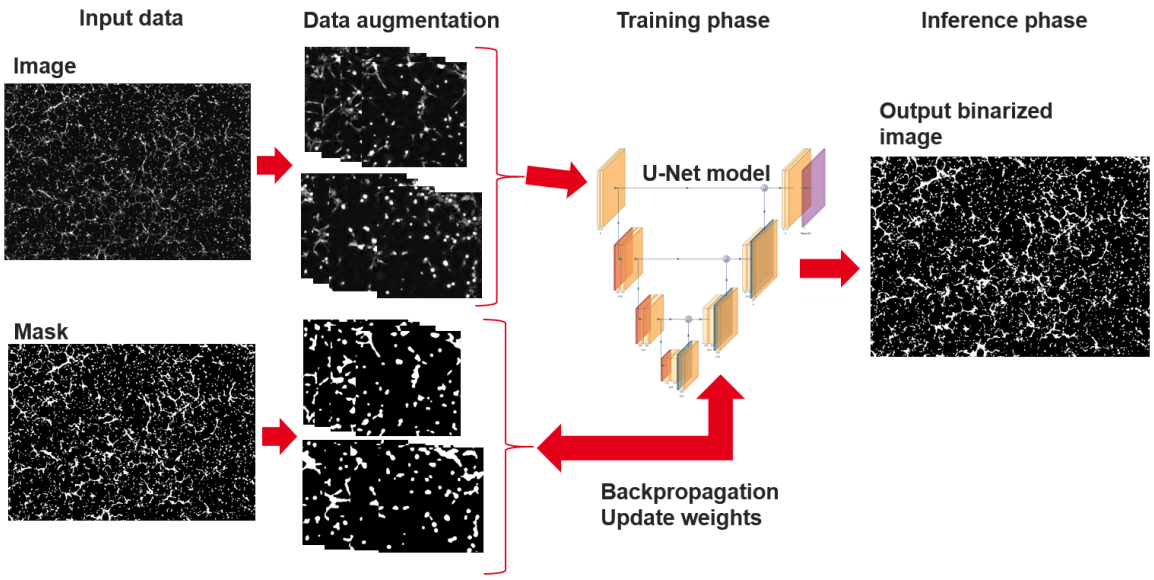


Figure 2.12: Segmentation process. Firstly, image were segmented through image processing tools and recent deep learning models. These first masked images were used as ground truth data to train a U-Net model. Cropping input and mask images is employed for data augmentation during training phase. The final segmentation is obtained by utilizing the U-Net trained model on the input images data.

in a supervised manner with ground truth data based on preliminary segmented images. Data augmentation was performed using cropped images of 512×512 pixels during training phase.

Further details about the different methods for initial segmentation and the hyperparameters used in the U-Net model for automatic segmentation can be found in Annex 6.1.

2.3.3 Graph conversion

Figure 2.13 illustrates the process for graph conversion and quantifying metrics related to the graph structure. The network is represented through an undirected and unweighted graph $G = (V, E)$ with V being the set of nodes and E representing the set of edges. Segmented images were skeletonized using a fast parallel thinning algorithm [297]. Then, graphs were built from the skeleton where nodes and edges were identified as key topological components to represent the network connectivity. Graph theory is employed to extract features from graphs to calculate metrics related to the network structure.

As it will be observed in the output graphs obtained from binarized images in Section 2.4, there could be many nodes on the graph, which is derived from the skeleton. This could introduce a considerable noise into the curves of the different metrics plotted. To address this, it is necessary to simplify the graph as much as possible through a pruning process to obtain its simplest topological structure. Inspired by the work on topological graph simplification solutions to the street intersection miscount problem [38] and the idea of angle analysis [271] to preserve the mesh structures formed on the graph, a pruning algorithm is proposed below. This method begins by identifying nodes with a degree lower than 2 in the graph and then removing them. This reduces the number of nodes with few connections in the graph. Next, to simplify the graph further, nodes with a degree of 2, along with their

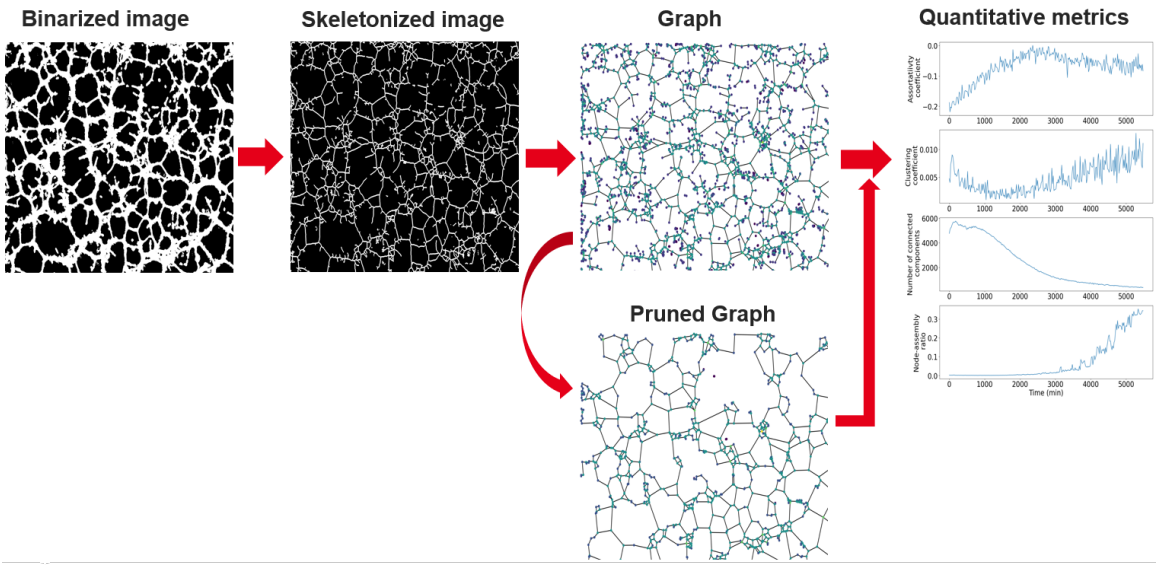


Figure 2.13: Graph conversion process. Segmented images were skeletonized using a fast parallel thinning algorithm [297]. Then, graphs were built from the skeleton where nodes and edges were identified as key topological components to represent the network connectivity. If necessary, graphs can be simplified through a pruning algorithm 1. Features are extracted from (simplified or original) graphs through graph theory to compute network metrics.

neighbors, are identified. This step aims to reduce the proximity between closely connected nodes by removing the intermediary node and connecting its neighbors directly. However, this procedure can result in the loss of the principal network structure, which is desirable to maintain. To avoid this, the angle θ between the node to be deleted and its neighbors connected by edges is computed based on their spatial positions. The criterion for maintaining or removing nodes based on the angle θ aligns with that described in [271]. If θ falls within the open interval $(30°, 150°)$, the node is retained in the graph; otherwise, it is removed. Finally, any isolated nodes are removed from the graph if they still exist.

2.3.4 Graph metrics

Quantitative metrics of the network are measured through the functions available in the Python package NetworkX [102]. For a graph $G = (V, E)$, the number of nodes and edges are denoted by $|V|$ and $|E|$, respectively.

To quantify the structural characteristics of the networks, the global clustering coefficient and the assortativity coefficient, previously defined in Section 1.3.2.3, can be computed. For considering macroscopic network arrangements, the number of connected components in the network is computed. Networks may be entirely connected or composed of many distinct components [5]. These network components can describe merging or splitting behavior in evolving networks over time. Moreover, in networks containing numerous components, the largest component typically represents the most characteristic topological features [252]. To take into account the connectivity behavior, two additional metrics related to the largest connected component Gcc_0 of the network are included: node- and edge-assembly ratios. The **node-assembly ratio** is defined in (2.1) where $|V_{Gcc_0}|$ is the number of nodes in Gcc_0. Equation (2.2) describes the **edge-assembly ratio** with $|E_{Gcc_0}|$ as the number

Algorithm 1: Pruning method

Data: Graph $G = (V, E)$, nodes position $u_{x,y} = (u_x, u_y)$ $\forall u \in V$
Result: Simplified graph G'
Create a copy of the graph G, denoted as G';
while G' *has nodes of degree* < 2 **do**
  Identify the nodes in graph G' that have a degree lower than 2 and save them in a list;
  Remove each of these nodes from graph G';
end
Identify the nodes in simplified graph G' that have a degree of 2 and save them in a list: list_nodes_deg_2;
for *i in list_nodes_deg_2* **do**
  Identify the neighbors of i: u and v;
  Compute the angle between edges $e_{i,u}$,$e_{i,v}$: $\theta_{i,u,v}$;
  if $\theta_{i,u,v} \notin (30°, 150°)$ **then**
    Remove the edges $e_{i,u}$, $e_{i,v}$ from G';
    Add the new edge $e_{u,v}$ to G';
    Remove node i from G';
  end
end
if G' *has isolated nodes* **then**
  Identify the isolated nodes (nodes with degree 0) in graph G' and save them in a list: list_nodes_deg_0;
  for *j in list_nodes_deg_0* **do**
    Remove node j from G';
  end
end

of edges in Gcc_0. These metrics range from 0 (completely disconnected) to 1 (fully connected).

$$\text{Node assembly-ratio} = \frac{|V_{Gcc_0}|}{|V|} \tag{2.1}$$

$$\text{Edge assembly-aratio} = \frac{|E_{Gcc_0}|}{|E|} \tag{2.2}$$

In addition to these metrics, those previously defined in Section 1.3.2.3 are also considered. These include global efficiency, local efficiency, density, and the number of bridges. Furthermore, the following metrics derived from the node degree are included:

- **Number of junctions:** It is the number of nodes with degree higher than 2 on the graph.
- **Number of extremities:** It is the number of nodes with degree 1 on the graph.
- **Number of isolated nodes:** It is the number of nodes with degree 0 on the graph.

- **Number of nodes per degree:** This represents the total number of nodes per degree in the network. This metric is computed for nodes with degree 0 (isolated nodes), 1 (extremities), 2, 3, 4, 5, and 6 or higher.
- **Percentage of nodes per degree:** This represents the percentage of nodes per degree with respect to the total nodes in the network. This metric is computed for nodes with degree 0 (isolated nodes), 1 (extremities), 2, 3, 4, 5, and 6 or higher.

Global and local efficiency measures can be used to examine the higher-order efficiency of transport across multicellular networks [88]. Local efficiency assesses how the random failure of nodes affects the number of shortest paths between nearby node pairs, providing insight into the network's resilience to random errors by evaluating the increased local transportation costs [88]. Global efficiency, on the other hand, measures the overall network's routing efficiency by considering the shortest paths between all node pairs and determining the ease with which information can traverse the network [88].

Network density indicates the overall level of connectivity among nodes (cell clusters). A high value of density represents that the network is more dense and nodes are more cohesive [31]. On the other hand, a low density value represents a less connected (sparse) network. In a dense network, the information can flow easily and faster than in a sparse network [31].

The number of bridges can allow the identification of critical connections that maintain the network's integrity [196]. A high number of bridges suggests that the network has several key connections that, if removed, could disrupt the overall connectivity [35]. This could indicate vulnerabilities or important pathways in the cellular network [196].

The number of junctions can also be related to the network connectivity. A high number of junctions could indicate that the network is highly interconnected. Few junctions might indicate a less connected network, which could be more vulnerable to disruptions.

The total number of nodes per degree can reveal how the connectivity of the network is evolving. An increase in high-degree nodes might indicate a growing centralization, while an increase in low-degree nodes might suggest a more decentralized or fragmented network. This behavior can also be observed in the percentage of nodes per degree, as it is related to the total number of nodes per degree.

2.4 Results and Discussion

The results obtained for the different datasets handled in this study are discussed below.

2.4.1 ECMstiff data

The outcomes obtained for the ECMstiff data are presented. Figure 2.14 shows the graph metrics obtained from the initial segmentation of this data. As mentioned previously, these metrics exhibit discontinuities due to the different methods considered for the first segmentation. This issue is corrected by the trained U-Net model, as shown in Figure 2.15, which illustrates continuous metrics. The assortativity

coefficient, in particular, shows an upward trend from negative values; however, this metric does not clearly capture the differences among all conditions. This behavior is also noticed for the clustering coefficient, which exhibits higher noise on the curves. The number of connected components seems to distinguish patterns between the first four conditions but does not illustrate differences between the last two conditions (5 and 6). This might be due to the lack of network development for these conditions. For the last two metrics, the node and edge-assembly ratios, there is no clear difference for almost all conditions.

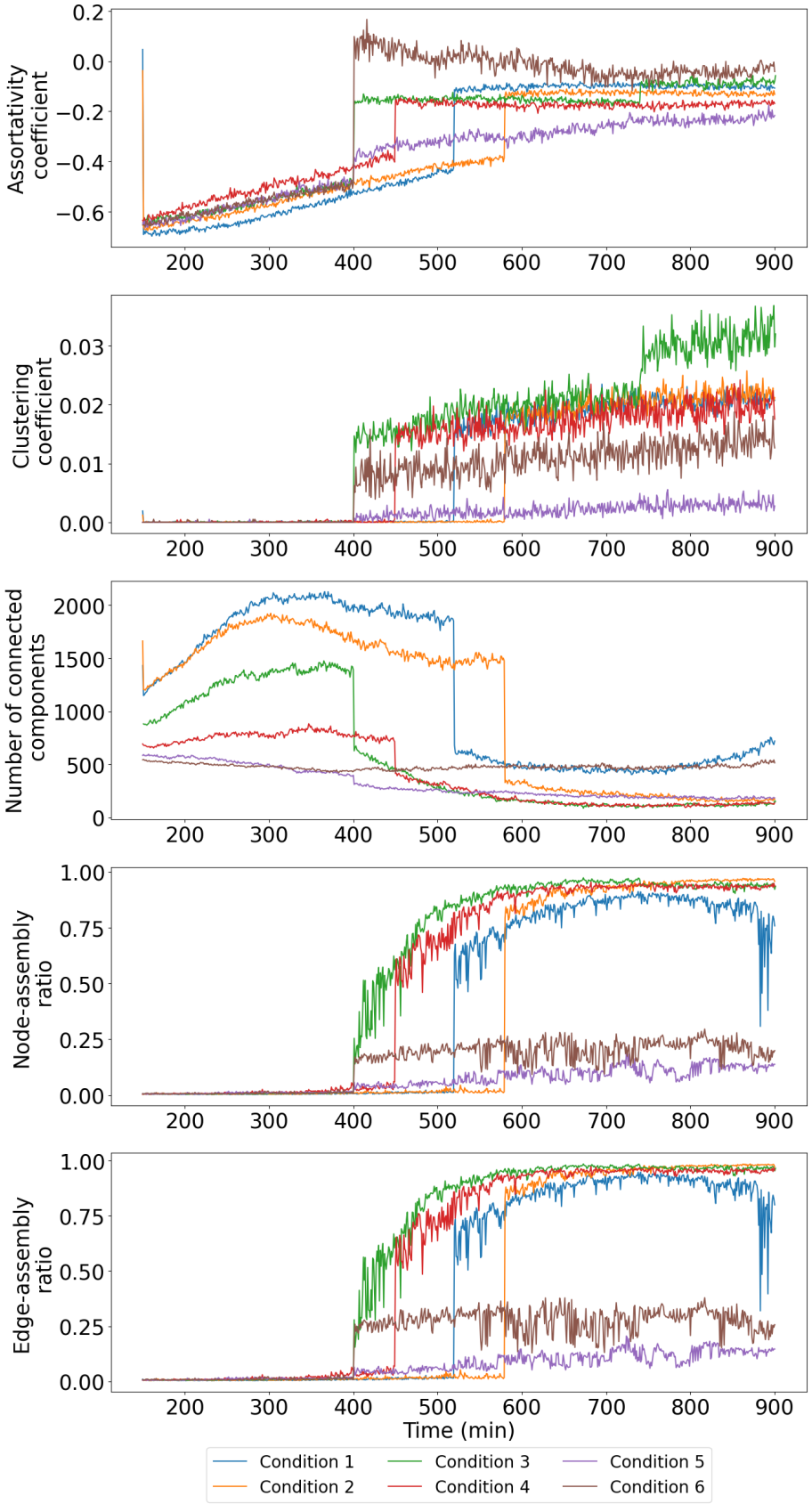


Figure 2.14: Graph metrics obtained from the initial segmentation of the ECMstiff data

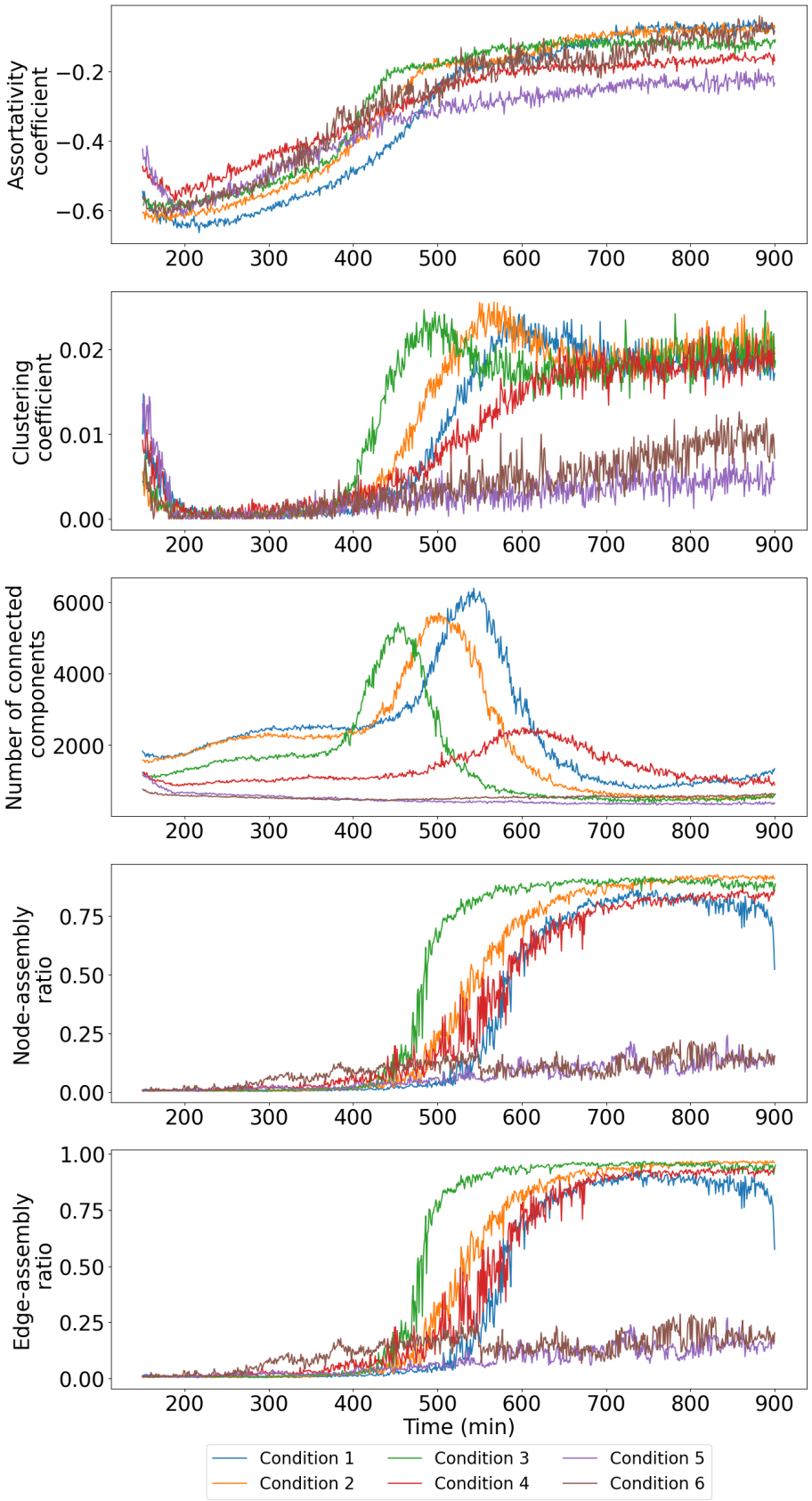


Figure 2.15: Graph metrics obtained from the automatic segmentation of the Lensless6X data

For the ECMstiff data, the observation time was set at frames $k = 150, \ldots, 1100$. In Figure 2.16, cropped (top left side) input images from condition 3 are shown,

taken at 5 time instances (in sequential order) from the time-lapse video. These images display their initial (as described in Section 6.1.1) and final (by the trained U-Net model) segmentation. It illustrates the process of network formation, starting from single cells and/or cell assemblies (at the beginning) and progressing to the complex cellular tissue network (by the last frame). When comparing the performance of the two segmentation steps, initial and automatic, it is noticed that the deep learning model demonstrates a good capacity for generating masks that closely match those produced by the initial segmentation method. Notably, the U-Net model addresses the issue of excess mask generation observed in the initial segmentation, particularly in thin cellular networks. This improvement is clearly visible at time point $t = 66h40min$. Furthermore, the U-Net model exhibits robust performance in identifying dense cellular clusters within complex networks. This capability is particularly observed in the final time point chosen ($t = 166h40min$), where the model successfully identifies the intricate structures and interconnected formations that characterize the overall cellular network at this advanced stage of development. These results are confirmed by examining the cropped graphs (top left side) generated from binarized images in Figure 2.17. These graphs provide a clear visualization of the key features and patterns that emerged from the analysis.

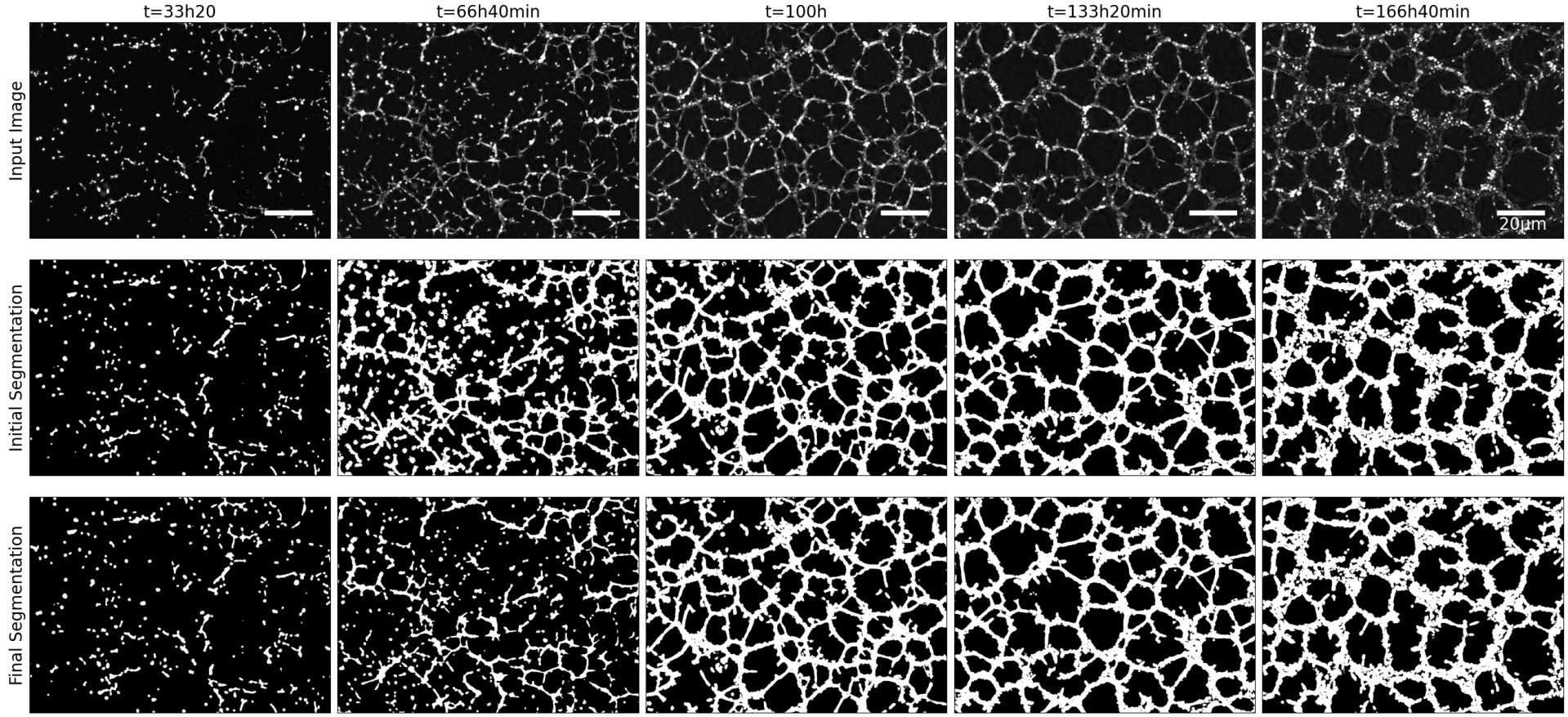


Figure 2.16: Evolving cellular network for condition 3 at 5 frames of the time-lapse video with their corresponding initial and final segmentation

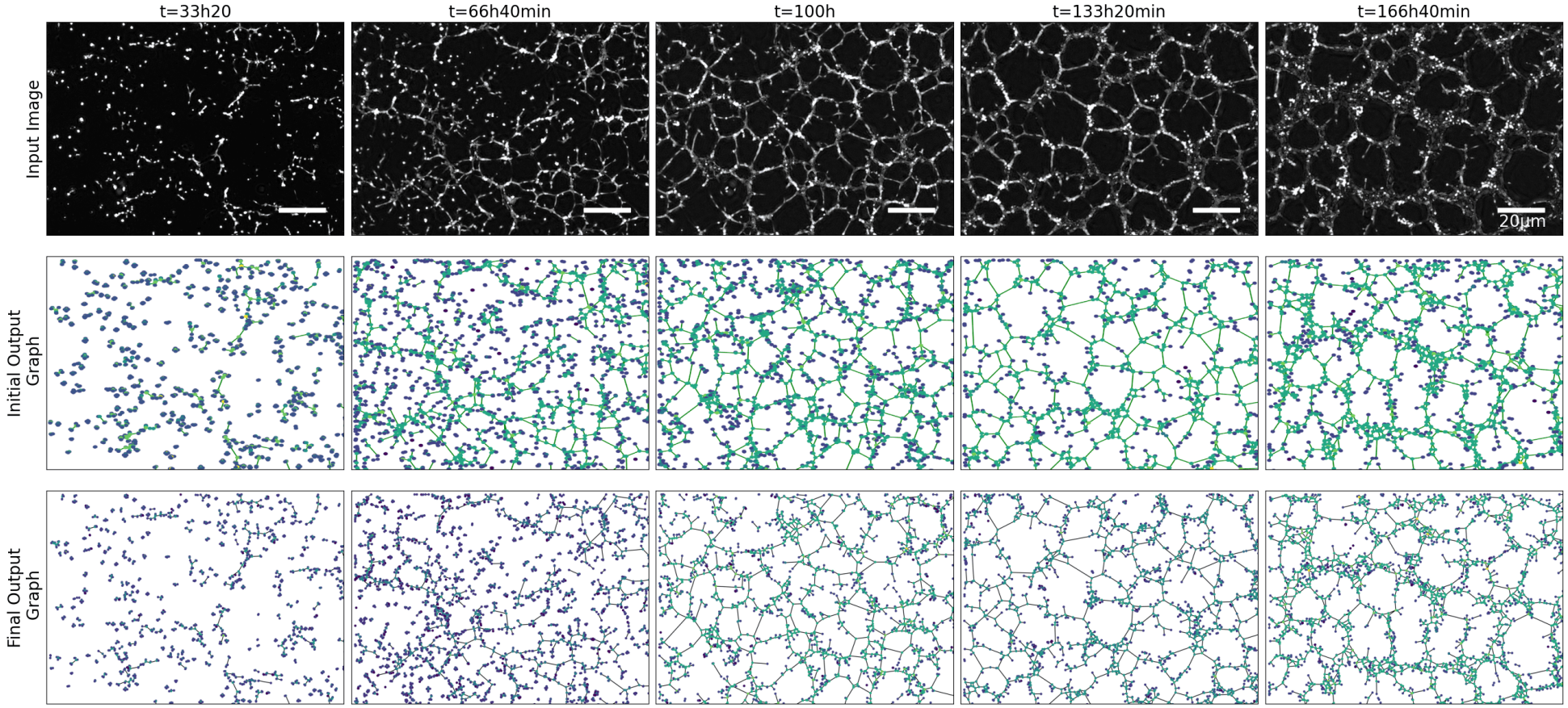


Figure 2.17: Evolving cellular network for condition 3 at 5 frames of the time-lapse video with their corresponding initial and final output graph

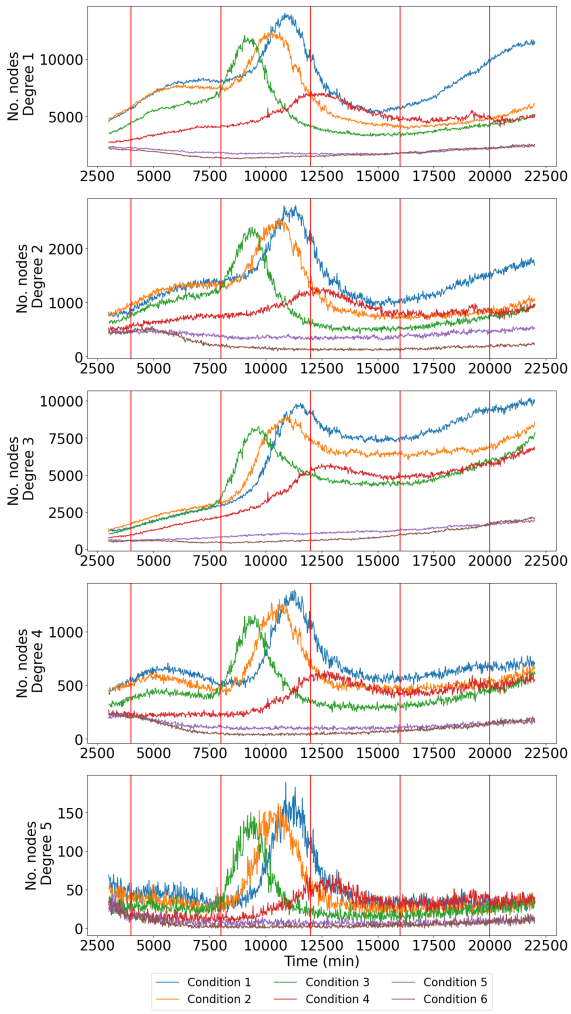
In Figures 2.18a and 2.19a quantitative metrics related to the degree per node are illustrated for each condition. Additional quantitative metrics for each condition in Figure 2.20 are also observed. For all these quantitative metrics, five time instances over the time-lapse video are highlighted (red vertical lines), and their corresponding graph-based network representations are shown in Figures 2.18b and 2.19b. Based on these results, condition 1 evolves into a dense, undifferentiated cellular mass by the final time point observed, lacking any discernible network structure. In contrast, conditions 2, 3, and 4 exhibit a well-defined and organized network structure at the same time point, indicating successful development of interconnected tubules or pathways. On the other hand, conditions 5 and 6 fail to establish a mature network by the final time point, likely due to the constraints imposed by their initial cellular assumptions, which could have limited the cells' ability to form or sustain interconnected structures over time.

It is observed that the temporal progression of the computed metrics on the final masked images exhibits a consistent trend without abrupt deviations from the initial segmentation. Additionally, the total number of nodes per degree (1, 2, 3, and 4) and the relative proportion of nodes for specific degrees (e.g., 1 and 3) demonstrate distinct differences between all experimental conditions. These differences are further validated by the graph-based visual representations shown in Figure 2.18b, which provide a clear and intuitive depiction of the underlying structural variations. Overall, these observations make possible the effective identification and analysis of the dynamic changes occurring within the evolving networks over time, offering valuable insights into their behavior and topology.

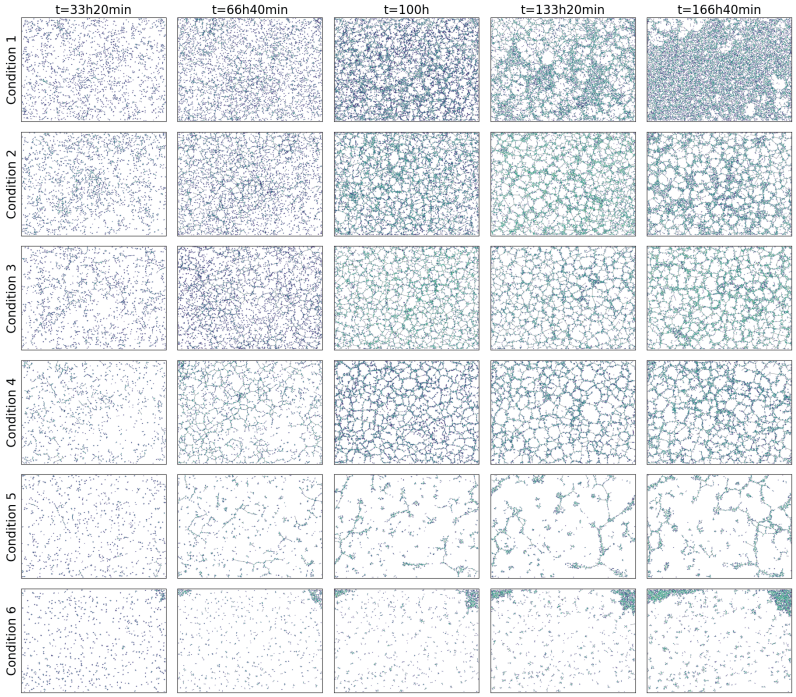
It is observed that the density, local efficiency, and the number of bridges exhibit differences that might characterize the network formation in this data. However, local efficiency shows high noise levels, making it difficult to describe differences between all conditions.

Interestingly, the number of junctions displays a pattern similar to that of the number of nodes with a degree of 3, which can be attributed to the scarcity of nodes with a degree of 4 or higher in comparison to the total number of nodes with lower degrees. This suggests that the network's topology is largely dominated by nodes of lower degrees, with higher-degree nodes occurring less often and not affecting the overall junction count.

Finally, the high levels of noise observed across the various metrics discussed earlier can likely be attributed to the presence of edge effects in the processed images and/or the method used to construct the graph from the skeleton. This motivates the application of the pruning algorithm to obtain simpler graphs that might better illustrate the differences observed previously.

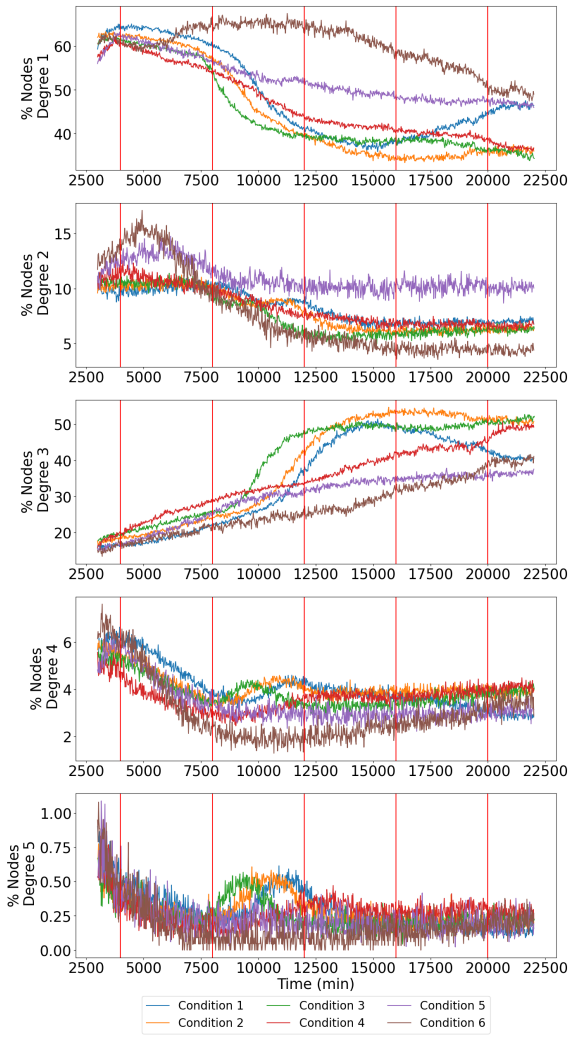


(a) Number of nodes per degree

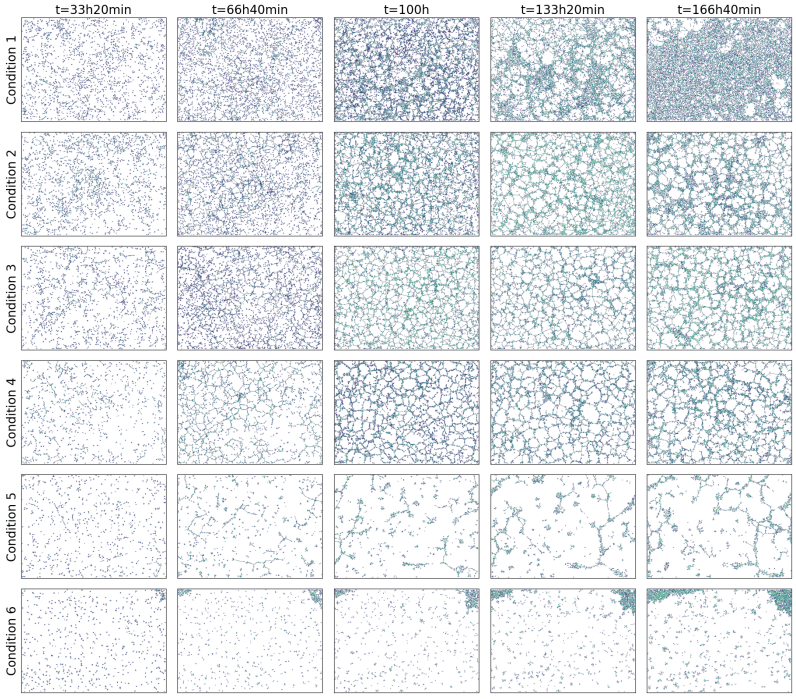


(b) Graphs for each condition at five selected frames, indicated by vertical red lines in the plots of node counts per degree.

Figure 2.18: Quantitative analysis for networks on ECMstiff data-Part 1



(a) Percentage of nodes per degree



(b) Graphs for each condition at five selected frames, indicated by vertical red lines in the plots of node percentage per degree.

Figure 2.19: Quantitative analysis for networks on ECMstiff data-Part 2

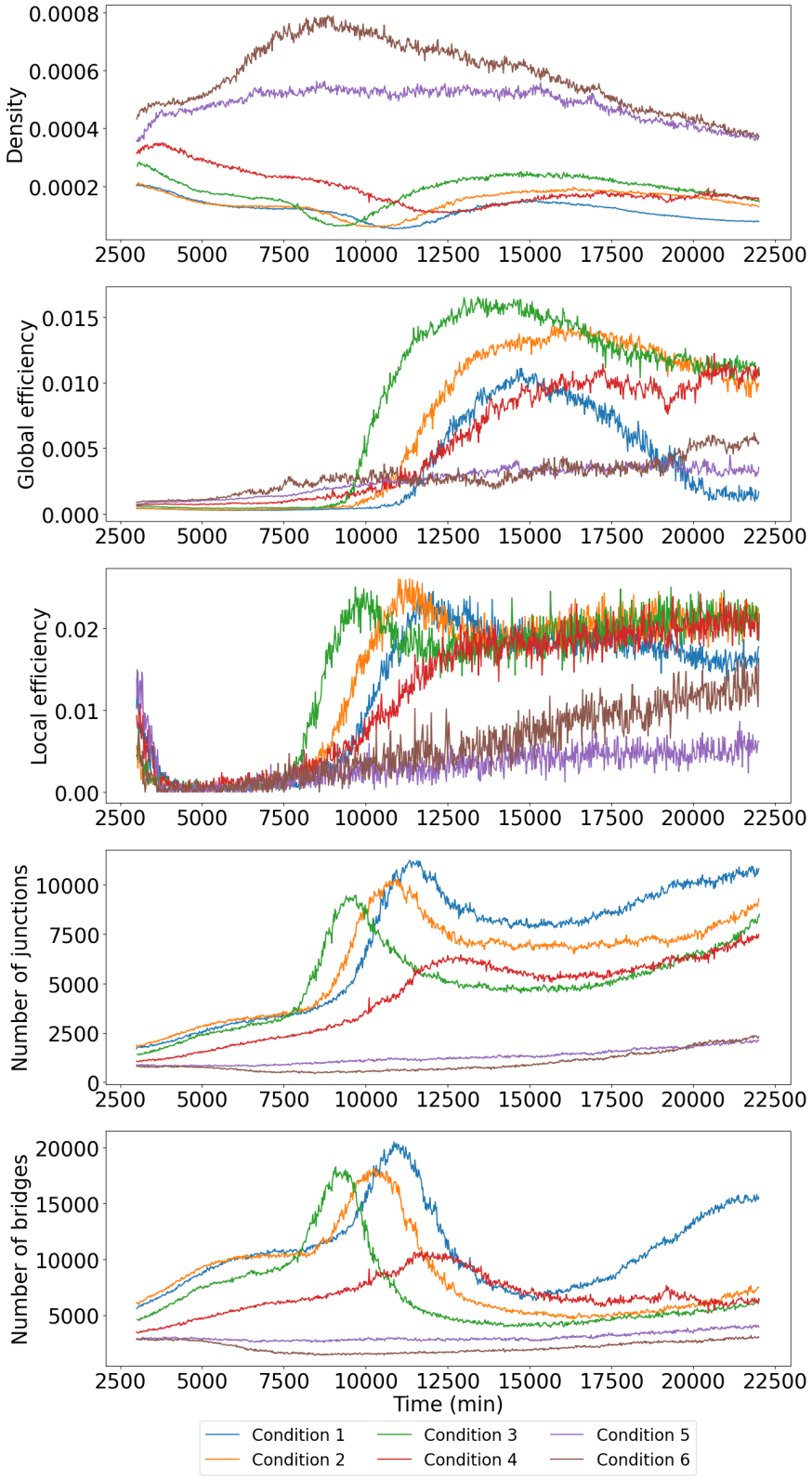


Figure 2.20: Quantitative analysis for networks on ECMstiff data-Part 3

2.4.2 ROCKi data

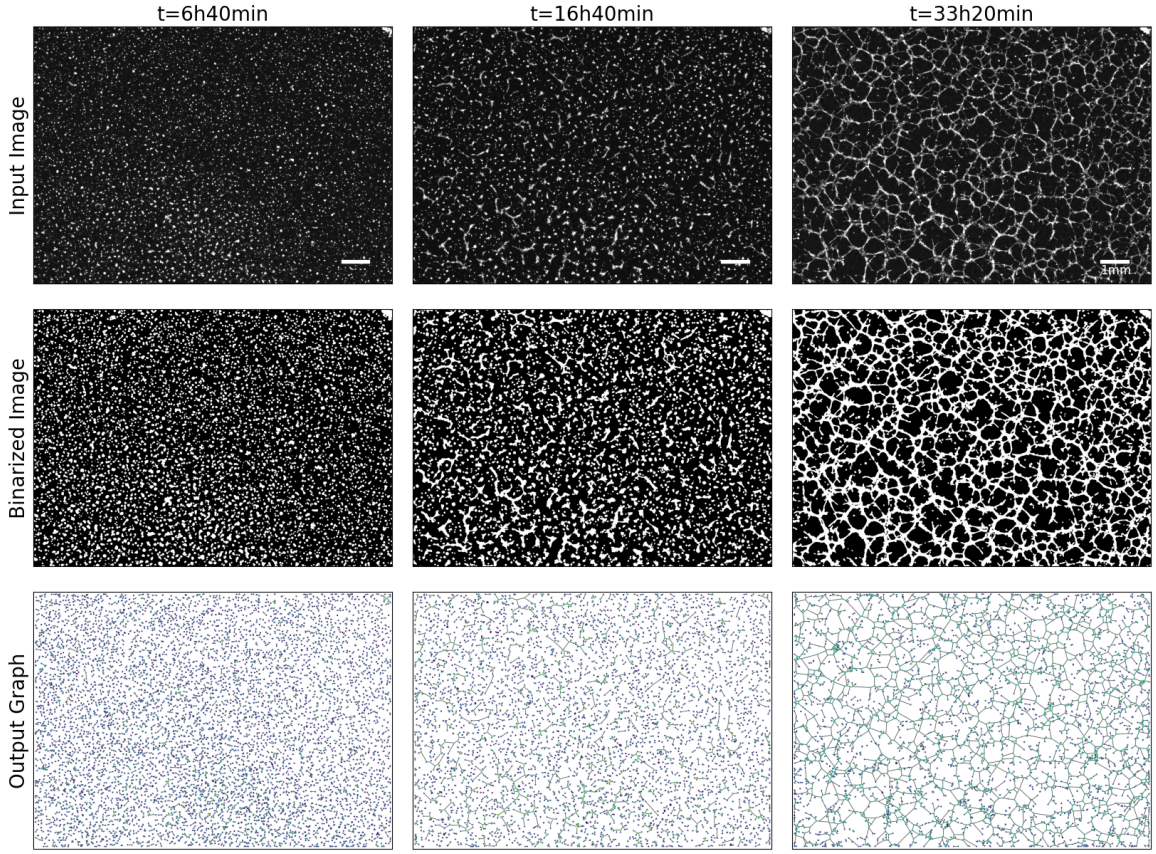


Figure 2.21: Evolving cellular network for condition 2 at 3 frames of the time-lapse video with its corresponding binarization and graph conversion

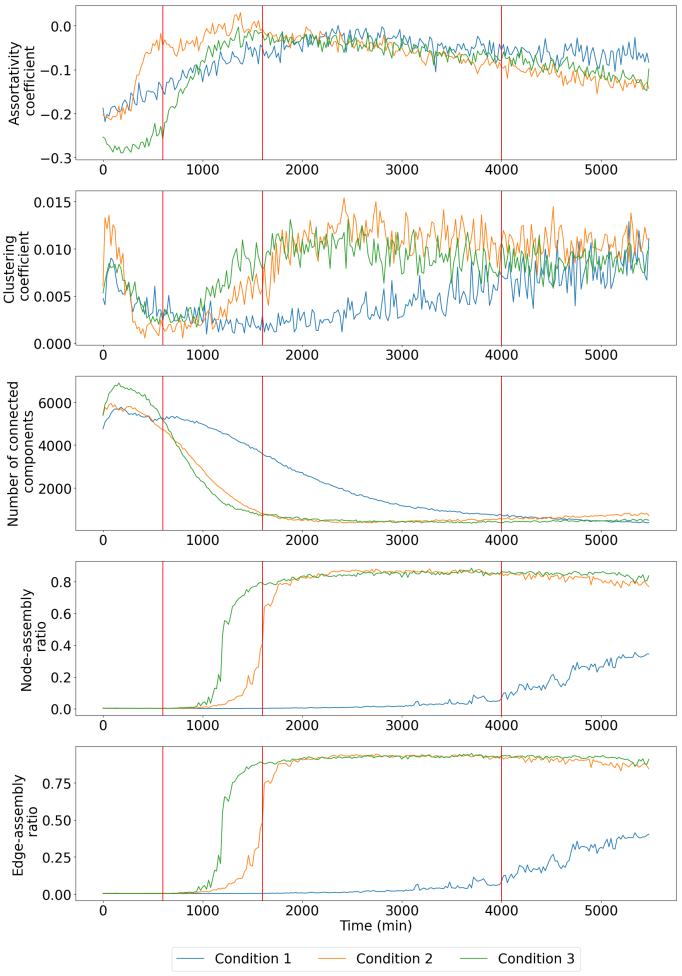
Figure 2.21 depicts the evolving behavior of the pancreatic epithelial network under condition 2 (1 μM ROCKi), showcasing its binarization and graph conversion at five time instances. Node colors illustrate their respective degrees within the connectivity network. This visualization illustrates how cells interact and associate over time, forming the network structure. Notably, network formation begins to take shape around $t = 33\,\mathrm{h}$, indicating that cell-cell interactions become more frequent during this period under this particular condition. To delve deeper into this observation, a thorough analysis of the metrics associated with each connectivity graph obtained at each time point is required. The observed merging behavior appears to be effectively captured by this approach. Furthermore, it is noticed that the segmentation method is capable of identifying and segmenting cells within evolving networks over time.

Quantitative metrics are computed and illustrated for each condition in Figure 2.22a. It is observed that there is no association between cells at the beginning in each condition. However, as time progresses, cell network assembly starts with dynamics depending on the experimental condition (ROCKi concentration). As expected, this behavior was reflected in all calculated metrics, albeit with varying degrees of effectiveness. Notably, the assortativity coefficient, showed a noisy increasing trend for all three conditions. However, despite the evident differences in the resulting graphs (Figure 2.22b), this metric did not allow for a clear distinction between the experimental conditions.

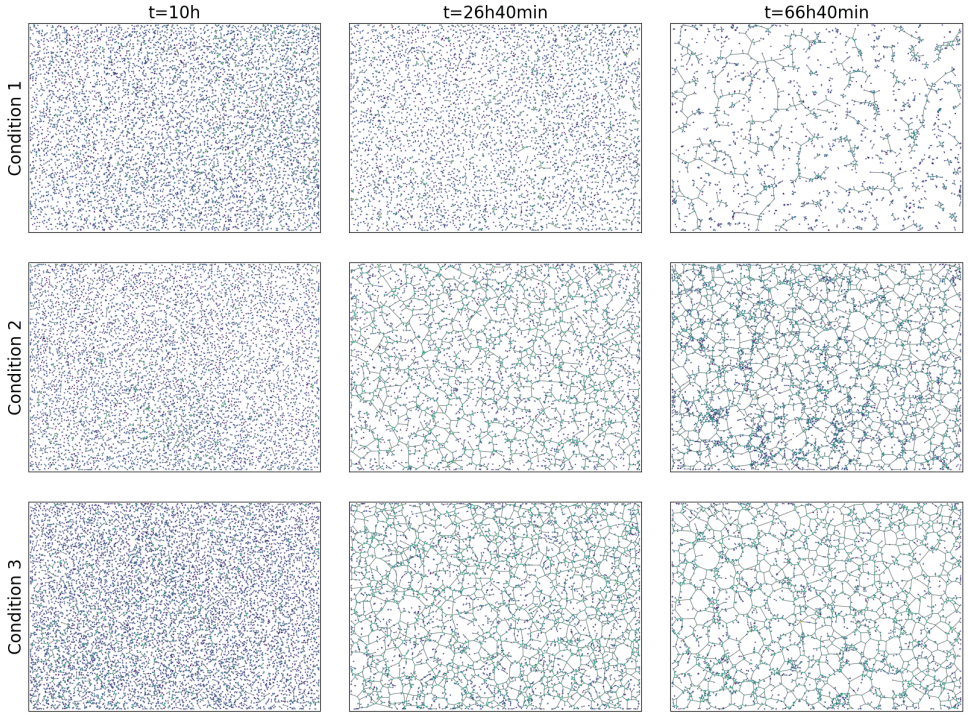
The global clustering coefficient demonstrates a positive (albeit low) increasing

trend with some fluctuations over time across all conditions. As this metric quantifies the relative frequency of triangles in the graph, the low values obtained may suggest that interactions between neighbors of the same cell assemblies are limited. Consequently, there is no clear connectivity behavior observed. However, despite this, the metric is capable of detecting the accelerating effect of ROCKi on cell network assembly, and even the subtle impact of ROCKi concentration (conditions 2 and 3), although the analysis remains somewhat noisy. Hence, in this study, both assortativity and global clustering coefficients appear to be suboptimal metrics for evaluating network architecture.

The number of connected components emerges as a reliable metric for identifying patterns in evolving networks over time. Initially, with no connections between cells, numerous components exist in the network. However, as time progresses, cells begin to associate, leading to a decrease in the number of connected components. This trend is also reflected inversely in the node- and edge-assembly ratios since increasing values for these metrics implies the connectivity of the largest component of the network is growing as well. This metric proves highly effective, clearly detecting changes in network dynamics induced by ROCKi. In comparison to the control condition (condition 1, 0 μM ROCKi), a gradual increase in the number of cell connections and accelerated network assembly is observed with increasing drug concentration. Furthermore, the related metrics, node- and edge-assembly ratios, demonstrate complementarity and performance by representing the evolution of the largest connected component and accurately recapitulating the observed acceleration effect of ROCKi on network dynamics.



(a) Quantitative metrics computed for all conditions



(b) Graph representation for each condition at $t = 10\text{h}, 26\text{h}40\text{min}, 100\text{h}$, indicated by vertical red lines in the plot of each metric.

Figure 2.22: Quantitative analysis for networks on ROCKi data

2.4.3 Actomyosin data

Finally, the performance of the U-Net model is tested on Actomyosin data. The observation time was set at frames $k = 5, \ldots, 300$, resulting in a total of 296 images for this analysis. From the available 2D microscopic videos in this dataset, six conditions were selected for analysis. Additionally, the same graph metrics were considered.

Figure 2.23 illustrates the characterization of epithelial network self-assembly. In the top left section, a plot of four different graph metrics, average clustering coefficient (ACN), assortativity coefficient (ASN), edge-assembly ratio (ERN), and global efficiency (GEN), with "N" denoting "network", and two geometric metrics, area (AreaN) and mean area (MAN), for a control condition is presented. The metrics are normalized with respect to their average value on the plateau at the end of the time-lapse video, except for the assortativity coefficient, which is normalized with respect to its average value at the beginning of the time-lapse and multiplied by -1. These metrics characterize the process of network formation in epithelial cells. Over time, each normalized metric, except for the assortativity coefficient, shows a growing positive trend that reaches a steady point, reflecting the consolidated network structure formed at the end of the time-lapse.

These plots also enable the identification of five states that characterize the process of network formation. These states are referred to as the lag, initiation, clustering, percolation, and relaxation phases. The lag phase refers to the scenario where cells are not yet moving or interacting with other cells. The initiation phase means that cells are beginning to move and associate with each other to form complex cellular components. This is followed by the clustering phase, during which different mesh structures resembling clusters in a graph are formed. Subsequently, the percolation phase begins, where cellular meshes start to connect with others to develop the network structure. Finally, in the relaxation phase, the already-formed network exhibits minimal changes in the size and shape of its mesh structures.

To better comprehend these network states, plots of these behaviors are illustrated in the top right corner of Figure 2.23. These plots include segmentation, graph-based representation, and simplified graph-based representation of the images at five time-steps of the video ($t = 2h, 37h, 45h, 60h, 100h$). The five states introduced earlier can be identified in the binarized images, which depict the process of epithelial network formation. This process is not clearly visible in the graph-based representation obtained. However, it can be more readily observed in the simplified graphs, which reduce the visual noise present in their original representation.

In the middle part, the first plot illustrates the evolution of mesh structures, which are defined by the edges of the graph formed within the network. Below this plot, the colored representation of these structures is shown at three time steps (60h, 80h, and 100h). It is observed how different meshes change their structures over time. To better quantify the dynamics of mesh formation, the shape factor q and the solidity[9] of the network are computed and plotted over time in the second and third plots, respectively. These metrics exhibit some fluctuations over time but ultimately display a steady trend by the end of the time-lapse, reflecting the formation of a natural network structure.

At the bottom, an analysis of the mesh structures within the network is presented. These structures are derived from the graph and generated using Lloyd's algorithm

[9]**Solidity** is defined as the image area, divided by the convex hull area [202].

[172]. This method creates mesh-like partitions from given points. The first graph corresponds to the probability density function (PDF) of Shape factor q values for the control condition, superimposed by the PDF of Shape factor q values for random Voronoi tessellations (in red), which are generated by Lloyd's algorithm [172]. The distribution of colored points over time is shown by the scatter plot at the bottom. Additionally, the inset shows the diagnosis, squared skewness versus kurtosis, of two distributions: control and random Voronoi, with two lines delimiting the normal log (red) and Gamma (blue) distributions. This indicates that both distributions fit the given distributions.

The second plot illustrates the mean Shape factor q versus its standard deviation (SD) for the control condition with its corresponding linear regression. It is clearly observed that there is a good fit for the proposed linear model.

The third plot displays the mean Shape factor q versus its standard deviation (SD) for a series of Voronoi graphs generated from the centroid positions of the experimental graph. The triangles in the regression plot represent data generated by Lloyd's algorithm, along with their corresponding regression lines. The images on the far right depict the initial and 20-step progress of Lloyd's algorithm and the distribution of centroids under control conditions (60h vs. 100h). Notably, there is a resemblance between the Voronoi tessellations generated by Lloyd's algorithm and the mesh structures within the network.

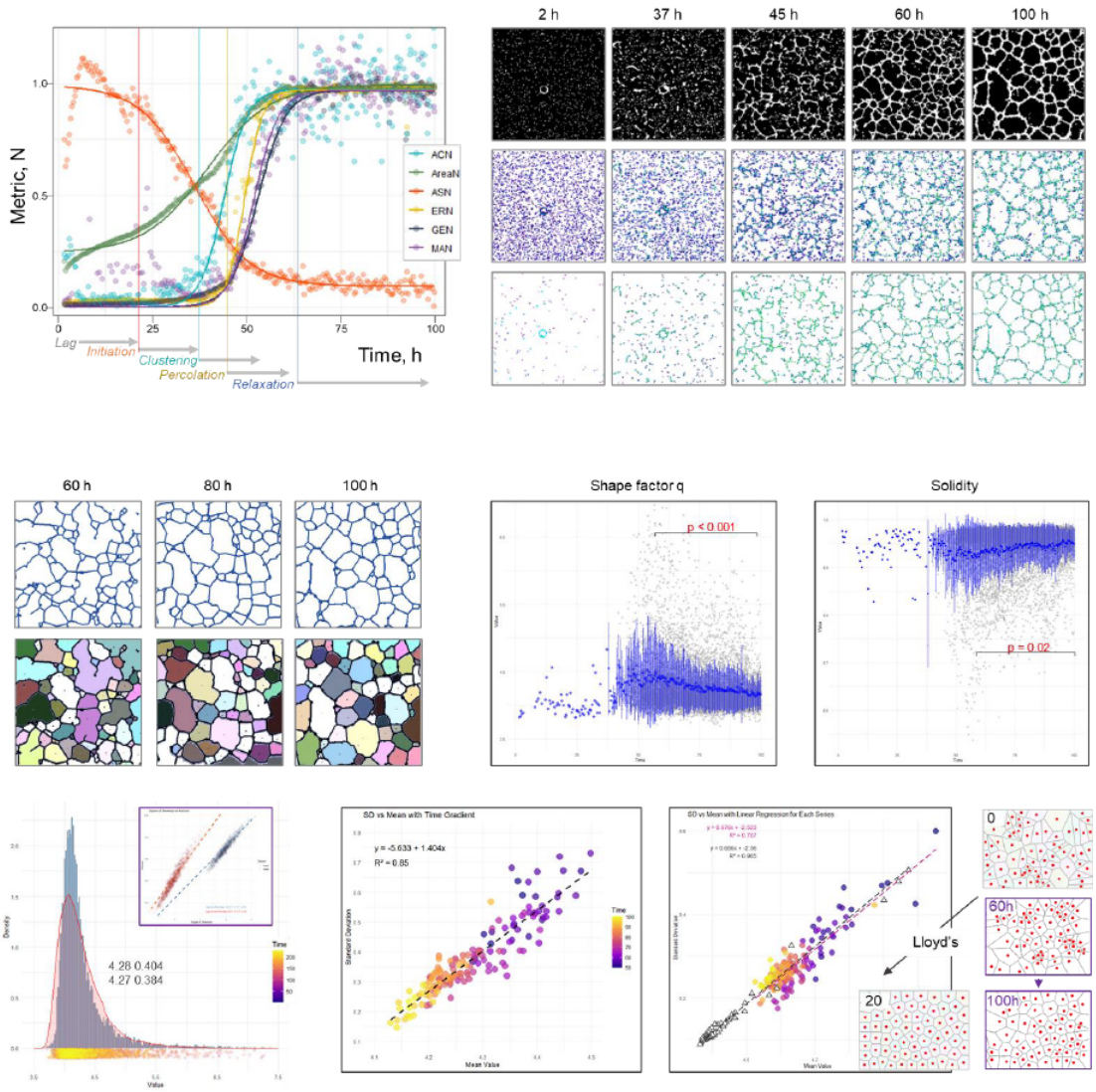


Figure 2.23: Characterization of epithelial network self-assembly. The top left panel depicts the plots of six graph and geometric metrics under a control condition. The top right panel illustrates the evolution of this condition through plots that include segmentation, graph-based representation, and simplified graph-based representations of the images at five time-steps of the video. The middle panel depicts the analysis of mesh structures within the network formation under the control condition. The bottom panel presents an analysis of the mesh structures within the network, derived from the graph and generated by Lloyd's algorithm.

Figure 2.24 presents an analysis of actomyosin regulation in the epithelial network. In this analysis, five conditions were considered: four representing different inhibitors (and their concentrations) of actomyosin, and the fifth serving as the control condition. These four conditions are referred to as Condition Y27632, Blebbistatin, ML7, and ML141, respectively.

In the top right section, graph-based representations derived from binarized images at five time steps ($t = 5h, 25h, 50h, 75h, 100h$) are plotted for each condition. These graphs highlight the differences in network formation resulting from the inhibitors considered. At $t = 100h$, Condition Y27632, a ROCK inhibitor, exhibits a dense network structure with numerous small mesh structures compared to the other conditions. Condition Blebbistatin, a Myosin inhibitor, also displays a developed

network structure with a considerable number of meshes varying in shape and size. The control condition shows a less dense network with fewer, larger mesh structures. Conditions ML7, an MLCK inhibitor, and ML141, a Cdc-42 inhibitor, exhibit the least network formation among all conditions.

These behaviors are also evident in the plots of graph metrics, illustrated in the top left section. The metrics considered include the assortativity coefficient, clustering coefficient, edge-assembly ratio, and global efficiency, which clearly allow to identify differences between all conditions.

Finally, an estimation of the duration of network states under these conditions is illustrated in the bottom left side. A graph-based representation of each network state, including initiation, clustering, percolation, and relaxation phases, is displayed in the bottom right side. It is observed that the conditions Blebbistatin and Y27632 exhibit the shortest (and equal) duration for the lag phase, approximately 3 hours, a brief duration for the clustering phase, and a prolonged duration for the percolation phase. These two conditions, along with the control, demonstrate the same approximate start time for the relaxation phase, which is the longest, approximately 50 hours.

The conditions ML7 and ML141 present the longest duration for the lag phase, approximately 25 hours. Condition ML141 exhibits a short initiation phase, followed by a moderate duration for the clustering phase. This is followed by the longest percolation phase observed, approximately 50 hours, ending with the shortest relaxation phase.

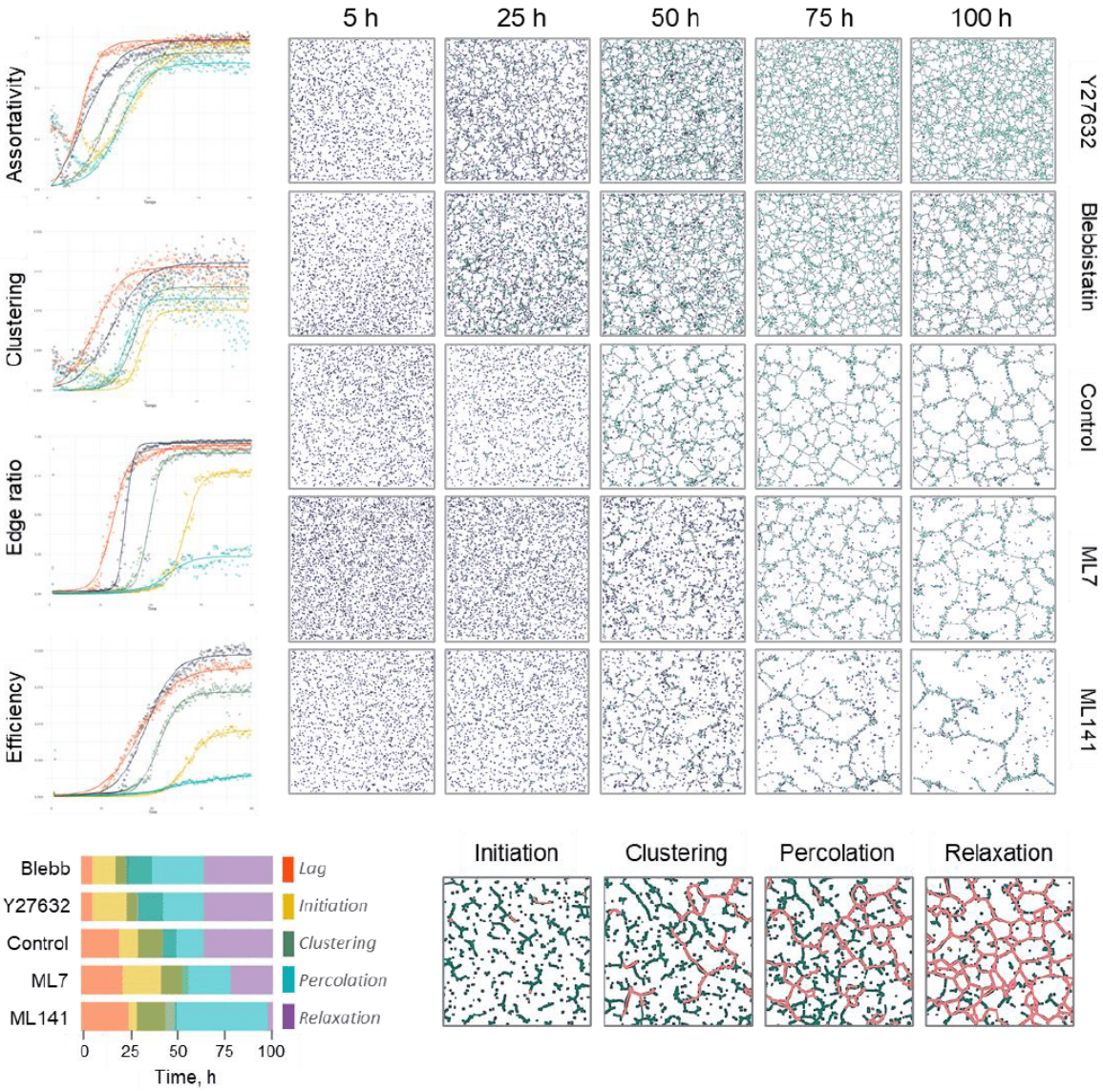


Figure 2.24: Analysis of actomyosin regulation of epithelial network. The top panel presents the analysis for five conditions, one of which serves as the control. The right side displays graph-based representations derived from binarized images at five time steps. The left side illustrates the plots of the four most representative graph metrics for the five conditions. The bottom panel provides an estimation of the duration of the five states in network formation for these conditions, accompanied by a graph-based representation of these states.

Conclusion

In brief, the proposed deep learning-based model addresses the problem presented for the segmentation task by leveraging the discontinuities observed in the plot of graph metrics, which were generated by the initial segmentation. Furthermore, the model demonstrated good performance and the ability to generalize to unseen data. This facilitates the transition for subsequent graph-based analysis, where it was possible to identify metrics that can explain and distinguish the dynamics of network formation under the various conditions studied in this chapter.

It is also important to note that the graph metrics employed in the analysis

are sensitive to the method used for graph construction. As previously described, the method for obtaining the graph structure is derived from the skeleton obtained through the proposed segmentation process.

Chapter 3

Graph-based clustering

In the preceding chapter, the behavior of cells over time was analyzed, from the formation of cell clusters to the development of cellular networks, under different initial conditions. These analyses were based on the overall network structure represented as a graph, from which metrics were primarily extracted at the graph level. The subsequent analyses will be conducted at the node level. Specifically, the work will focus on clustering nodes, i.e., cells, to attempt to separate two cell types.

Single-cell analysis has been extensively developed within the research community. In the field of microscopic imaging, this analysis involves the detection, segmentation, and feature extraction of cells, followed by the utilization of these features to achieve a biological objective. However, single-cell analysis in imaging microscopy faces several limitations. Some of them include the lack of information regarding neighboring cells, which could provide valuable insights into biological phenomena, and the difficulty in tracking cells, which often leads to errors in identifying and monitoring cells over time.

To address these limitations, a spatio-temporal graph structure leveraging spatial and temporal edges is introduced. This newly proposed graph structure integrates both spatial and chronological links between cells to extract distinctive attributes at the node level. Therefore, the utility of this graph structure will be evaluated in this chapter.

The datasets employed in this work consist of 2D microscope images of cell cultures that evolve over time, as well as simulations based on the same characteristics, i.e., 2D images of objects that evolve over time. Different datasets will be available, which primarily differ according to two characteristics: the first pertains to the type of data, specifically experimental data from HeLa, fibroblast, perKO, and HDF (Human Dermal Fibroblasts)-HUVECs (Human Umbilical Vein Endothelial Cells) cultures, and simulations. The second characteristic refers to whether the two cell types are mixed in culture (both cell types are present in the same image) or whether two cultures are performed in parallel, each containing one type of cell.

Two learning-based prediction tasks have been designed: the identification of cell types from two distinct populations and two mixed populations. To address these tasks, unsupervised and supervised methods leveraging the extracted node-level attributes were employed. As unsupervised methods, a Gaussian Mixture Model (GMM) and the He2CL algorithm [188] were utilized, the latter specifically designed for the aforementioned cell cultures. As supervised methods, a Support Vector Machine (SVM) model and a specialized Graph Neural Network (GNN)-based model were employed.

Dataset	Task		Method	
	Two distinct populations	**Two mixed populations**	**Unsupervised**	**Supervised**
HeLa and fibroblast cells	X		GMM, He2CL	
PerKO cells		X	He2CL	
Simulations	X		GMM	SVM
		X		SVM, GNN
HDF-HUVECs		X		GNN

Table 3.1: Summary of the analysis conducted. Gray boxes indicate that the task and/or method was not performed on that dataset.

Table 3.1 presents an overview of the analysis conducted in this chapter. Initially, the first prediction task, which involves the identification of cell types from two distinct populations, was addressed. For this task, two cell cultures were employed: one consisting of HeLa cells and the other of fibroblast cells. These cultures were imaged separately and analyzed using distinct graphs. Node-level attributes were computed through the spatio-temporal graph structure for each cell culture. These attributes were then merged to perform clustering, which aimed to identify both cell cultures. Two clustering methods were employed: a GMM and the He2CL algorithm [188], which was specifically designed to process these cell cultures. The results indicate that the incorporation of node-level attributes improved the performance scores for both methods compared to considering only cell morphological or cell morphological-dynamic features. Notably, the outcomes for the He2CL algorithm were superior to those previously obtained. For simulated data, it was observed that the GMM performed well in this task, achieving high scores when attempting to separate two distinct simulated datasets.

Subsequently, cell cultures containing mixed cell types were utilized to evaluate the second prediction task, which involves the identification of cell types from two mixed populations. For this purpose, node-level attributes were computed based on the spatio-temporal graph structure extracted from the cell culture. These node features were then employed for clustering, similar to the previous approach, to identify the cell types. The perKO cell culture was analyzed using the He2CL algorithm, which performed well and yielded qualitative results. However, simulated data containing two cell types exhibited poor performance for this task when using the GMM.

Finally, since the unsupervised method did not successfully enable the second prediction task, which involved the identification of cell types from two mixed populations, a supervised method based on a GNN model was proposed to classify the two cell types. This supervised model leverages node-level attributes extracted from the spatio-temporal graph structure. Initially, simulated data were used to evaluate the supervised model on the second prediction task, achieving high performance. Subsequently, a culture comprising Human Umbilical Vein Endothelial

Cells (HUVECs) and Human Dermal Fibroblasts (HDF), with HUVECs labeled with a fluorescence tag, was employed to evaluate the Graph Neural Network (GNN)-based model for this task. However, this model yielded poor performance. This outcome can be attributed to challenges associated with data preprocessing and the determination of the ground truth.

3.1 Research context

Cells are one of the smallest unit of life capable of independent existence and of performing essential biological functions, such as metabolism, growth, and reproduction [81]. Cell communication and interactions are fundamental for maintaining the structure and function of multicellular organisms. Through direct contact or chemical signaling, cells coordinate vital processes such as tissue development, immune responses, and homeostasis [8].

Assessing the impact of these communications requires a quantitative spatial and temporal analysis of cells. Microscopy is generally recognized as a powerful tool for this task [68]. Live or video microscopy extends this capability by capturing dynamic processes in real-time, enabling the study of cellular movements, interactions, and responses over time. Lens-free microscopy is a particular tool employed for this purpose [12]. This method presents several benefits, including not requiring lenses, providing a large field of view, and supporting label-free, real-time imaging [10, 13, 29, 222].

Single-cell analysis is a valuable method for studying heterogeneity within biological systems, as it enables the investigation of individual cells and cell-to-cell variation across diverse biological contexts, including organs, tissues, and cell cultures [121]. In a broader context, single-cell analysis corresponds to the genomic and transcriptomic analysis of the cell. In the context of microscope image analysis, single-cell analysis comprises the detection of cells from acquired images using microscopic tools such as lens-free microscopy, their segmentation through deep-learning methods, feature extraction of each cell, and the subsequent use of these features for a biological objective.

However, single-cell analysis in imaging microscopy has several limitations. Some of them include the absence of information related to neighboring cells, which could be necessary for better characterizing single cells, and the difficulty in tracking cells, which often results in errors in identifying and monitoring cells over time.

To overcome these mentioned limitations, a spatio-temporal graph structure based on spatial and temporal edges is proposed. This structure, referred to as Graph 3D, offers a solution to address the identified challenges. This structure captures connections with spatial neighbors and goes further by encoding temporal dynamics (connecting each cell to its neighbors in the following frame) without the need to track individual cells (only their positions are required). Therefore, the objective of this chapter is to evaluate the utility of Graph 3D.

As detailed in the following sections, this work focuses on a graph-based clustering application from an alternative perspective. Specifically, 2D microscope images of cell cultures that evolve over time or simulations based on the same characteristics, including only the cell x-y positions, will be utilized to construct Graph 3D. Subsequently, certain node features derived from the topology of this new graph structure will be extracted.

To determine whether those new cells features can have a biological interest, an analysis of two case studies was conducted. The first case study involved the calculation of spatial and temporal features at the node level based on graph theory. The second one requires to develop a Graph Neural Network (GNN)-based model, which allows for the incorporation of edge information and the calculation of more complex features than classic graph theory metrics.

In these case studies, attempts will be made to separate cells from two distinct populations or from a population consisting of two cell types using Graph 3D-based node features.

3.2 Literature review

Graph clustering is a learning technique focused on partitioning nodes in a graph into tightly connected groups (clusters) based on shared attributes (features) [217]. The field of graph clustering includes a wide range of techniques [137], though some are more frequently applied in practice, such as: K-Means [173, 185], Hierarchical [200], Gaussian Mixture Model (GMM) [190], etc. These methods use similarity measures to assign a specific node to the appropriate cluster.

In some cases, raw node data do not provide enough context to assess similarity [217]. Therefore, graph embeddings must be generated (as discussed in Section 1.3.3.1) to effectively apply the relevant clustering methods [217]. Node embedding methods, such as Node2Vec [100] and DeepWalk [212], map graph nodes to vectors in a reduced-dimensional space. Generally, nodes that are similar will have closely related embeddings (vectors), and conversely, dissimilar nodes will have different embeddings.

On the other hand, some graph clustering techniques do not rely on similarity data but instead use connection data to group nodes. Modularity-based algorithms, such as Louvain [36] and Leiden [259], aim to find the graph partition with the highest modularity [196], a measure that indicates the strength of relationships within clusters relative to a random distribution.

Besides, label propagation [220, 302] is another graph clustering method that relies on network connectivity. This method assigns labels to nodes and iteratively updates them by passing labels between neighboring nodes. Each label has a probability, and each node receives the label with the highest probability.

GNN-based methods for clustering utilize encoders such as graph convolutional networks [146], graph attention networks [269], and graph auto-encoders [145] to model non-Euclidean graph data, leveraging their strong ability to represent graph structures [168]. These methods have demonstrated promising results, including MGAE (Marginalized Graph AutoEncoder) [273], which learns both node attributes and graph structure, and GALA (Graph convolutional Autoencoder using LAplacian smoothing and sharpening) [210], a symmetric graph auto-encoder. However, the combination of transformation and propagation in GNNs can result in high computational costs. To address this, SCGC (Simple Contrastive Graph Clustering) [170] was introduced to enhance the efficiency and scalability of deep graph clustering by decoupling these two operations.

When time is a factor, nodes and their interactions with neighboring nodes can change and evolve over time, a dynamic that traditional static graphs (whether undirected or directed) are unable to capture. This necessitates the use of more

sophisticated structures capable of representing the temporal dynamics exhibited by nodes. Temporal graphs offer a solution to this challenge.

Temporal (or dynamic) graphs are data structures whose connections and/or nodes change over time [122]. A temporal graph can be conceptualized as a series of time-stamped events, denoted as $\mathcal{G} = \{x(t_1), x(t_2), \ldots\}$, which signify the addition or modification of a node or the interaction between two nodes at discrete time points $0 \leqslant t_1 \leqslant t_2 \leqslant \ldots$ [229]. Each event $x(t)$ can be categorized into two types:

- **Node-specific Events:** These events involve the creation or updating of a node. Each node-specific event is represented by a vector attribute associated with a particular node index. If the node index is new, the event signifies the creation of a node with specified attributes. If the node index already exists, the event updates the node's existing attributes [229].

- **Interaction Events:** These events represent interactions between pairs of nodes and are depicted as directed temporal edges. Multiple edges between the same pair of nodes are allowed, making the graph a multigraph [229]. A multigraph is characterized by the presence of multiple edges between any two nodes, enabling the representation of complex interactions over time.

Dynamic graphs can be modeled using two main approaches. The first approach, known as *discrete-time dynamic graphs* (DTDG), involves capturing a sequence of static graph snapshots at set time intervals. Specifically, a graph snapshot represents the state of the graph, including its present nodes and edges, at a given point in time. The second approach, *continuous-time dynamic graphs* (CTDG), provides a more flexible representation by recording time-stamped events. These events can involve changes to the graph's structure, such as adding or removing nodes and edges, as well as alterations to the properties of nodes and edges [229].

Early research on dynamic graph learning mainly targeted discrete-time dynamic graphs (DTDGs) [229]. Methods included aggregating graph snapshots for use with static models [120, 163], factorizing snapshot tensors [78, 184], or generating embeddings for each snapshot. The embeddings were then combined through weighted aggregation, modeled with time-series approaches, incorporated into recurrent neural networks (RNN), or learned under temporal smoothness constraints [235, 288]. Another direction involved random-walk–based methods, where walks were initialized on the first snapshot and adapted for subsequent ones [186, 293].

Recent studies on continuous-time dynamic graphs (CTDGs) have introduced methods ranging from random walk models with time-dependent transition probabilities [24, 197, 198] to RNN-based sequence models that update node embeddings upon edge arrivals [152, 183, 261]. Parallel efforts in dynamic knowledge graphs highlight similar challenges [71, 94, 98, 286]. While many frameworks maintain node memories updated by RNNs, they often omit neighborhood-level aggregation, which not only risks producing outdated embeddings but also constrains representational power.

Due to their topology, temporal graphs present additional challenges for clustering. To address this issue, CGC (Contrastive Graph Clustering) [211] uses an incremental learning approach for graph clustering. VGRGMM (Variational Graph Recurrent Gaussian Mixture Model) [160] employs a variant of gated recurrent units (GRU) to capture temporal information. Additionally, Liu et al. [166] present a general framework for temporal graph clustering that allows adaptation to the interaction sequence-based batch-processing pattern of temporal graphs.

MAGIK is a recently GNN framework designed to capture object dynamics and physical interactions in microscopic videos using graph-based representations. In this model, nodes correspond to object detections at specific time steps, edges link objects that are spatially and temporally proximate, and global features represent system-level properties [213]. MAGIK supports a range of graph learning tasks, with a particular emphasis on node classification. This focus aligns with the supervised methods discussed in this chapter.

All previously introduced graph-based models for clustering utilize feature matrices derived from different methods as input data for these models. These methods include the concatenation of binary word vectors (with a fixed dictionary) from paper citation, patent citation, or author networks [170, 210, 211, 273]; one-hot encoding representations of node degrees from air traffic networks [193]; the concatenation of vectorized connectivity patterns across time points and subjects from human brain tissue connectivity graphs [160]; and bag-of-words representations from product reviews [169]. In contrast, the proposed feature matrix will be constituted solely by Graph 3D-based node features. Initially, these features will be composed of metrics related to the spatial-temporal structure of this graph, given that only the x-y positions of cells are known. These metrics are computed based on the spatial and temporal proximity of nodes across continuous time points or frames. These node attributes can be complemented by cell morphological information when making comparisons of outcomes, as will be shown later.

In some cases, the feature matrices derived from the graph-based models presented before may be more complex in size than the one proposed using Graph 3D-based node features. This could increase the time and computational effort required for calculations. Furthermore, GNN methods based on temporal graphs require knowledge of the tracking information of nodes for their learning processes. This information is not necessary for the proposed GNN-based model, as it will be encapsulated through the temporal connections created in the Graph 3D.

It is important to note that it was not feasible to employ all previously introduced graph-based clustering methods to evaluate the two proposed prediction tasks due to computational and memory constraints.

3.3 Methodology

Figure 3.1 provides a global overview and summary of the proposed workflow. A spatio-temporal graph framework was developed using all the time-lapse images from simulated or experimental data to derive node attributes associated with the spatial and temporal architecture of the network. Subsequently, either an unsupervised or supervised method was utilized to classify all cell types.

To evaluate the performance of this spatio-temporal network structure in identifying cell types, two tasks were proposed, as described below.

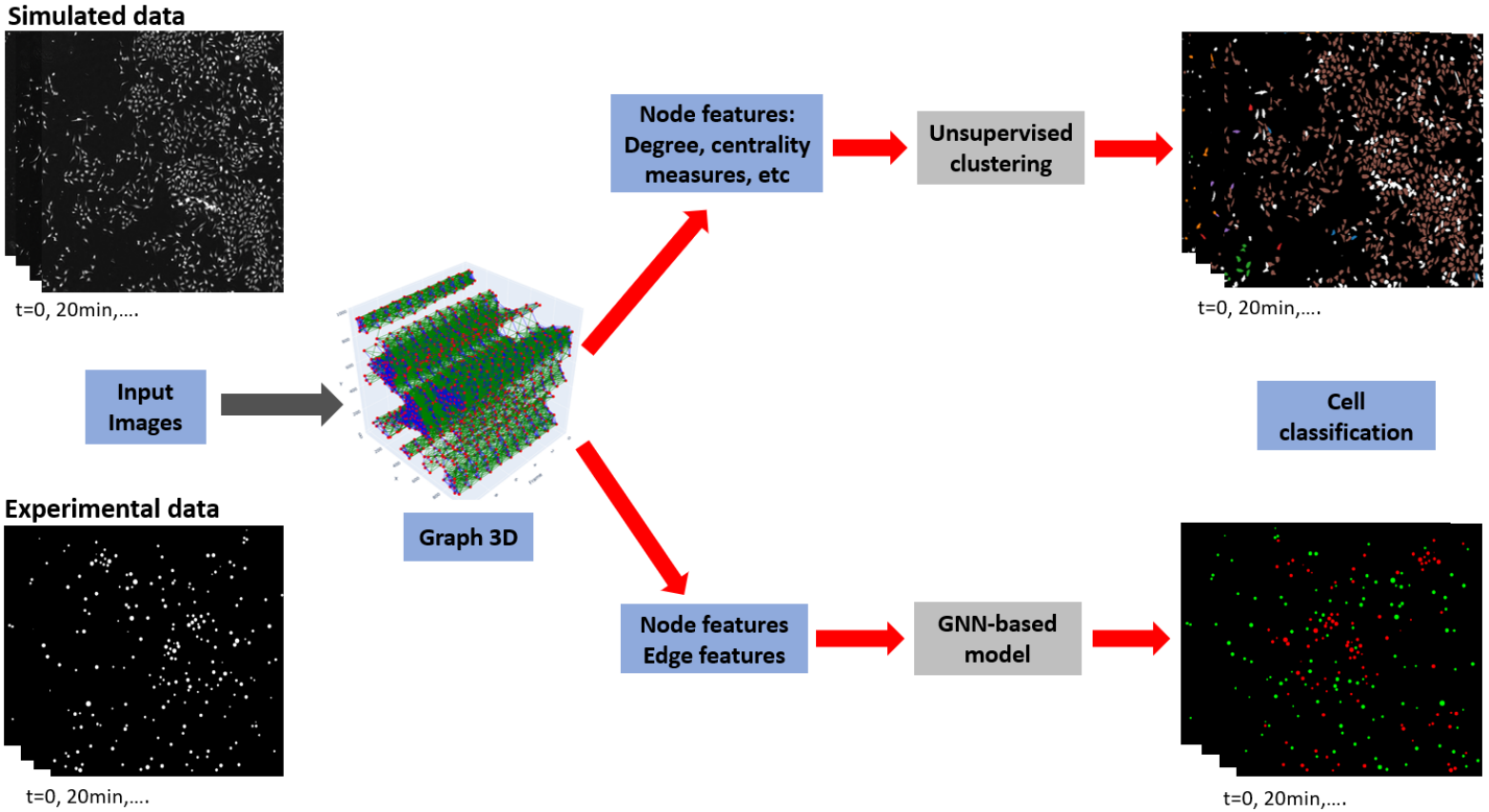


Figure 3.1: Proposed workflow for graph-based clustering. A spatio-temporal graph structure was built using all the time-lapse images from simulated or experimental data to extract node features related to the spatial and temporal topology of this network. Finally, either an unsupervised or supervised approach was employed to identify all cell types.

3.3.1 Prediction tasks

For the analysis, two prediction tasks are proposed. In each task, different combinations of spatial and temporal aspects on graph structures are considered. The first prediction task involves identifying cells within two distinct populations. The idea is as follows: given two datasets, the Graph 3D and its respective node features are computed for each. These features are then merged, and an attempt is made to separate them using either a supervised or an unsupervised approach (clustering). Specifically, a Support Vector Machine (SVM) model and a Gaussian Mixture Model (GMM), a well-established clustering technique, are applied to the calculated node features to identify and segregate the distinct datasets.

The second prediction task involves identifying cells within two mixed cell populations. In this case, for a given dataset, the Graph 3D and its respective node features are computed. These features are then utilized to identify the two cell types using either a supervised (SVM) or an unsupervised approach (GMM). In addition to the SVM model, a Graph Neural Network (GNN)-based model is proposed to address this task, as detailed in Section 3.3.3.2.

3.3.2 Datasets

3.3.2.1 Simulated data

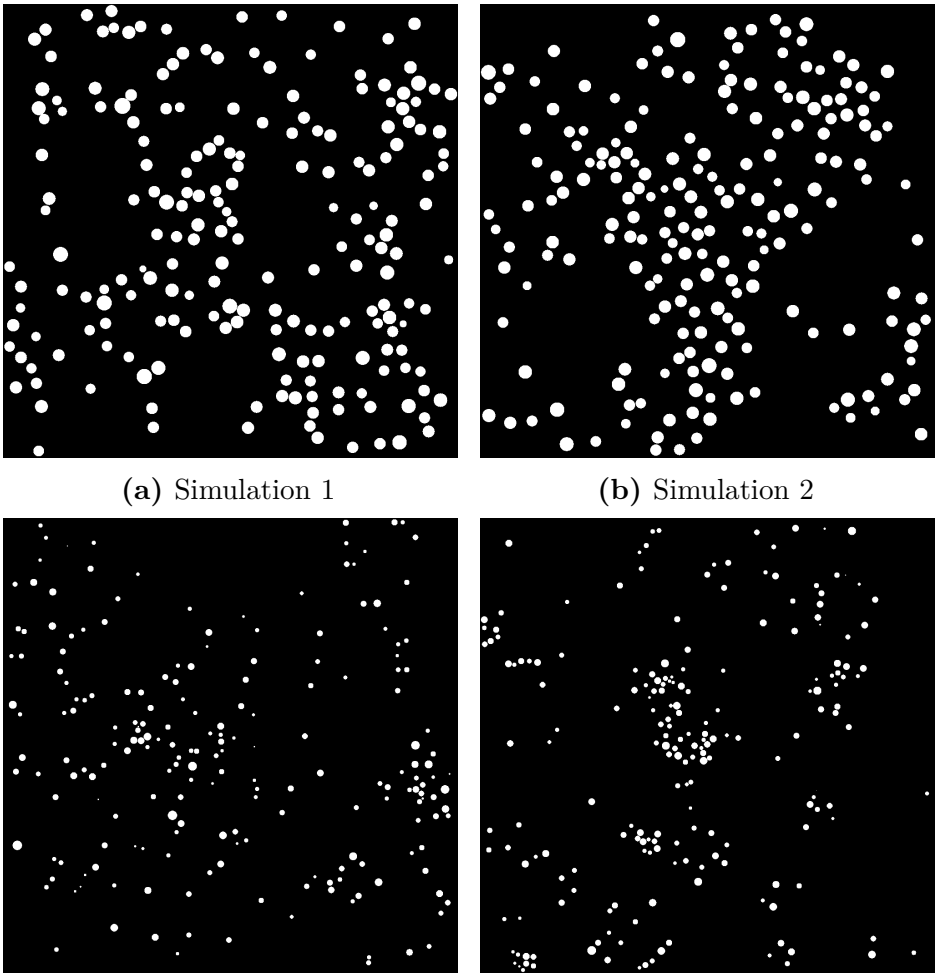
The simulated data are derived from 2D video simulations of cell cultures. The displacement of each cell is governed by its spatial proximity to other cells. At the beginning, cells move randomly. As time goes by, they tend to approach neighboring cells, which causes a force of attraction or repulsion between them. For all simulations, only the x-y position is considered, and no other variables are simulated. This approach is taken to determine whether Graph 3D can effectively extract neighborhood information.

The simulation data were divided into two datasets for the analysis of the two prediction tasks (described in detail above). These datasets correspond to the identification of cell types from two distinct (artificially mixed) and mixed cell populations, respectively. The first set of simulated data comprises 2D videos featuring a unique cell type. Each video contains 200 cells per frame, with a total of 51 frames. This set of simulated data is employed to attempt the separation of two distinct simulations through artificial mixing, utilizing both supervised and unsupervised approaches. Figure 3.2 illustrates one frame for four of these simulations, with the last one corresponding to the simulation with two cell types. These images showcases different behaviors as previously mentioned.

The second set of simulated data consists of 2D videos containing two cell types per frame image. In total, nine simulations were performed, each consisting of a video with 201 frames and 200-217 cells per frame. This set of simulated data is utilized to evaluate the second prediction task using both supervised and unsupervised approaches. Figure 3.3 illustrates one frame for four of these simulations, showcasing different behaviors as previously mentioned.

3.3.2.2 Experimental data

Experimental data comprise four 2D time-lapse cell microscopic videos: the first video depicts a culture of HeLa cells (Figure 3.4), the second video shows a culture



(a) Simulation 1 **(b)** Simulation 2

(c) Simulation 3 **(d)** Simulation 4

Figure 3.2: Frame representation of 4 simulations from the first simulated dataset

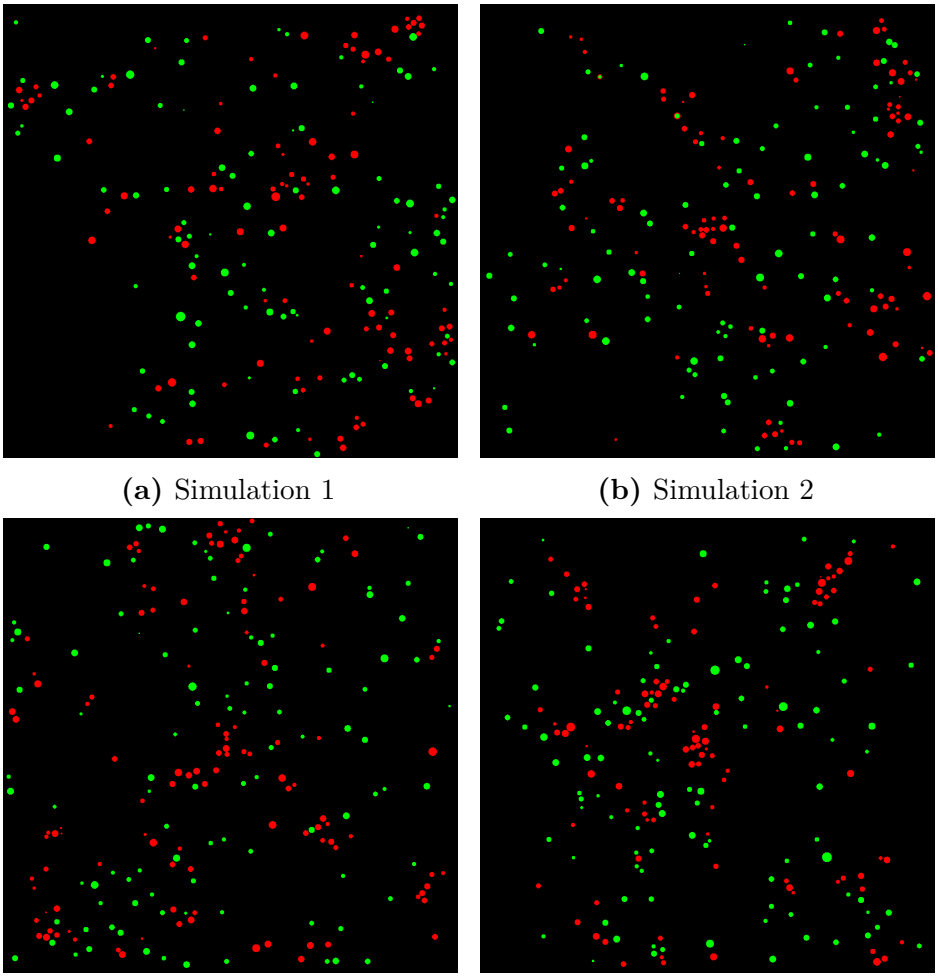
of fibroblast cells (Figure 3.5), the third video presents a culture of perKO cells (Figure 3.6), and the fourth video illustrates a culture containing Human Dermal Fibroblasts (HDF) and Human Umbilical Vein Endothelial Cells (HUVECs), denoted as HDF-HUVECs (Figure 3.7). The HeLa and fibroblast cell cultures each consist of 225 frames, whereas the perKO cell culture contains 179 frames. Each frame in these videos has a resolution of 3840×2748 pixels, pitch $1.67\,\mu$m. The HDF-HUVECs culture consists of 130 frames, with each frame having a resolution of 11530×6940 pixels, pitch $1.67\,\mu$m.

HeLa cells

HeLa cells, derived from Henrietta Lacks' cervical cancer tissue in 1951, were the first human cells to grow and divide indefinitely in a lab, earning them the label *immortal*. Their unique immortality made HeLa cells the preferred choice for biomedical research worldwide [113].

Fibroblast cells

Fibroblasts are cells that form connective tissue, supporting and connecting other tissues and organs. They secrete collagen to maintain tissue structure and are crucial



(a) Simulation 1 **(b)** Simulation 2

(c) Simulation 3 **(d)** Simulation 4

Figure 3.3: Frame representation of 4 simulations from the second simulated dataset with two cell types. Red and green colors are used to distinguish between cell types.

for wound healing. Fibroblasts can be cultured from a person's skin biopsy for genetic and scientific research [132].

The two cell cultures described above, HeLa and fibroblast cells, will be utilized to evaluate the first prediction task. Specifically, a Graph 3D will be constructed for each cell culture, and node features will be extracted from each spatio-temporal graph structure. Subsequently, the Graph 3D-based node features will be merged, and a clustering method, Gaussian Mixture Models (GMM), will be applied to this dataset to try to separate the two cell cultures. In addition, Graph 3D-based node features will be utilized to determine whether these features enhance the performance of separating these two cell cultures using a recent unsupervised method, the He2CL algorithm [188]. This method is employed to analyze cellular diversity based on cell morpho-dynamic behaviors.

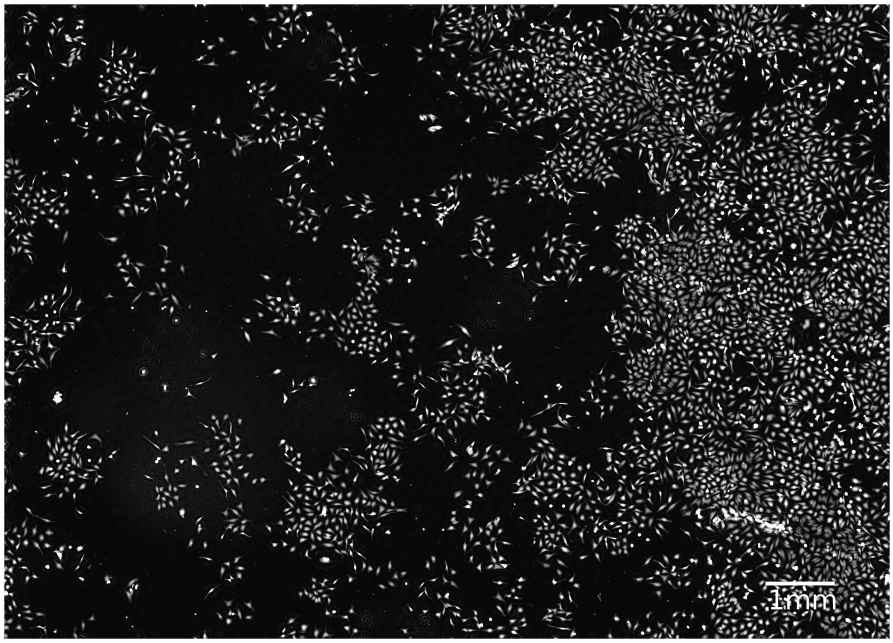


Figure 3.4: Lens-free image of HeLa cells

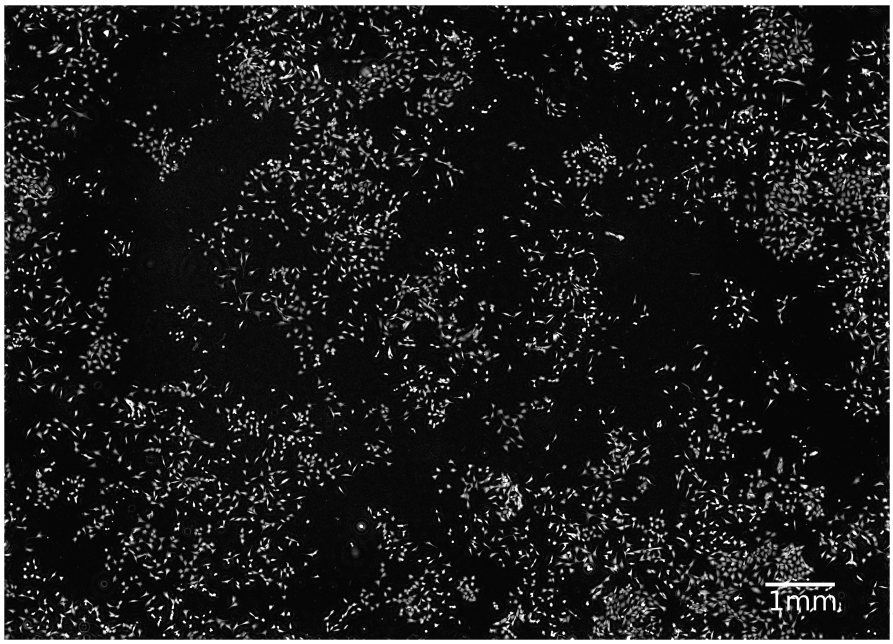


Figure 3.5: Lens-free image of Fibroblast cells

PerKO cells

PerKO (Peroxisome[1] Knockout) cells are genetically engineered mammalian cell lines in which one or more PEX (peroxin) genes, critical for peroxisome biogenesis[2], have been deliberately knocked out (KO). This can lead to disrupt cell-cycle progression and cellular proliferation [43, 218]. In this study, PEX genes were knocked out in fibroblast cell cultures. From this point forward, *perKO* cells will refer to these genetically modified cells.

The perKO cell culture will also be utilized to assess the performance of Graph 3D-based node features in an unsupervised classification task (clustering) using the He2CL algorithm. The objective is to identify and classify cells into clusters that exhibit specific behaviors within the cell culture. Cells exhibiting such behaviors may be cancerous, and the goal is to detect them for subsequent genomic analysis. However, this genomic analysis is beyond the scope of this PhD research.



Figure 3.6: Lens-free image of PerKO cells

HDF-HUVECs

HDF (Human Dermal Fibroblasts) are primary cells derived from normal human skin. These cells are one of the most important architects of cutaneous would healing [260]. HDF generate and maintain connective tissue, and produce proteins for the extracellular matrix that joins together the dermis and the epidermis [241].

HUVECs are primary cells derived from the vein of the umbilical cord. They have served as a crucial model system for investigating the regulation of endothelial cell function and the role of the endothelium in the blood vessel wall's response to stretch, shear forces, and the development of atherosclerotic plaques and angiogenesis [209].

[1]**Peroxisomes** are small and single membrane-delimited organelles that execute numerous metabolic reactions [141].

[2]**Peroxisome biogenesis** encompasses the processes of peroxisome membrane formation, import of matrix proteins, and proliferation and inheritance of the organelle [79].

The HDF-HUVECs culture comprises HUVECs and HDF. The HUVECs were transfected with Green Fluorescent Protein (GFP)-lentiviral particles to allow their identification. This fluorescent labeling enables the establishment of a reference, thereby allowing the differentiation of the two cell types. This cell culture will be utilized to assess the performance of Graph 3D-based node features for the second prediction task, which involves the identification of the two cell types within the culture. This evaluation will be conducted using a supervised Graph Neural Network (GNN)-based model, which is described later.

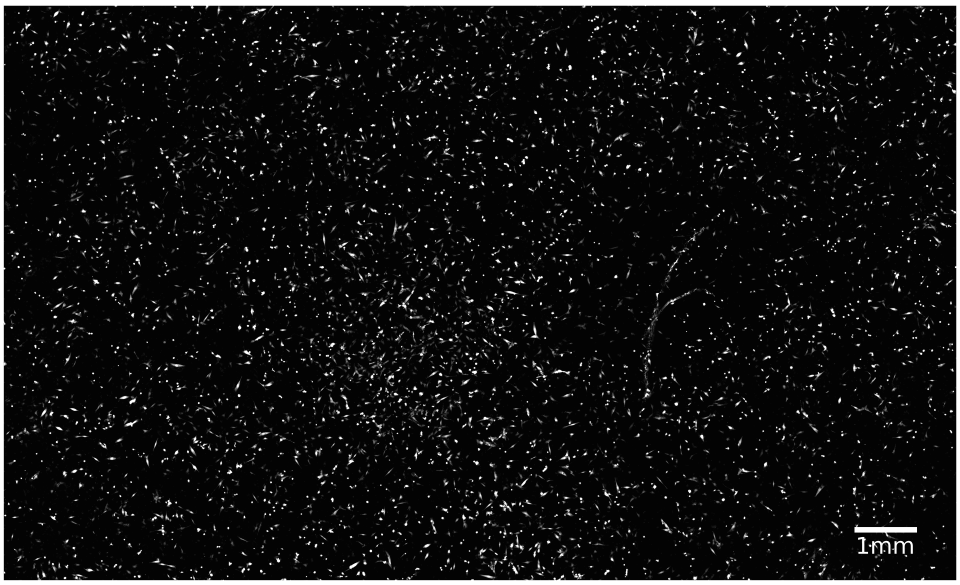


Figure 3.7: Lens-free image of HUVECs and HDF

3.3.3 Graph 3D

Graph 3D (2D position + time) is described in the following manner. For $t = 0, \ldots, T$ (with T representing the last time-step of the time-lapse video), spatial edges are computed at current time-step, while temporal edges are calculated between time-steps t and $t + 1$. Spatial edges represent interactions between cells in the same time-step, such as cell-cell adhesion, and proximity. Temporal edges, calculated based on the proximity of cells between consecutive time steps, represent the relationships and changes over time. They encode the movement, dynamic processes, and trajectories of the cells. Figure 3.8 presents a Graph 3D generated over 5 frames for the simulated data. This figure illustrates the spatial (blue) and temporal (green) connections built within each frame and between them.

Spatial and temporal edges are computed utilizing graphs derived from graph theory. The x-y positions of each node are employed, and threshold values are fixed to remove some distant connections. Different graph structures, which were introduced in Chapter 1, are considered. For spatial connections, Delaunay, Gabriel, Urquhart, and Distance graphs were utilized, with a threshold value τ_{sp} fixed to remove distant links. For establishing temporal connections between two consecutive (or non-consecutive) frames, the Distance graph was utilized. A **temporal threshold** value τ_{temp} was employed to link nodes from one frame to either themselves or their neighboring nodes in the subsequent frame. This linkage was determined based on the spatial proximity between cells belonging to two consecutive frames.

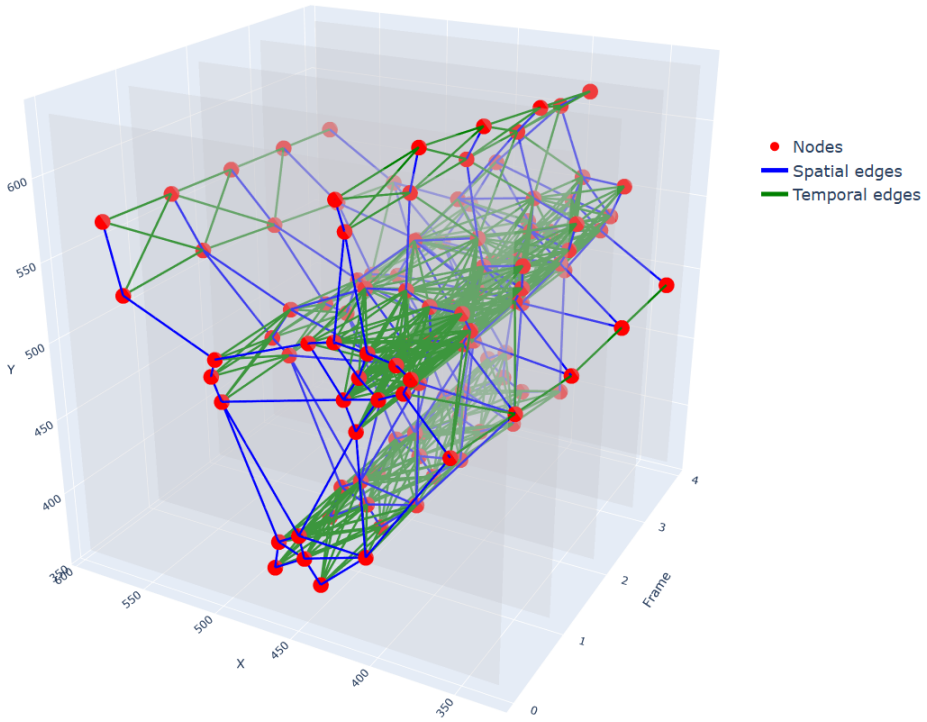


Figure 3.8: An example of Graph 3D over 5 frames. Each light-gray 2D plane corresponds to a time-step (or frame) of the images. Nodes, represented by red points, can be connected based on their spatial proximity, as indicated by blue edges. Temporal connections can be inferred by linking nodes with their close neighbors or with themselves through proximity across consecutive frames.

3.3.3.1 Unsupervised classification

For the unsupervised classification or clustering, node features were computed from the structure of the Graph 3D. These features are introduced below.

Node features

Upon obtaining the graph structure, the node and edge features are calculated. As node features, the metrics defined in Section 1.3.2.1 are taken into account: degree, betweenness centrality, closeness centrality, mean shortest path length, mean neighbor degree, and clustering coefficient. Additionally, core number and harmonic centrality are considered as node features for Sections 3.4.1, 3.4.2, 3.4.3.2 and 3.4.4. These metrics were computed using the functions provided by the Python package NetworkX [102] and some complementary calculations. A brief description of the node features obtained from the graph structure according to the edge type—spatial or temporal—is provided below.

In Graph 3D, nodes exist across multiple time-steps $t = 0, \ldots, T$, with spatial edges representing intra-time-step connections and temporal edges linking nodes across consecutive time-steps. Node features, including degree, mean shortest path length, local clustering coefficient, core number, betweenness centrality, closeness centrality, and harmonic centrality, can be computed based on the spatial edges, the temporal edges, or all the edges simultaneously. For example, the spatial **degree** of a node at time t consists of its connections to other nodes within the same time-step. Its temporal metric represents its connections to neighbouring cells between current

time-step and previous or future time-steps.

The **local clustering coefficient** [279] quantifies the likelihood that two cells interacting with a given cell also interact with each other. Spatially, this is limited to intra-time-step connections, while temporally, it considers inter-time-step relationships. This measure can offer insight into the interconnectedness of a particular cell's neighborhood.

Harmonic centrality [39, 187] is a measure that can indicate the ability with which a cell can communicate with all other cells, even in a disconnected network. It can be computed temporally (across the temporal structure) or spatially (at the same time-step). Temporally, this metric quantifies the influence or importance of a node based on its role in connecting other nodes over time. In this context, harmonic centrality assesses how easily a node can be reached from other nodes within the temporal network [201].

Prior to describing the analysis, several assumptions were made. The temporal clustering coefficient was removed from the analysis due to its consistent valuation of zero, a consequence of the proposed construction of temporal edges. Additionally, the spatial clustering coefficient was omitted for Urquhart graphs due to the inherent properties of their construction, which consistently yield zero values. Furthermore, missing values were replaced with zeros.

3.3.3.2 Supervised classification

In this section, the procedure for supervised classification is described. Initially, a Support Vector Machine (SVM) [214] is applied to distinguish between two different simulations when they are mixed together. Additionally, this supervised method is employed on a simulation containing two cell types. However, as will be demonstrated later, the SVM method does not perform well in this context. Consequently, it is proposed to incorporate the behavior of nodes and their neighbors, as indicated by the graph structure, alongside the already calculated node features. This approach is facilitated by GNN models, which take into account the graph structure in their learning process. The effectiveness of this approach in enhancing performance for this task will be evaluated. Therefore, a GNN-based model is proposed to address the classification of cell types within the same dataset, which will be presented below.

Model architecture

To address the new supervised classification task, a GNN-based model is proposed, as illustrated in Figure 3.9. This model utilizes an existing GNN layer. The general approach is as follows: First, an input graph, Graph 3D $\boldsymbol{G_{3D}}$, along with its node features matrix $\boldsymbol{X}$, is taken. The graph is then divided into two structures based on edge type: spatial ($\boldsymbol{E_{sp}}$) and temporal ($\boldsymbol{E_{temp}}$) edges. Additionally, the edge features matrices are defined: spatial ($\boldsymbol{X_{E,sp}}$) and temporal ($\boldsymbol{X_{E,temp}}$) corresponding to each structure, respectively. Next, a GNN model is applied to the node features of each structure to obtain the encoded features: spatial ($\boldsymbol{Z_{sp}}$) and temporal ($\boldsymbol{Z_{temp}}$). Following this, the encoded features ($\boldsymbol{Z_{sp}}$ and $\boldsymbol{Z_{temp}}$) are concatenated. Subsequently, a Multi-Layer Perceptron (MLP) classifier is employed to produce an output value (probability) between 0 and 1.

The GNN encoder (see Figure 6.19) is designed to encode graph-structured data by integrating node and edge features into meaningful latent representations. This

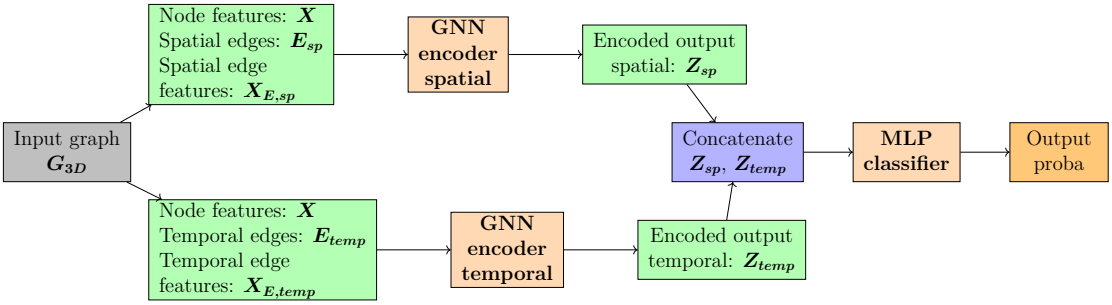


Figure 3.9: GNN-based workflow for supervised classification

encoder structure is based on the GENeralized Graph Convolution (GENConv) layer [158], which consists of a generalized aggregation function suited for graph convolutions.

In each GNN encoder, node features are initially embedded into a 32-dimensional latent space. This operation is followed by a ReLU activation to introduce non-linearity. Then, the GNN layers consist of a GENConv layer to aggregate neighborhood and edge information, and a linear layer to combine the initial and processed features. ReLU activation functions are applied after each transformation in each GNN layer. During the forward pass, node features are transformed, processed through graph convolutions, and concatenated with intermediate features before producing the final output. This design enables flexible and efficient representation of graph data, leveraging both local node attributes and edge relationships.

The MLP classifier is a neural network with three hidden layers (32, 16, 16). Each hidden layer includes a fully connected (Linear) layer, a ReLU activation function, batch normalization (BatchNorm1d) [133] for stabilizing and accelerating training, and dropout for regularization. The output layer is a fully connected layer that maps the last hidden layer to the output size. Weights for Linear layers are initialized using Xavier Normal Initialization [97] to improve convergence, and biases are initialized to zeros.

Model training

To train the proposed model, the Binary Cross-Entropy (BCE) is used as the loss function, which is defined as

$$BCE(\boldsymbol{u}, \hat{\boldsymbol{u}}) = -\frac{1}{N}\sum_{i=1}^{N}[u_i \log \hat{u}_i + (1-u_i)\log(1-\hat{u}_i)],$$

where N corresponds to the total number of nodes over all images, $\boldsymbol{u} = (u_1, \ldots, u_N)$ is the ground truth binary class, and $\hat{\boldsymbol{u}} = (\hat{u}_1, \ldots, \hat{u}_N)$ is the predicted class from the deep-learning model.

AdamW [177] is chosen as optimizer algorithm during training process with a learning rate (LR) of 0.001 and weight decay of 0.0005. Furthermore, an exponential learning rate (LR) scheduler is applied to enhance the update of model parameters during the back-propagation phase. During each training epoch, graph batches of size 2 are taken from the total training data. The training phase is halted by defining an early stopping criterion: if the validation loss does not improve over 10 consecutive epochs, the model stops training. Finally, the model is trained over 100 epochs, and the best model is retained based on the lowest validation loss achieved.

The input data for the model is defined below, following the computation of different Graph 3D structures and their node features. The node features matrix $\boldsymbol{X}$ is defined, which is composed of Graph 3D-based node features and the x-y position of each cell. Additionally, edge features are computed both spatially and temporally. These features include edge betweenness centrality, Euclidean distance, relative displacement, and cosine similarity (see Section 1.3.2.2) for each pair of nodes that form a link. The edge features matrices are then defined as spatial ($\boldsymbol{X}_{E,spatio}$) and temporal ($\boldsymbol{X}_{E,temp}$) corresponding to the features mentioned above.

3.4 Results and Discussion

3.4.1 HeLa and fibroblast cells

In this study, the analysis begins by considering two independent cultures, each containing a unique type of cell. This process yields two series of microscope images and, consequently, two graphs. The features extracted from these graphs are then combined, and an attempt is made to separate them.

Experimental data consist of two 2D time-lapse cell microscopic videos: the first one is a culture of HeLa cells and the second one is a culture of fibroblast cells, each consisting of 225 frames with a resolution per image of 3840×2748 pixels, pitch $1.67\,\mu$m. To mitigate the high computational and memory costs associated with calculating node features, each frame image was divided into four equal quadrants based on x-y pixel positions. Furthermore, the time-lapse videos were partitioned into four equal time intervals to address the aforementioned computational constraints. In the end, the results for the first 150 frames were retained, as HeLa and fibroblast cells exhibited distinct migratory patterns during this time interval in their respective time-lapse videos.

Furthermore, the following graph structures have been considered:

- **Delaunay:** with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} \in \{10\,\mu\text{m}, 20\,\mu\text{m}, 30\,\mu\text{m}, 50\,\mu\text{m}, 100\,\mu\text{m}\}$.
- **Gabriel:** with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} \in \{10\,\mu\text{m}, 20\,\mu\text{m}, 30\,\mu\text{m}, 50\,\mu\text{m}, 100\,\mu\text{m}\}$.
- **Urquhart:** with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} \in \{10\,\mu\text{m}, 20\,\mu\text{m}, 30\,\mu\text{m}, 50\,\mu\text{m}, 100\,\mu\text{m}\}$.
- **Distance:** with $\tau_{sp}, \tau_{temp} \in \{10\,\mu\text{m}, 20\,\mu\text{m}, 30\,\mu\text{m}, 50\,\mu\text{m}, 100\,\mu\text{m}\}$.

The experimental data presented herein are employed to evaluate the first prediction task proposed in Section 3.3.1. Specifically, Graph 3D is computed using the data provided through their x-y positions and the graph structures proposed for HeLa and fibroblast cultures. Subsequently, spatial and temporal metrics are calculated from these graph structures at the node level for each cell culture. These node features, previously computed for each cell culture, are then artificially combined to create a new dataset. Finally, a clustering method is applied to the node features of this dataset to identify HeLa and fibroblast cells.

Node features derived from Graph 3D provide crucial information for distinguishing each cell type originating from the two cell cultures. This is evidenced by the histograms of each spatial and temporal node feature, computed through a

given graph structure for HeLa and fibroblast cells, as presented in Annex 6.3.1 (see Figures 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, and 6.10). In this example, it is observed that all node features exhibit visually distinct distributions, aiding in the differentiation of HeLa and fibroblast cells. These results suggest that these metrics could be valuable in separating the two cell cultures through the methodology proposed for the unsupervised classification method.

Before presenting the results, it is emphasized that the Gaussian Mixture Model (GMM) was employed as the clustering approach to test the methodology previously described. The mean accuracy score (over 10 runs) of GMM for all graphs was calculated, and the maximum value from the obtained scores was selected.

In addition to the GMM chosen for the unsupervised classification method, a comparison of its performance with a recent unsupervised approach called He2CL (HEterogeneity detection in cell cultures with a 2-step CLustering algorithm) is proposed [188]. This method was specifically designed for these datasets as it analyzes cellular diversity based on cell morpho-dynamic behaviors. The He2CL algorithm can be briefly described in the following way. A first clustering method assigns a cluster label to each cell and time-step throughout the time-lapse sequence. These labels are then used to compute a representation of the time-series data associated with each cell trajectory [188]. Subsequently, a second clustering method is applied to this representation to assign a cluster group to each cell trajectory [188].

Therefore, the objective for this study is to analyze and determine whether Graph 3D-based node features can enhance the performance of both methods (GMM and He2CL) beyond relying solely on cell morpho-dynamic features. For the He2CL method, features derived from Graph 3D will be utilized only in the first step of the algorithm to determine whether this configuration performs equally well or better than the complete method, which utilizes tracking information in the second step.

Table 3.2 presents the accuracy scores (in %) for the GMM and He2CL methods, each evaluated with different feature combinations over the first 150 frames. **Morpho** refers to cell morphological features, such as: dry mass, thickness, perimeter, area, and eccentricity. **Morpho/Dyn** refers the usage of cell morphological features, and cell dynamic features related to the tracking, including cell speed, dry mass growth and cell age. **Only Graph** means the utilization of only Graph 3D-based node features. **Graph + Morpho** means the use of Graph 3D-based node features (previously described) and cell morphological features. Furthermore, in each row, the first value represents overall accuracy, while the second value, preceded by F: (H:), indicates accuracy specifically for Fibroblast (HeLa) cells.

In the He2CL method, the accuracy scores derived from the second clustering step are shown for the Morpho and Morpho/Dyn features. For the remaining features derived from Graph 3D, the accuracy scores correspond to the first clustering step applied.

Regarding the results presented in Table 3.2, the GMM performs better using only morphological features than using morphological and dynamic features. This implies that dynamic features are not relevant for this method and contribute nothing to its performance. In contrast, the He2CL method outperforms GMM for both morphological and morphological-dynamic features, highlighting the importance of cell dynamic features in identifying cell types. These features capture the dynamic behavior of cells, which is utilized by the He2CL algorithm during its second clustering step.

The incorporation of Graph 3D-based node features enhances the scores obtained

Table 3.2: Comparison of GMM and He2CL algorithms with different feature combinations. In the table, the highest overall accuracy score for each method is highlighted in black. The highest overall accuracy score derived from applying the two clustering steps for the He2CL algorithm is highlighted in red.

	Method	
Features	**GMM**	**He2CL**
Morpho	68.7 (F:60.2, H:77.6)	80.9 (F:79.7, H:79.8)
Morpho/Dyn	51.1 (F:43.1, H:60.7)	83.9 (F:79, H:89)
Only Graph	78.5 (F:75.6, H:82.1)	83.9 (F:85, H:83)
Graph + Morpho	**82.3 (F:74.1, H:88.4)**	**90.4 (F:96, H:85)**

by considering cell morphological or morphological-dynamic characteristics for both methods. This underscores the importance of these metrics, which are derived from the spatio-temporal graph structure constructed. Furthermore, it is observed that by integrating Graph 3D-based node features with cell morphological features, both methods achieve higher scores than when using only node features. This highlights the complementarity between the metrics derived from cell morphology and those derived from Graph 3D.

Notably, the best performance for the GMM is achieved by considering Graph 3D-based node and cell morphological features. This implies that the dynamic characteristics of cells, as derived from their tracking, are not entirely useful for this method in distinguishing between the two cell types. An identical pattern is observed when the He2CL method is applied. In this case, the He2CL method also achieves its best performance by utilizing Graph 3D-based node and cell morphological features, suggesting that cell dynamic features derived from tracking may not be necessary for identifying the two cell types.

It is important to note that the accuracy scores obtained for the He2CL method using cell morpho or morpho-dynamic features were computed through the application of the two clustering steps. In contrast, the accuracy scores derived from incorporating Graph 3D-based node features were computed using only the first clustering step. This outcome suggests that the He2CL algorithm does not necessarily require the second clustering step, as the tracking information of cells, used for this step, may be fully captured by the temporal structure of Graph 3D and represented by its temporal node features.

By plotting and identifying each cluster (obtained using the He2CL algorithm) across all frames for HeLa and Fibroblast data (as illustrated in Figures 3.10, 3.11, 3.12, 3.13, 3.14, and 3.15), a cluster demonstrating a higher cell concentration than the others is observed from the beginning of the time-lapse video until frame 150 (used as a reference) for the HeLa data. Additionally, this behavior is explicitly noticed at frame 150 for the Fibroblast data. This phenomenon (or potential bias) can be attributed to the fact that two different graph structures are obtained for each time series of microscopic images. These structures generate node features that a priori represent the global information about the graph topology, making it easier to distinguish between the two cell cultures. Consequently, the method works well and outperforms the classic He2CL algorithm, which involves two steps.

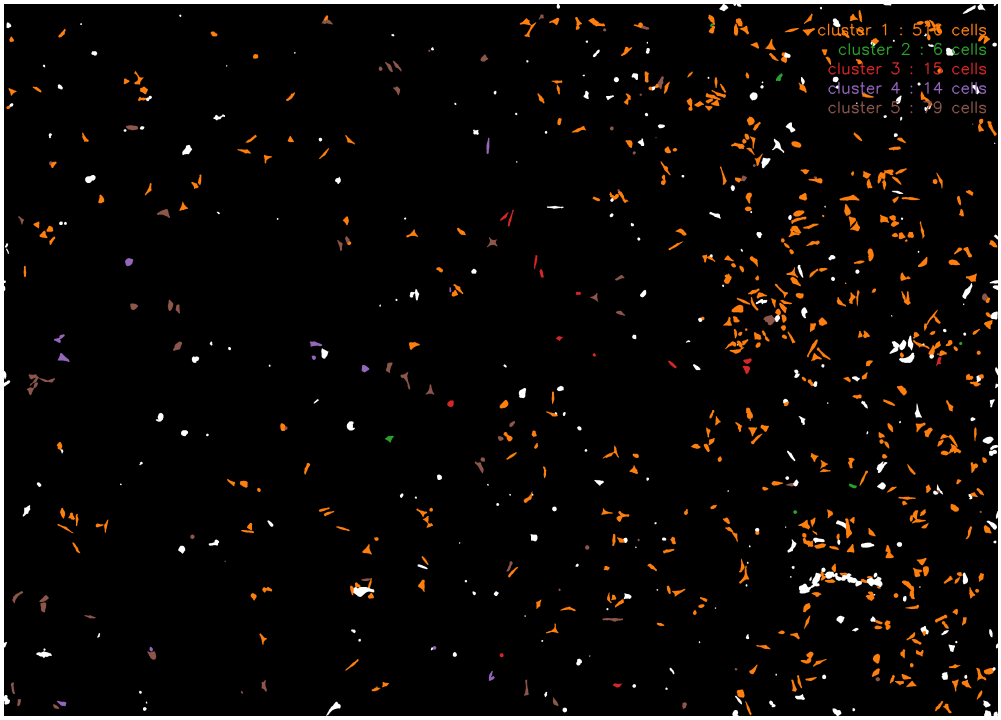


Figure 3.10: Clusters on HeLa cells over frame 25

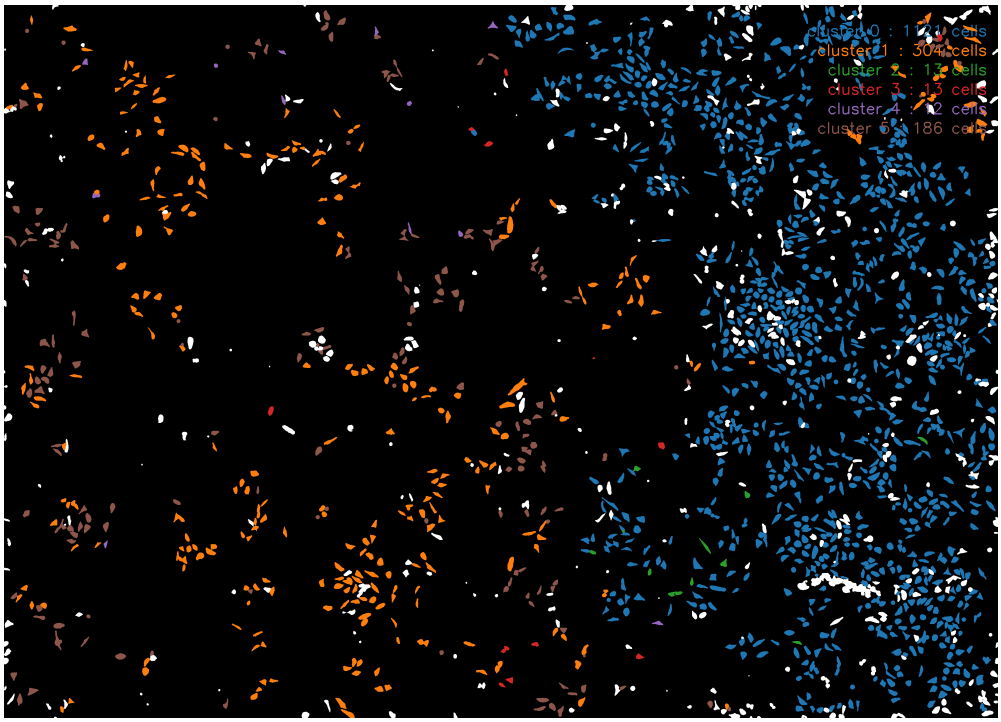


Figure 3.11: Clusters on HeLa cells over frame 125

Figures 3.16 and 3.17 present radar plots that characterize the cluster mean values relative to the overall population mean for each cell morpho-dynamic feature and graph-based node feature utilized in the He2CL method (first clustering step) for HeLa and fibroblast cell cultures. The application of this method enabled the identification of six clusters. It is important to note that in these figures, **path len corr** refers to the mean shortest path. **Cycle length**, the time it takes one cell to divide into two daughter cells, is defined as the number of time-steps in the cell

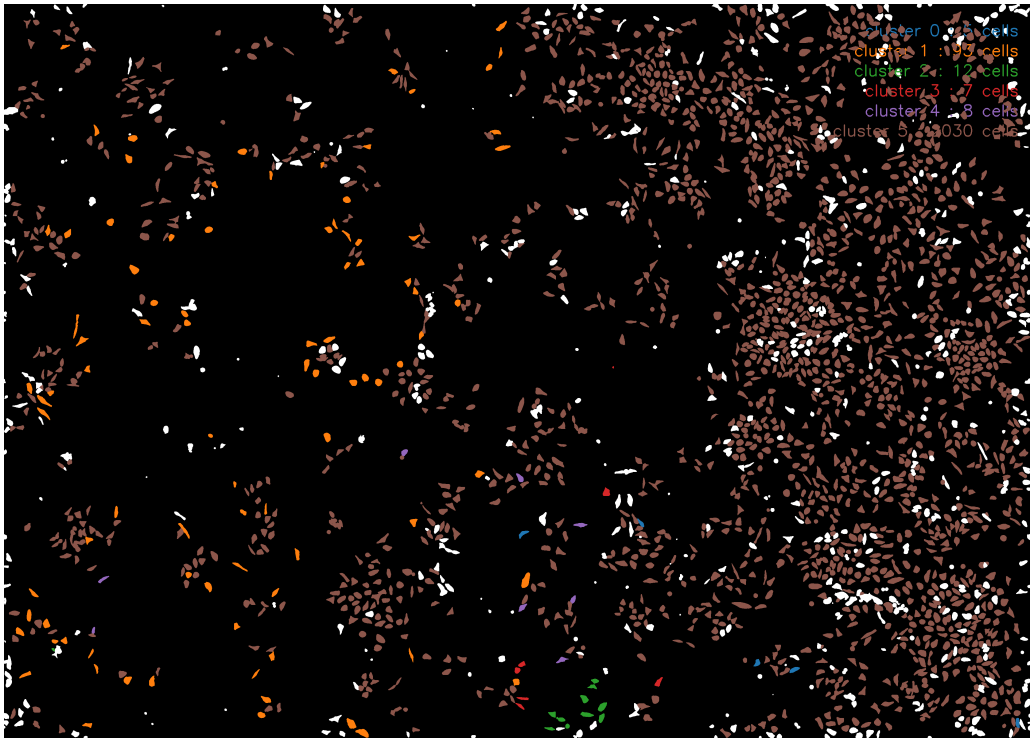


Figure 3.12: Clusters on HeLa cells over frame 150

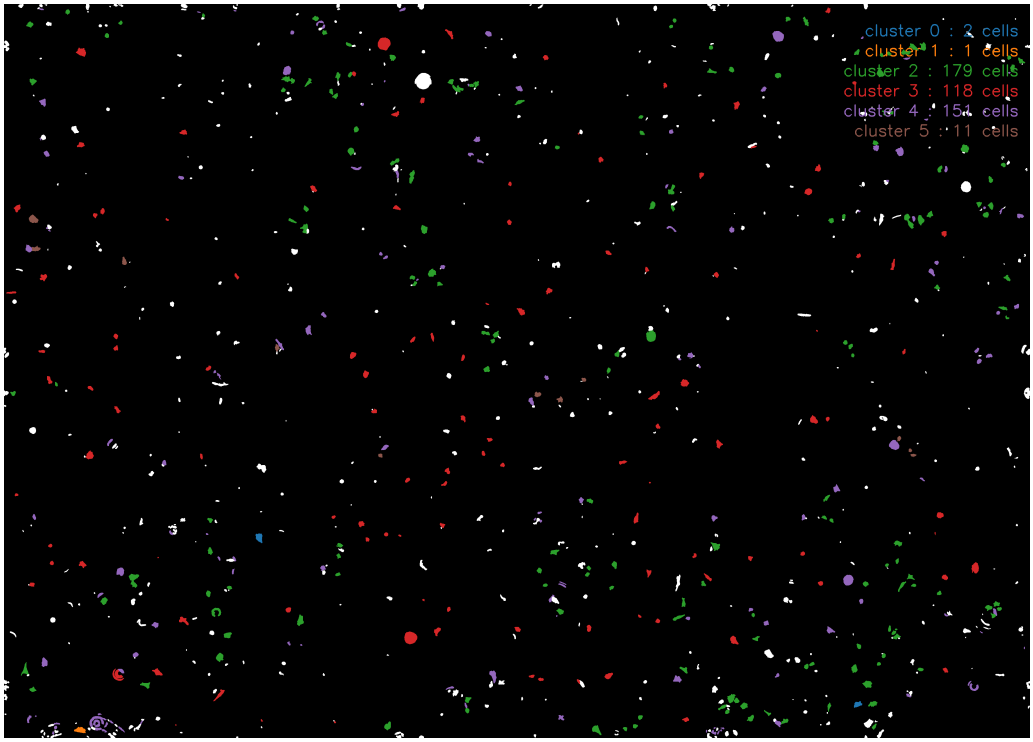


Figure 3.13: Clusters on Fibroblast cells over frame 31

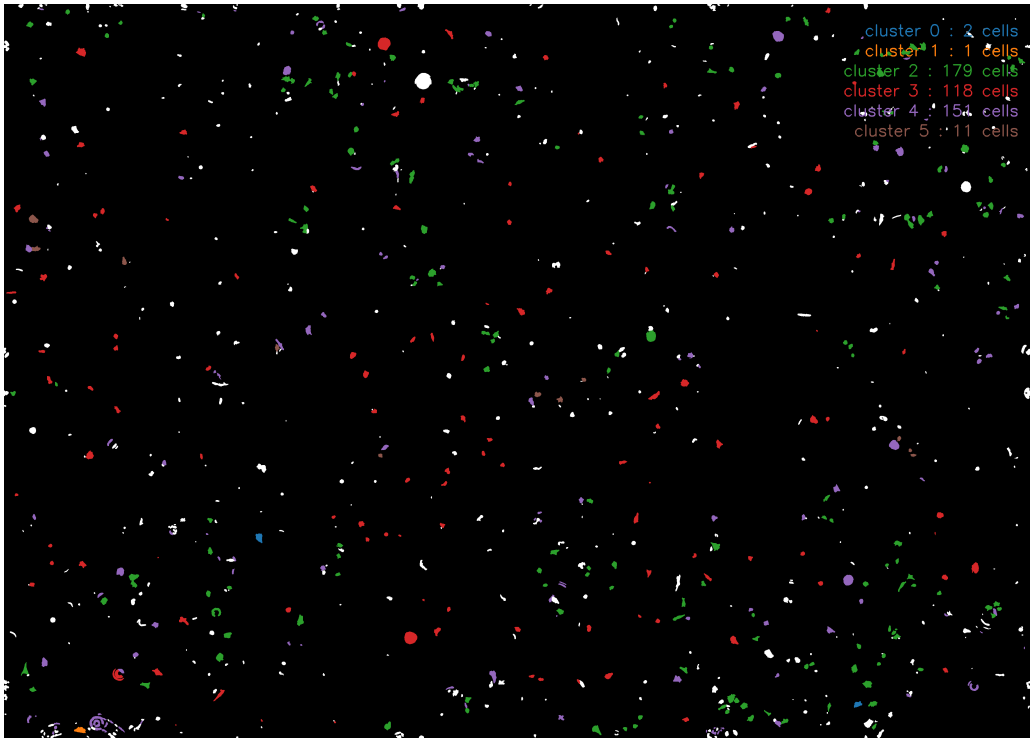


Figure 3.14: Clusters on Fibroblast cells over frame 125

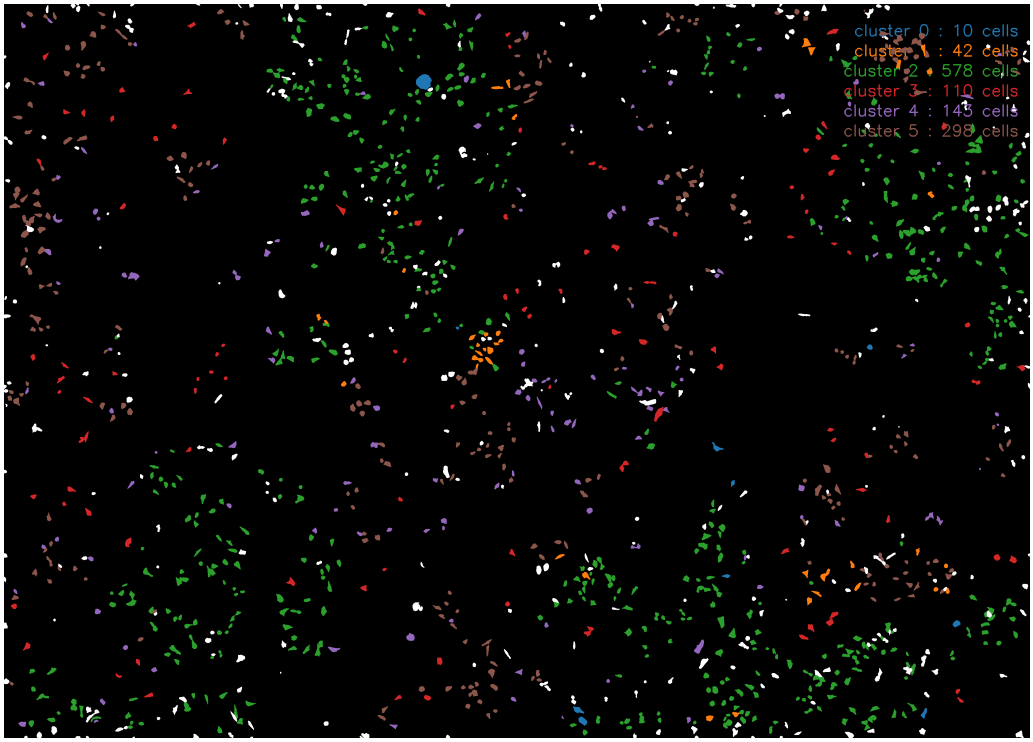


Figure 3.15: Clusters on Fibroblast cells over frame 150

trajectory multiplied by the rate, which is the time interval between two consecutive time-steps in the time-lapse video. **Growth** (%) is measured as the difference in cell dry mass between two consecutive time-steps. **Cycle growth coeff** represents the trend in dry mass growth over each cell trajectory and is computed as the slope of the affine function fitted to the cell trajectory dry mass values.

Regarding the cell morpho-dynamic features, it is observed that clusters 0 (blue) and 1 (orange) comprise big cells with dry mass, area, perimeter, and growth that exceed the population mean. Additionally, these cells exhibit slight differences in cycle length and thickness. Cluster 5 (brown) contains the second-biggest cells, which have cycle lengths longer than the population mean. Clusters 2 (green) and 3 (red) consist of smaller cells that exhibit the highest speed. These cells share similar morphological characteristics, with slight variations in area, perimeter, and growth. Cluster 4 (purple) comprises a distinct population of cells that tend to be small, with very low dry mass, low thickness, and low growth.

For the remainder, the HeLa algorithm processes time-lapse sequences in two main steps. The first step involves labeling each cell and time-step, then using these labels to create a time-series data representation for each cell trajectory [188]. The second step assigns cluster groups to each cell trajectory based on this representation [188]. In summary, all clusters exhibit properties related to cell morpho-dynamic features, with no substantial differences among them. This suggests that a second clustering step may be necessary to visualize explicit differences in the clusters with respect to cell morphological and dynamic behaviors.

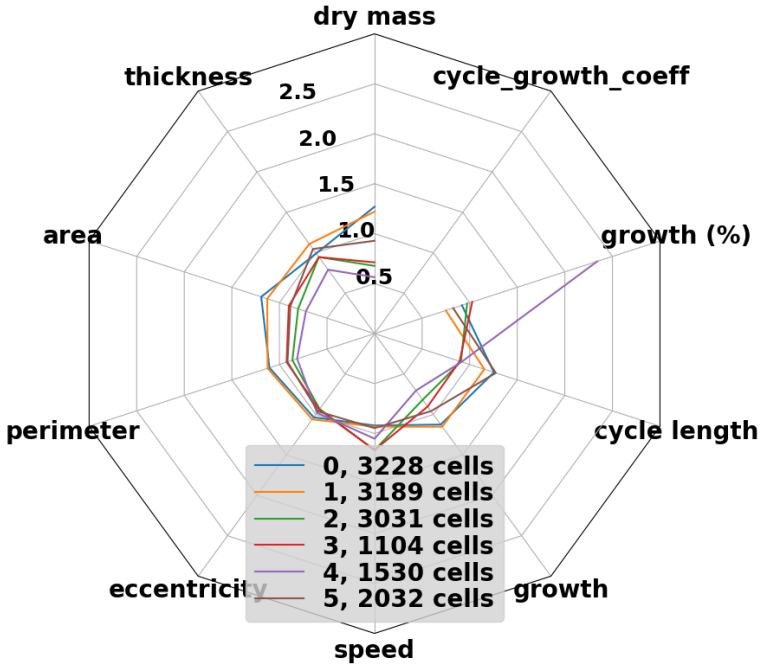


Figure 3.16: Radar plot for cell morpho-dynamic features per cluster on mixing HeLa and fibroblast cells

Regarding the Graph 3D-based node features in Figure 3.17, it is observed that Cluster 0 (blue) comprises cells with the highest temporal harmonic and closeness centrality, along with high spatial clustering coefficient and core number, but low temporal mean shortest path. Cluster 1 (orange) includes cells with the highest spatial betweenness and closeness centrality, along with low temporal closeness

centrality, mean shortest path, and harmonic centrality. Cluster 4 (purple) consists of cells with high spatial clustering coefficient and temporal mean shortest path, but low values for the other node features. Clusters 2 (green) and 5 (brown) encompass cells with similar spatial clustering coefficient, closeness centrality, and core number, but exhibit significant differences in temporal closeness centrality, mean shortest path, harmonic centrality, and spatial betweenness centrality. Cluster 3 (red) comprises a unique population of cells that tend to have high temporal mean shortest path and very low values for the other node features.

In summary, it is noted that Cluster 0 is characterized by temporal proximity, Clusters 2 and 4 depend on both spatial and temporal proximities, Clusters 1 and 5 are influenced by spatial proximity, and Cluster 3 is defined by temporal proximity, particularly described by the shortest paths. These outcomes demonstrate that clusters characterized by temporal proximities may contain cells with their tracking information. In contrast, clusters characterized by both spatial and temporal proximities can be able to differentiate between the two cell cultures through their displacement patterns and identify their respective tracks.

It is also evident that the most critical node metrics used to distinguish among all clusters are those characterized by spatial and temporal connectivity within the network, even though degree was not a relevant feature in this case. This underscores the importance of defining an appropriate spatio-temporal structure for Graph 3D to capture these dynamics effectively.

Furthermore, all clusters exhibit properties with more distinct differences than those observed in the case of cell morpho-dynamic features. This implies that there is no need to apply a second clustering step, as previously supposed, since the tracking information required for this step is captured by the presence of temporal features, as shown in this plot. This improvement enhances the identification of cell types (HeLa and fibroblast) within each cluster.

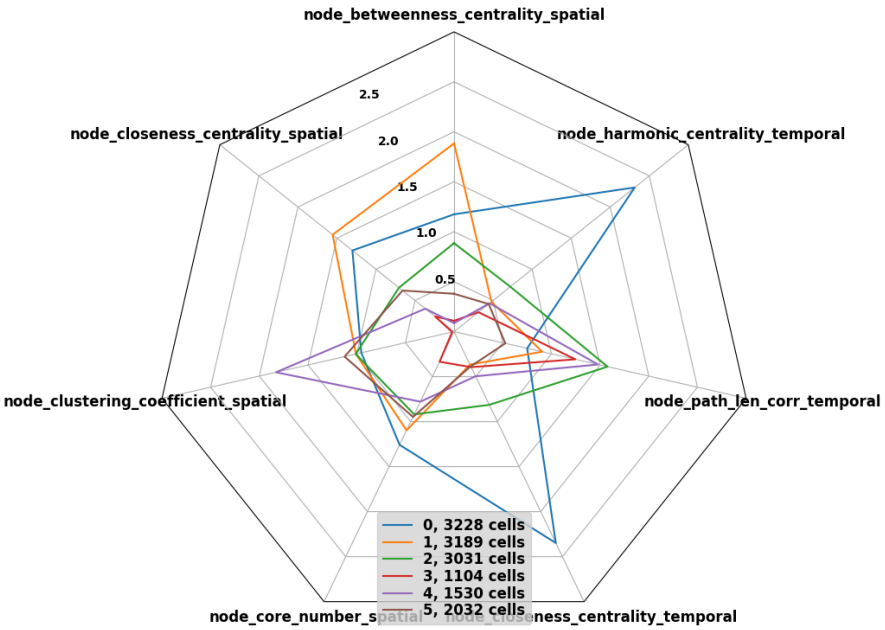


Figure 3.17: Radar plot for Graph 3D-based node features per cluster on mixing HeLa and fibroblast cells

The bias previously observed when plotting the clusters in each cell culture can be attributed to the spatial and temporal proximity of the cells. In this context,

it would be more appropriate to define node features that are not directly related to the cell proximity, such as the calculation of a ranking of nodes based on their links. This approach could help determine whether the bias is reduced by these new metrics.

3.4.2 PerKO cells

In this analysis, the methodology of Graph 3D-based node features, in conjunction with the He2CL algorithm, is applied to the dataset comprising the perKO cell culture (see Figure 3.6). The objective is to identify groups of cells within this culture that can be characterized by their morphological features and/or node features.

The perKO cell culture consists of fibroblast cells in which one or more genes related to peroxisomal function have been knocked out. These cells exhibit unusual behaviors, such as aberrant displacements, and have the potential to become cancerous. The importance of this study lies in the ability to detect these cancerous cells. Therefore, the objective is to determine whether Graph 3D-node features can enhance the performance of the He2CL method in identifying clusters containing only cancerous cells. In this approach, the graph was constructed as before, but a new method was employed. Each image (3840×2748 pixels, pitch 1.67μm) was divided into 4 quadrants, and 20-frame overlapping intervals were selected: $1-20, 6-25, 11-30, \ldots, 161-179$. This method allows for the consideration of previous and subsequent connections of the endpoints of each interval when calculating temporal node features. Finally, a Delaunay graph was built in each overlapping interval with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 50\,\mu$m. This graph structure was chosen because it had shown the best results previously.

The average value within each cluster relative to the overall population mean for each morpho-dynamic cell feature and graph-based node feature in the He2CL approach (first clustering step) is illustrated in Figures 3.18 and 3.19. Initially, the method identified eight clusters, four of which corresponded to anomalies primarily derived from acquisition and computational errors. Consequently, it was decided to exclude these clusters, as they were deemed irrelevant to the analysis.

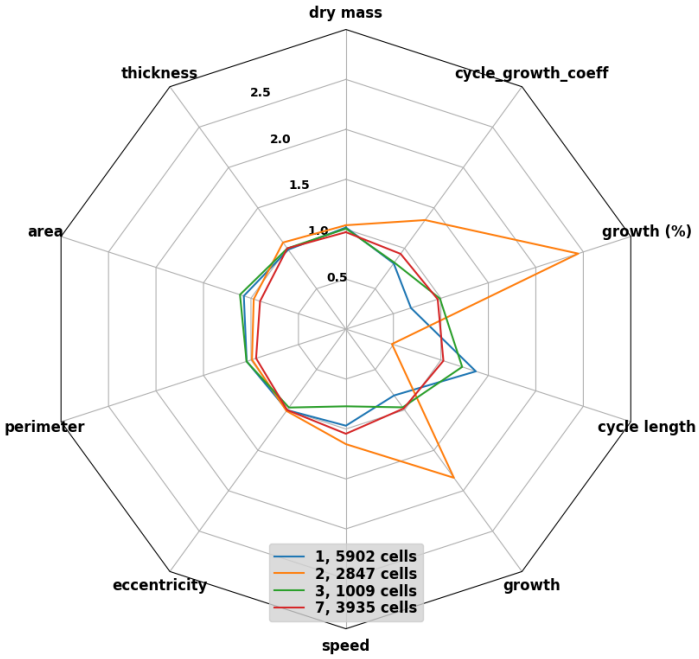


Figure 3.18: Radar plot for cell morpho-dynamic features per cluster on PerKO cells

Regarding the morpho-dynamic features of cells, it is observed that clusters primarily separate cells based on their growth rates and cycle lengths. Clusters 1 and 3 comprise the largest cells, with minimal differences in cycle length and growth. Cluster 1 (blue) contains cells with the longest cycle lengths and the lowest growth rates. In contrast, Cluster 3 (green) includes cells with very slow growth. Cluster 2 (orange) consists of smaller cells than the population mean, exhibiting the highest growth rates and very short cycle lengths. Cluster 7 (red) comprises smaller cells with morphological characteristics similar to the population mean. Notably, three of the identified clusters do not provide sufficient evidence to detect or distinguish between possible groups of non-cancerous cells and those with the potential to develop into cancerous cells. However, Cluster 2, characterized by cells with short cycles and rapid growth, could offer a valuable indicator for identifying cancerous cells. Chen et al. [58] demonstrated that cell cycle length serves as a hallmark of cancer susceptibility, regardless of the organ, cancer type, cancer-driving mutation, or the timing within the lifespan when the mutation occurred.

Regarding the Graph 3D-node features in Figure 3.19, it is observed that clusters separate cells by considering all spatial and temporal node features described in the radar plot. Cluster 7 comprises cells with the highest values for all spatial and temporal node metrics. Cluster 2 contains cells with the second-highest values for all spatial and temporal node metrics. Cluster 1 includes cells with the third-highest values for all spatial and temporal node metrics. Conversely, Cluster 7 also encompasses cells with the lowest values for all node metrics. In brief, it is noted that the Graph 3D-node features do not exhibit explicit differences among all identified clusters to distinguish cells based on their features derived from spatial or temporal proximities. This observation can be attributed to the fact that the method chosen to construct the spatio-temporal structure of Graph 3D may not be sufficient to capture the dynamics of this cell culture, as it is difficult to identify highly isolated cells. It is important to note that this analysis is entirely exploratory, as there is no

definitive proof that cancerous cells are present in this cell culture.

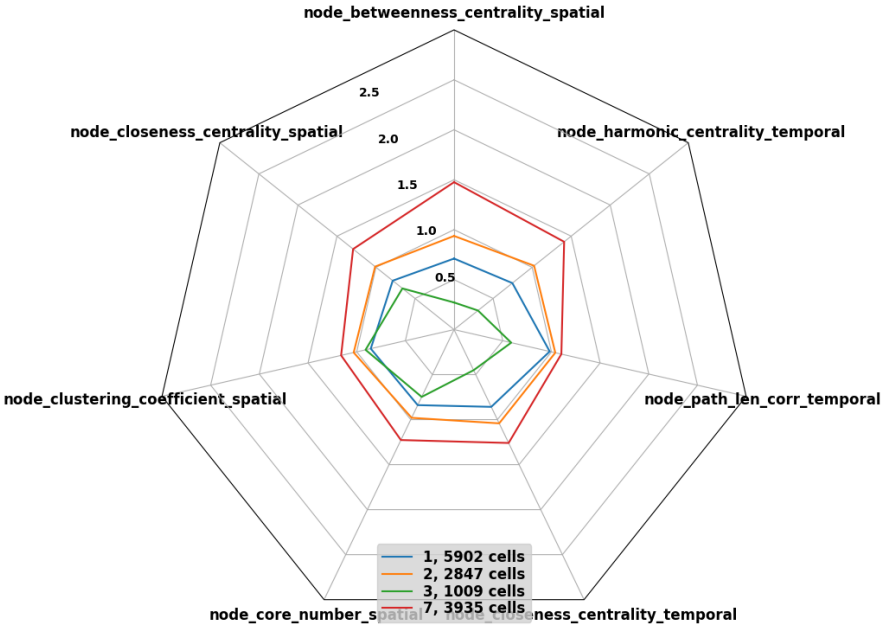


Figure 3.19: Radar plot for Graph 3D-based node features per cluster on PerKO cells

3.4.3 Simulations

The analysis focuses on addressing the two prediction tasks using simulated data. This is achieved by applying the methodology of Graph 3D-based node features in conjunction with both unsupervised and supervised methods. For the first prediction task, which involves identifying cell types from two distinct populations, Gaussian Mixture Models (GMMs) and Support Vector Machines (SVMs) are employed. For the second prediction task, which involves identifying cell types from two mixed populations, GMM, SVM and a Graph Neural Network (GNN)-based model are utilized.

3.4.3.1 Two distinct populations

The first prediction task is addressed through the use of simulations, as it is possible to control and define the rules governing the movements of cells based on their proximity to other cells and their neighborhood. This approach allows for the proposal of distinct simulated conditions. The simulated data are obtained from 2D video simulations of cell cultures. The movement of each cell is determined by its spatial proximity to other cells. Initially, cells move randomly. Over time, they tend to approach neighboring cells, resulting in forces of attraction or repulsion between them.

The first set of simulation data, comprising simulations with a unique type of cell, was utilized for addressing the first prediction task. For this dataset, graph structures were employed to construct their Graph 3D associated and extract the node features. These graph structures are described in detail in Annex 6.2.1.

For each possible combination of two distinct simulations from the first dataset, a Graph 3D was extracted from each series of microscopic images, and the node features were computed. Subsequently, the node features from these two simulations

were combined, and the GMM was applied to these features to attempt to separate the two simulations. The accuracy score between the cluster labels and the simulation labels was then computed. This procedure was repeated ten times, and the mean value of the accuracy scores was calculated. This procedure is repeated ten times, and the mean value is calculated. In Table 6.1 (see Annex 6.2.1), the mean accuracy score obtained for all possible combinations of two distinct simulations across all different graph structures is presented. It is observed that almost all combinations of two distinct simulations can be separated with a score higher than 80%. Furthermore, it is noted that the graph structure **Distance** with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 20\,\mu$m is the most useful for separating two simulations, as it achieved a score of more than 80% (and even exceeding 90% in some instances) across a large number of combinations of two distinct simulations.

To determine the relevance of spatial or temporal aspects in separating two simulations, the previously described procedure is applied separately to spatial and temporal node features, and the results are compared. Table 6.2 (see Annex 6.2.1) presents the best mean accuracy score (defined as the maximum value over all graphs) achieved by the GMM for all possible combinations of node features. The results indicate that temporal features are more significant than spatial features in distinguishing between simulations. In the majority of the combinations of two distinct simulations, the temporal node features achieved scores ranging from 80% to 98% (see Table 3.3). In contrast, the spatial node features obtained scores between 59% and 81%. This finding underscores the importance of the temporal structure in Graph 3D.

Table 3.3: Best mean accuracy score on GMM for all possible node features combinations

Spatial and temporal node features	**Spatial node features**	**Temporal node features**
0.9804	0.6493	0.9780

It is proposed to analyze whether a supervised method will perform better than the GMM for the task of identifying two distinct simulations. Subsequently, a supervised approach using a Support Vector Machine (SVM) [214] is applied to separate the two simulations. 70% of the total data is used to train this classifier, and the remaining 30% is used to test the model's performance. In Table 6.3 (see Annex 6.2.1), the best mean accuracy score (given by its maximum value over all graphs) of the SVM model is observed for all possible combinations of two distinct simulations with regard to the possible combinations of features. The features considered include x-y positions, node (spatial and temporal) features, and x-y positions with node (spatial and temporal) features. Poor performance is noted when only x-y cell positions are considered. When only node (spatial and temporal) features are considered, the best accuracy score is high and close to 1, indicating good performance of this model. Moreover, similar performance is achieved by adding x-y positions to node features (see Table 3.4).

In conclusion, this analysis demonstrated that node features are sufficient for identifying two distinct simulations through both unsupervised and supervised methods. This can be attributed to the fact that temporal node features made the most significant contribution to distinguishing between the two simulated datasets.

Table 3.4: Best mean accuracy score on SVM for all possible combinations of features

x-y position	Node features	x-y position and node features
0.6932	0.9999	0.9999

These features aim to capture temporal connections, which facilitate the identification of cells based on their displacement patterns, as these displacements are partially defined by the cells and their neighborhood. This means that temporal node features derived from Graph 3D were able to identify the dynamics associated with cells and their tracking information through the temporal displacement patterns described by the cells and their neighborhood.

3.4.3.2 Two mixed populations

The results derived from applying both unsupervised and supervised methods to simulated data, aimed at identifying cell types from two mixed populations, are analyzed below.

As introduced previously, the simulated data originate from two-dimensional video simulations of cell cultures. The displacements of each cell type are governed by the rules outlined in Section 3.3.2.1. In this study, the second set of simulated data, comprising simulations with two cell types within the culture, is employed.

Initially, a simulation from this dataset is utilized. As before, different graph structures are employed to construct the Graph 3D and extract the node features. The same graph structures used in separating two distinct simulations are considered.

Before presenting the results for the methods (unsupervised and supervised) applied in this simulation, an analysis of the distribution of each node feature for the two cell classes derived from Graph 3D with respect to all graph structures is provided. Figures 3.20, 3.21, and 3.22 display the histograms obtained for a node feature on a chosen graph structure. It is noted that the distribution of spatial degree does not clearly differ between the two classes, as there appears to be no difference between the histograms of the two classes. The same behavior is observed for spatial degree on the Delaunay graph and spatial betweenness centrality on the Urquhart graph.

To determine whether the difference is significant, the Kolmogorov-Smirnov test is applied to each node feature and graph structure. The null hypothesis for this test is H_0 : the two distributions for the selected node feature are equal. The p-value is computed, and a threshold of 0.05 is used to accept or reject H_0. The results of this test are summarized in Table 6.4 (see Annex 6.2.2.1). In this table, **X** indicates that H_0 is not rejected, meaning a p-value greater than 0.05 was obtained. It is observed that the distribution of the spatial mean shortest path between the two classes shows no difference in most of the considered graph structures.

Graph 3D-based unsupervised approach is first applied on this simulated data for all graph structures considered. Table 6.5 (see Annex 6.2.2.1) presents the mean score (over 10 iterations) on GMM for the simulated data across all possible node feature combinations with respect to the graph structures. It is observed that the method performs poorly across all graphs in this dataset, achieving a maximum score of 58%. This score is attained when using the Distance graph with parameters

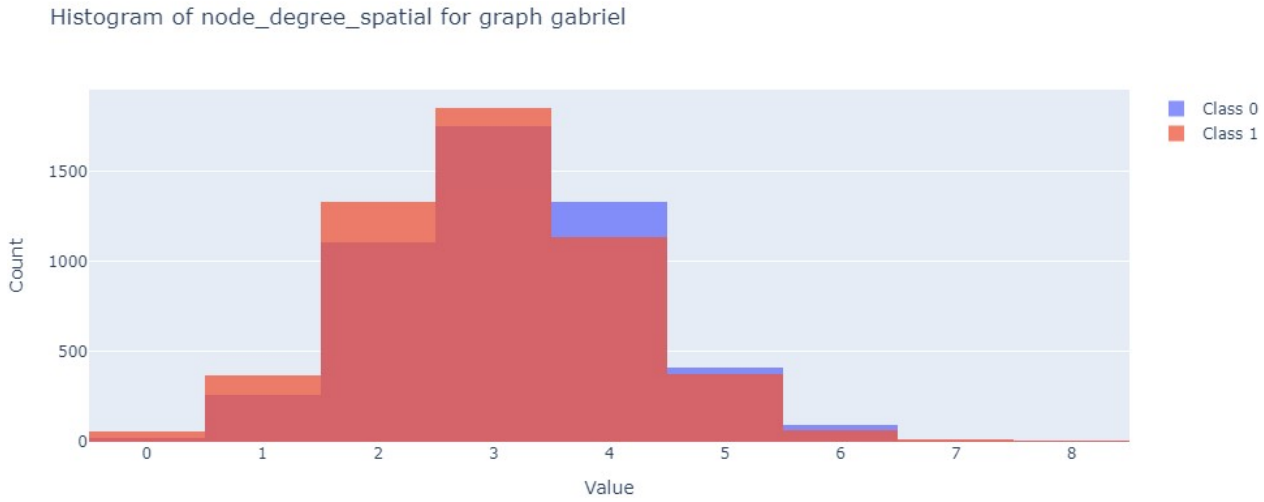


Figure 3.20: Histogram of spatial degree distribution between two cell populations on Gabriel graph

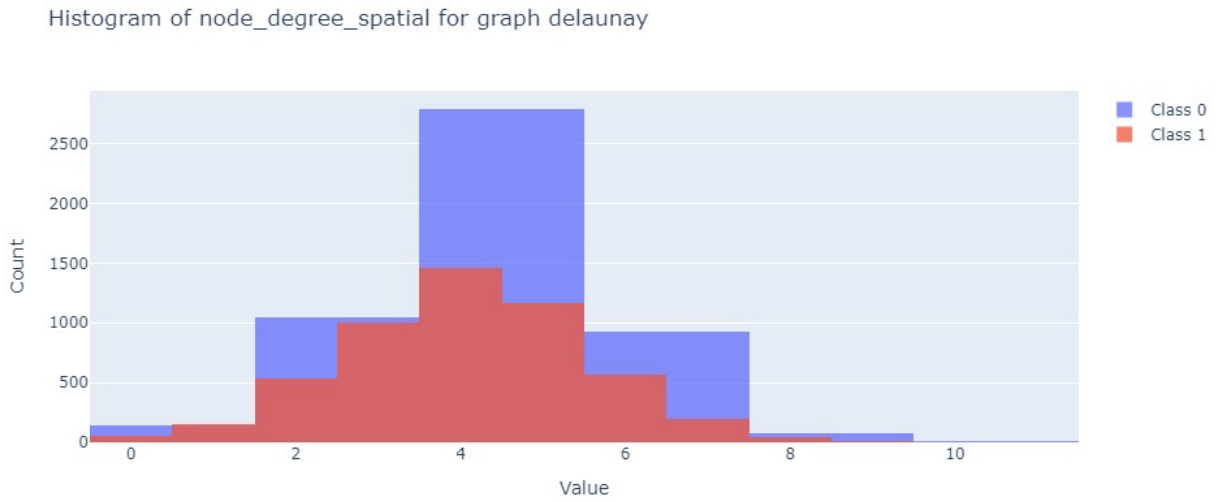


Figure 3.21: Histogram of spatial degree distribution between two cell populations on Delaunay graph

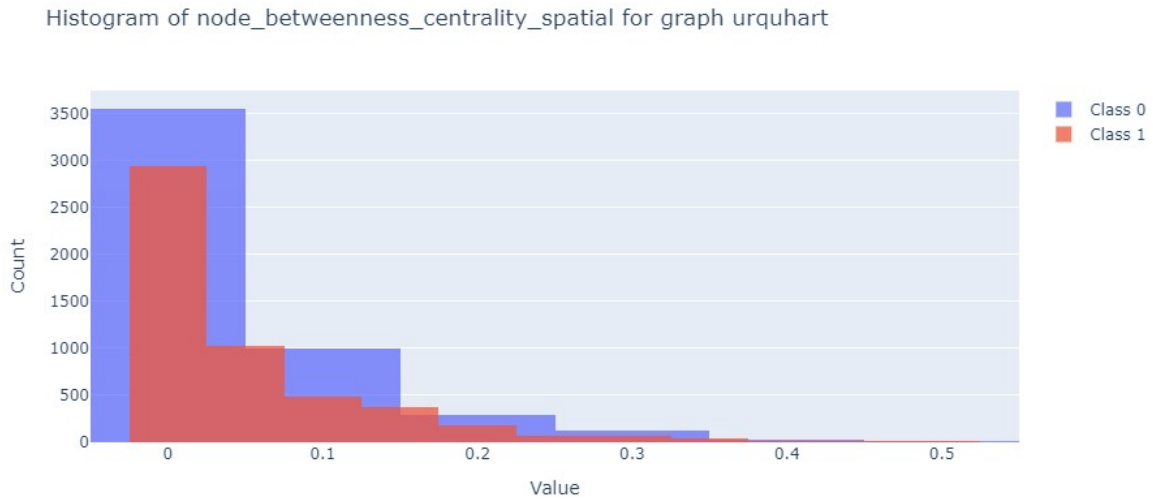


Figure 3.22: Histogram of spatial betweenness centrality distribution between two cell populations on Urquhart graph

$\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 30\,\mu$m (see Table 3.5), suggesting that the GMM identifies cell types slightly better than random guessing. This performance may be attributed to the fact that the proposed features (spatial and temporal) are insufficiently capable of capturing the differences between the two population types at both spatial and temporal levels, given the assumed type of displacement. Indeed, node features are predominantly based on x-y positions, which means that the displacement factor is not adequately described by these features. Consequently, this leads to poor performance of the method when used to distinguish between the two cell types. These outcomes highlight the necessity of exploring new features that might more effectively capture the proposed displacement behavior in the two mixed cell populations.

Table 3.5: Mean accuracy score on GMM for the simulated data containing two cell types

Graph	**Spatial and temporal node features**	**Spatial node features**	**Temporal node features**
Distance	0.5788	0.5107	0.5178

The SVM model model is utilized as a supervised learning method to evaluate the second prediction task for this simulation, which involves identifying two cell types per frame image. It remains to be determined whether this approach can enhance the accuracy of classifying both cell types. Table 6.6 (see Annex 6.2.2.1) presents the mean accuracy score when utilizing a supervised approach (70% for training and 30% for testing) for all possible combinations of features. It is observed that this approach performs better when considering x-y positions in combination with node features across most graph structures, achieving a maximum score of 64% on the Distance graph with $\tau_{sp} = \tau_{temp} = 150\,\mu$m (see Table 3.6). However, this score is insufficient to ensure that the SVM model can accurately classify both cell types.

Table 3.6: Mean accuracy score on a supervised method (SVM) for the simulated data containing two cell types

Graph	**x-y position**	**Node features**	**x-y position and node features**
Distance	0.6108	0.6197	0.6440

Based on previous results, it is observed that data from two cell populations can be effectively separated using node features from the Graph 3D structure. However, the performance is not satisfactory when attempting to identify two cell types within the population, as the graph does not sufficiently explain and capture the different behaviors present in each cell class. Therefore, other features need to be explored to help distinguish them. It is expected that the performance of the clustering method will improve by combining these features with those obtained through the graph structure.

As previously observed, both unsupervised (GMM) and supervised (SVM) methods exhibit suboptimal performance in the scenario where there exists a population with two cell types. The aim is now to approach the second prediction task through a graph-based supervised classification method that involves developing a GNN-based model to enhance the performance of detecting the two cell types. This type of model

can take into account the structure of Graph 3D (in addition to its computed node features) for its learning process. Subsequently, the second set of simulated data is considered for this analysis. Additionally, another graph structures are utilized in this case, which are described in detail in Annex 6.2.2.2.

To evaluate the supervised classification method, six simulations, along with their corresponding graph structures, are used as training data. Two simulations are chosen as validation data, and one simulation is reserved for testing the model. Furthermore, it was proposed to compute different graph structures for the training and validation datasets to augment the data and potentially reduce overfitting during the training phase of the GNN-based model. These graph structures were also computed for the test data to obtain a mean classification score across this dataset.

Figure 6.2 (see Annex 6.2.2.2) shows the loss and accuracy scores for the model (with a single GNN layer per encoder) on the training and validation datasets across epochs. The results indicate that the model follows a stable learning process with relatively few oscillations. As training progresses, the model gradually converges, exhibiting improved consistency in both accuracy and loss reduction.

Furthermore, it is observed that the model performs well as training progresses, achieving an accuracy score exceeding 90% in the later epochs. This suggests that the model effectively captures the underlying structure of Graph 3D-based data and generalizes well to the validation set. The steady improvement in accuracy and reduction in loss also imply that there is no overfitting, as the validation performance remains strong over time. Finally, the mean accuracy score on the test data is computed, and it is found that the model maintains a high score ($\approx 96\%$) on this dataset.

The GNN-based model outperforms the GMM because it not only considers the information from neighboring nodes within a given cell but also incorporates tracking information derived from the temporal structure. This capability is absent in the GMM, even when node features from Graph 3D are used to try to encode such information.

The influence of the number of GNN layers on the model's performance is investigated. Table 6.7 (Annex 6.2.2.2) displays the average accuracy score obtained per number of GNN layers (1, 2, 3, 4, and 5) on test data. The results indicate that all model setups perform well. It is observed that the number of GNN layers does not significantly affect the performance of the model, as the difference in accuracy score is not significant. This could imply that other factors might be more critical in determining the model's performance.

It is also noted that a single Graph Neural Network (GNN) layer may be sufficient to achieve optimal model performance. This suggests that cells require information only from their immediate neighbors within both the spatial and temporal structures of Graph 3D. This could be plausible due to the topology of Graph 3D, which might enable access to information for all nodes through their neighbors. Additionally, the temporal links between cells from consecutive frames could potentially encode tracking information and identify cells across the entire time-lapse video. Another possible explanation is that the simulated data used to evaluate this model represented adequate cell displacements, allowing the model to capture the spatial and temporal dynamics within the culture.

Next, a comparison is conducted between GNN-based models, each comprising two GNN layers per encoder, by varying the type of GNN layer in each encoder structure. It is proposed to test the architecture with Graph Convolutional Network

(GCN) layers, which are classic GNN layers commonly used in the literature. The model that combines GCN layers is denoted as GNN-GCN, which incorporates two GNN layers per encoder. For this model, edge betweenness centrality is used as the edge weight. Table 3.7 presents the mean accuracy per node features for all models. In this table, **position** refers to the x-y cells position; **position + spatial** refers to cells position and spatial features; **position + temporal** refers to cells position and temporal features; **position + spatial + temporal** includes cells position, spatial, and temporal features; and **position + spatio-temporal** refers to cells position, with spatial and temporal characteristics combined through summation.

Table 3.7: Comparison of model performance by node features

	Node Features				
Model	**position**	**position + spatial**	**position + temporal**	**position + spatial + temporal**	**position + spatio-temporal**
GNN	0.9894	0.9774	0.9839	0.9873	0.9874
GNN-GCN	0.6507	0.6124	0.6983	0.6964	0.6870

It is observed that the GNN-based model outperforms the other proposed model for all different node features considered. It is also noted that considering the x-y cell positions appears to be sufficient for identifying each class in Graph 3D. To understand why the x-y cell positions might be the sole requirement for the performance of the GNN-based model, an analysis of the GNN encoder structure (both spatial and temporal) is conducted below.

It is also noted that the GNN model utilizing GCN layers achieved scores approximately between 60% and 70% for all node features considered, which are lower than those obtained for the proposed model. This discrepancy can be attributed to the fact that GCN layers do not consider edge features beyond edge weights when computing messages, unlike GENConv layers, which may account for the difference in performance between the two models. It is also observed that the GNN-GCN model showed a higher score when considering the temporal knowledge of the graph structure.

Finally, the procedure (with 2 GNN layers per encoder) is compared by evaluating models that use only spatial or temporal information from the GNN encoder, as shown in Figure 3.9. The model using spatial information is denoted as GNN_{sp}, and the model with temporal information is denoted as GNN_{temp}. Table 3.8 reports the mean test accuracy for these two models per node features. Notably, GNN_{temp} outperforms GNN_{sp}, emphasizing the importance of temporal information, extracted from the spatio-temporal graph structure, in identifying two cell types without the need for tracking cells.

It is observed that the GNN-based spatial model's performance improves when temporal features are considered in the input data. This underscores the necessity of temporal dynamics for effectively solving the supervised task.

3.4.4 HDF-HUVECs

Finally, to validate the performance of the GNN-based model, it is proposed to apply this approach to experimental data. These data consist of two cell populations:

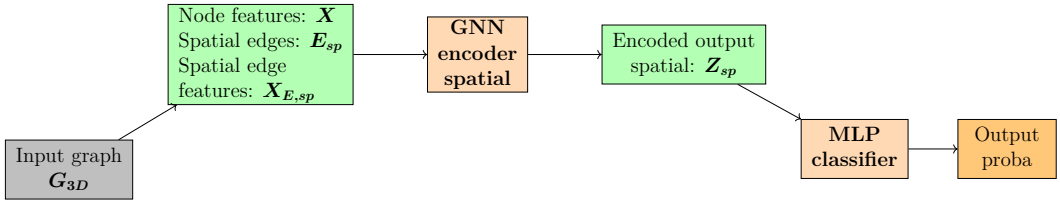


Figure 3.23: GNN-based spatial model GNN_{sp}

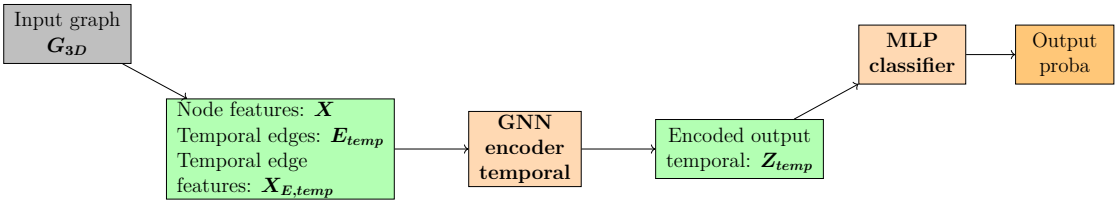


Figure 3.24: GNN-based temporal model GNN_{temp}

Table 3.8: Comparison of model performance per node features between GNN_{sp} and GNN_{temp}

	Node Features				
Model	**position**	**position + spatial**	**position + temporal**	**position + spatial + temporal**	**position + spatio-temporal**
GNN_{sp}	0.5809	0.5748	0.7535	0.7600	0.7763
GNN_{temp}	0.9813	0.9863	0.9861	0.9821	0.9866

HUVECs and HDF. The HUVECs were labeled using the green fluorescence protein [264, 64], which allows for the acquisition of a reference and the subsequent measurement of the model's performance. To mitigate the computational burden associated with calculating longer node features over the entire image size (11530 × 6940 pixels) in the time-lapse data, four regions of interest were initially selected, as shown in Figure 3.25. This strategy was also employed due to the availability of only one cell culture for analysis, which required the establishment of distinct training and testing datasets. However, it was observed that the bottom-left region contained unbalanced data for cell types. Therefore, it was discarded for the subsequent analysis.

To evaluate the performance of the GNN-based model, the remaining three regions were utilized as the training, validation, and test datasets, as depicted in Figure 3.25. As previously done with simulated datasets, graph structures were computed for the training, validation, and test datasets. This approach aimed to augment the data, potentially reduce overfitting during the training phase of the GNN-based model, and enhance the classification scores obtained. Model parameters were selected, and training was stopped early if the validation loss failed to improve for 10 consecutive epochs, with a maximum training limit of 100 epochs. The best-performing model, defined as the one achieving the lowest validation loss, was retained. The accuracy score was then computed on the test dataset using this model.

Table 6.8 (Annex 6.3.2) reports the mean accuracy on the test dataset for different node features and numbers of GNN layers, showing that the models exhibited poor performance overall. When only the x–y position was used as node features, the model's performance did not improve with additional GNN layers. A similar trend

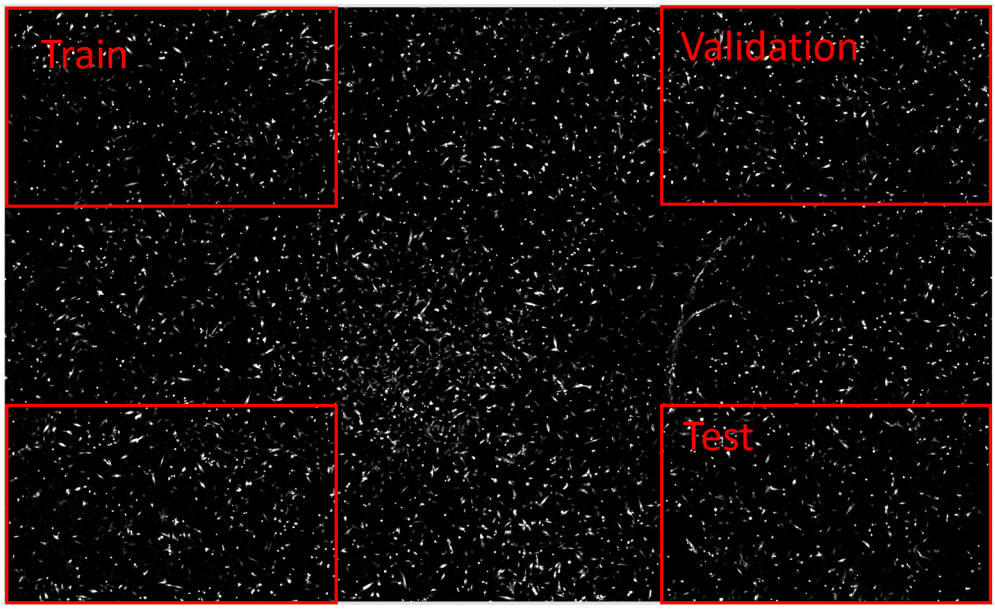


Figure 3.25: Region partition for images on experimental data

was observed across all tested combinations of node features, suggesting that model performance is largely independent of the number of GNN layers.

Subsequently, it was proposed to analyze the spatial and temporal structures of the GNN-based model, considering the previously introduced models GNN_{sp}, and GNN_{temp}. A third model, denoted as GNN_{sp_temp} (Figure 3.26), was considered. This model takes into account the entire structure of Graph 3D. In these models, each encoder was configured with three GNN layers for testing purposes. Table 6.9 (Annex 6.3.2) illustrates the mean accuracy obtained on the test dataset for different node features and the proposed models. It is observed that the three models show poor performance across the different node features considered, with the temporal structure showing a slight difference from the other models (Table 3.9). This might suggest that the different graph structures were unable to capture spatial and temporal information effectively, or that the proposed model was not suitable for these data. This finding encourages the exploration of alternative GNN architectures that could better address this supervised classification task.

Table 3.9: Model performance per node features for experimental data

	Node Features				
Model	**position**	**position + spatial**	**position + temporal**	**position + spatial + temporal**	**position + spatio-temporal**
GNN_{sp_temp}	0.5110	0.4990	0.5111	0.5110	0.4893

The suboptimal performances observed can be attributed to the challenges encountered during data acquisition, the unreliability of the ground truth data, which is based on fluorescence labeling. Although the chosen regions for testing were those that contained cleaner data in each image, difficulties persisted in processing the acquired images, accurately segmenting cells, and detecting them. These challenges constrained the ability to fully assess the potential of the proposed GNN-based model in addressing the supervised classification task.

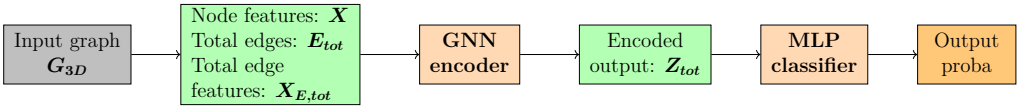


Figure 3.26: GNN-based global model GNN_{sp_temp}

Conclusion

In recapitulation, the proposed node features derived from the spatio-temporal graph structure **Graph 3D** facilitate the separation of two distinct datasets under identical experimental conditions through an unsupervised clustering approach. This was particularly evident in both simulated and experimental data. The proposed methodology could significantly enhance techniques that incorporate these node features to improve the analysis of individual cells, particularly when considering the lineage information of cells.

It was also possible to develop a Graph Neural Network (GNN)-based model to identify two cell types within the same population without the need for tracking, through a supervised learning task. This model demonstrated good performance in simulations but exhibited reduced performance when applied to experimental data, which may be attributed to the difficulties in data acquisition as previously described. However, this does not limit the exploration of alternative architectures for this model that could enhance its performance on this type of data. The developed GNN model could also be of significant importance for its application in tasks such as tracking and identifying cells across time-lapse sequences, which is not entirely straightforward. This would contribute to the subsequent analyses of cells once they have been detected and identified.

Chapter 4

Dynamic prediction of dry mass

This chapter presents the final application addressed in this PhD research. The primary objective of this application is to analyze the influence of a cell's neighborhood on the evolution of its morphological features, particularly the dry mass.

The study of dry mass, which is defined as the cellular mass after the removal of water content [22], is particularly fundamental for understanding key aspects of cellular behavior, including cell size, cycle, state, and homeostasis [95, 167, 265]. Moreover, Kandel et al. [139] present evidence suggesting that a cell's neighborhood influences its cellular features. Therefore, it is essential to consider the potential interactions between cells and their neighbors. This work focuses on forecasting dry mass by incorporating information from each cell's neighborhood, which will provide clear evidence of the impact of these interactions on dry mass. To achieve this objective, graphs can be employed.

In this study, 2D time-lapse microscopic videos and 2D simulation videos of cell cultures were utilized. For each frame of the video, a graph representation was constructed, where nodes represent cells and edges correspond to their spatial proximity. Data analysis was conducted on both experimental and simulated data, primarily involving spatial autocorrelation analysis of dry mass across different graph structures and correlation analysis of cell morphological features. The spatial autocorrelation analysis aimed to determine whether a cell's neighbors significantly influence the evolution of its dry mass. The results indicated that there exists the presence of correlation between the spatial proximity of the neighborhood of cells and the dry mass growth. Subsequently, Graph Neural Network (GNN)-based models were proposed to achieve the proposed prediction task. For evaluation purposes, the models were tested using different combinations of morphological features. The outcomes demonstrated that it was feasible to predict the trend of dry mass in simulated data. However, the models were unable to capture this behavior in experimental data.

4.1 Research context

Deep learning models integrated with recent bioimaging techniques have enabled the visualization, quantification, and monitoring of key cellular features over time, across large populations, and under label-free conditions [14]. Lens-free microscopy enables the extraction of important cellular features over time [22], such as dry mass, which refers to the cell's mass after its water content has been removed [189]. This

cellular feature plays a critical role in biology by remaining consistent across cell generations and revealing the stages of cell growth and division [22]. Therefore, dry mass becomes essential for understanding of cellular behavior, including cell size, cycle, state, and homeostasis [95, 167, 265].

Cell-cell communications are essential for various dynamic biological processes, including cellular differentiation, organ development, tissue homeostasis, and immune system function [17]. In particular, these cellular interactions could influence cell growth within cell populations. A study by Kandel et al. [139] examined the spatial correlations of growth rates within large populations of adherent cells, specifically fibroblasts and epithelial HeLa cells. The research revealed that the extent of correlation between growth rates and spatial coordinates can serve as a quantitative measure of "cellular influence", indicating how certain cells affect their neighbors [139]. Notably, the findings showed that while most cells do not exert significant influence, a small subset can impose their growth rates over a broad area around them [139]. These outcomes underscore the importance of studying how cells and their neighborhood can influence their growth. Graphs can serve as a means to achieve this.

As previously discussed in the preceding chapter, graphs can be utilized to try to represent cellular interactions or communications derived from 2D microscope images of cell cultures. In these graphs, nodes correspond to cells, and edges indicate their spatial proximity. Furthermore, temporal proximity can be inferred by calculating edges between two ordered or consecutive frame images. These spatial and temporal interactions could be employed in graph-based models for the study of dry mass over time.

GNN-based models can be utilized for the analysis and quantification of dry mass dynamics based on the spatial and temporal connections derived from the graph topology. This might enable the computation of the $l-$th graph representation state and its associated cellular features based on its past information $l-1, l-2, \ldots, 0$, where l represents a time-step of the video [232].

In this study, the graph representation states are fixed, and the prediction of dry mass at instant l is performed using both its past values and historical cellular features at time steps $l-1, l-2, \ldots, 0$. Dry mass can be predicted through two approaches: a classic and a dynamic method. The classic approach forecasts the dry mass at instant $l+1$ based on its past values from the graph at instant l. This procedure can then be repeated for $l+2, l+3, \ldots$. The dynamic approach, in contrast, predicts the dry mass at instant l, then updates the input graph information with the current prediction before estimating the next dry mass value at $l+1$. This process is similarly iterated for $l+2, l+3, \ldots$.

For this study, it will be necessary to define an appropriate graph construction method, with node (and even edge) features that capture both spatial and temporal aspects to the prediction of cell dry mass over time. This will help determine the suitable GNN structures to utilize on the models for performing this task.

4.2 Literature review

Graph-based deep learning architectures have been developed and have demonstrated strong performance across various domains and prediction tasks in recent years [28, 32, 57, 63, 142, 213, 232, 275, 284, 291, 298]. Research in this area often focuses

on spatio-temporal analysis to understand dynamic processes over time. In this context, the development of spatio-temporal graph neural networks is necessary, which typically consist of two main components: a spatial component and a temporal component.

The spatial domain derived from spatio-temporal GNNs has been modeled using graph-based deep learning architectures that incorporate the spatial structure of graphs [57, 285]. Chen et al. [57] utilized recurrent graph neural networks (RGNNs) with skip connections to model spatial dependencies in traffic forecasting. Wu et al. employed Graph Convolutional Networks (GCNs) with skip connections in their model for multivariate time series forecasting [285].

Models can address the temporal component through dynamic graphs and the historical information of the variable to be estimated (or predicted). Dynamic graphs can be modeled in two main ways: discrete-time dynamic graphs (DTDGs) and continuous-time dynamic graphs (CTDGs). DTDGs take graph snapshots at fixed intervals to show its state at specific times. Specifically, a graph snapshot represents the state of the graph, including its present nodes and edges, at a given point in time. CTDGs record events as they happen, providing a more flexible view of changes in the graph.

Early research on dynamic graph learning focused on DTDGs. Methods included combining graph snapshots for static models [120, 163], breaking down snapshot tensors [78, 184], or creating embeddings for each snapshot and combining them using various techniques like weighted aggregation, time-series methods, recurrent neural networks (RNNs), or temporal smoothness constraints [235, 288]. Some studies also used random walks, starting on the first snapshot and adjusting for later ones [186, 293].

More recent work has looked at CTDGs, using methods like random walks with changing transition probabilities [24, 197, 198] and RNN-based models that update node embeddings when new edges are added [152, 183, 261]. Similar challenges are seen in dynamic knowledge graphs [71, 94, 98, 286]. Many approaches update node memories with RNNs but often skip updating the neighborhood, which can lead to outdated embeddings and limited representation.

Spatio-temporal GNN models have been applied in various domains, including traffic forecasting for both long-term and short-term predictions[57, 157, 292, 298], dynamic representation of brain graphs [142], emotion perception [32], spatio-temporal interactions between objects [63], and pandemic forecasting [140, 275].

Sanchez-Gonzalez et al. [232] have proposed an approach, called "Graph Network-based Simulators" (GNS), for learning to simulate physical domains, which can generalize from single-time point predictions with thousands of particles during training, to different initial conditions. Their GNN model represents the state of a physical system with particles, expressed as nodes in a graph, and computes dynamics via learned message-passing.

However, certain deep learning models assume a fixed number of nodes across all frames, with connections between them changing over time. Some of these architectures do not incorporate message passing with multi-dimensional edge features, which could significantly influence information transmission and updates to node and edge features (if applicable) when applying GNN layers. Examples of these architectures include graph convolution layer (GCN) [146]; gated graph convolution layer [161], which is a GNN layer with gated recurrent units (GRUs). Another models have been adapted to handle multi-dimensional edge features in message passing,

such as graph attention networks (GATs) [267].

Yu et al. [291] introduced ROLAND, an effective graph representation learning framework designed for real-world dynamic graphs, which enables the adaptation of any static GNN to dynamic graphs. This approach consists of viewing the node embeddings at different GNN layers as hierarchical node states and then recurrently update them over time. Moreover, this framework facilitates the addition of new nodes and the deletion of existing ones on the graph.

MAGIK [213] is a deep learning model based on GNNs that handles both spatial and temporal dynamics. This model demonstrates strong performance in estimating dynamical properties in biological scenarios, such as cell division. MAGIK utilizes a generator function that creates graph snapshots by considering spatial and temporal connections. Each snapshot is generated by reusing the previous one, allowing the model to make predictions from data that are unrelated to the training data.

For the prediction of dry mass, it was decided to implement the model proposed by Sanchez-Gonzalez et al. [232], which offers an innovative approach to making predictions. Additionally, it was decided to combine this method with the attentional mechanism provided by MAGIK [213] to determine whether it enhances the model's performance on this regression task. The MAGIK model was initially utilized for comparison purposes but was discarded for further analyses because it did not enable the computation of dry mass prediction values for the entire time-lapse video due to memory issues.

4.3 Methodology

Figure 4.1 illustrates the methodology employed to address this work. Images acquired were segmented using a deep learning model (Cellpose[1]) [251], which is primarily utilized for cell segmentation and detection. A convolutional neural network (CNN) deep learning method was applied to measure cellular morphological features from the segmented images. Subsequently, graphs were constructed from the x-y cell positions, with nodes representing cells and edges representing their spatial proximity. Features of nodes and edges were computed to define the input data for the GNN model. Once training is completed, the dry mass is predicted through a dynamic approach, which is described in Section 4.3.2.

[1] https://www.cellpose.org/

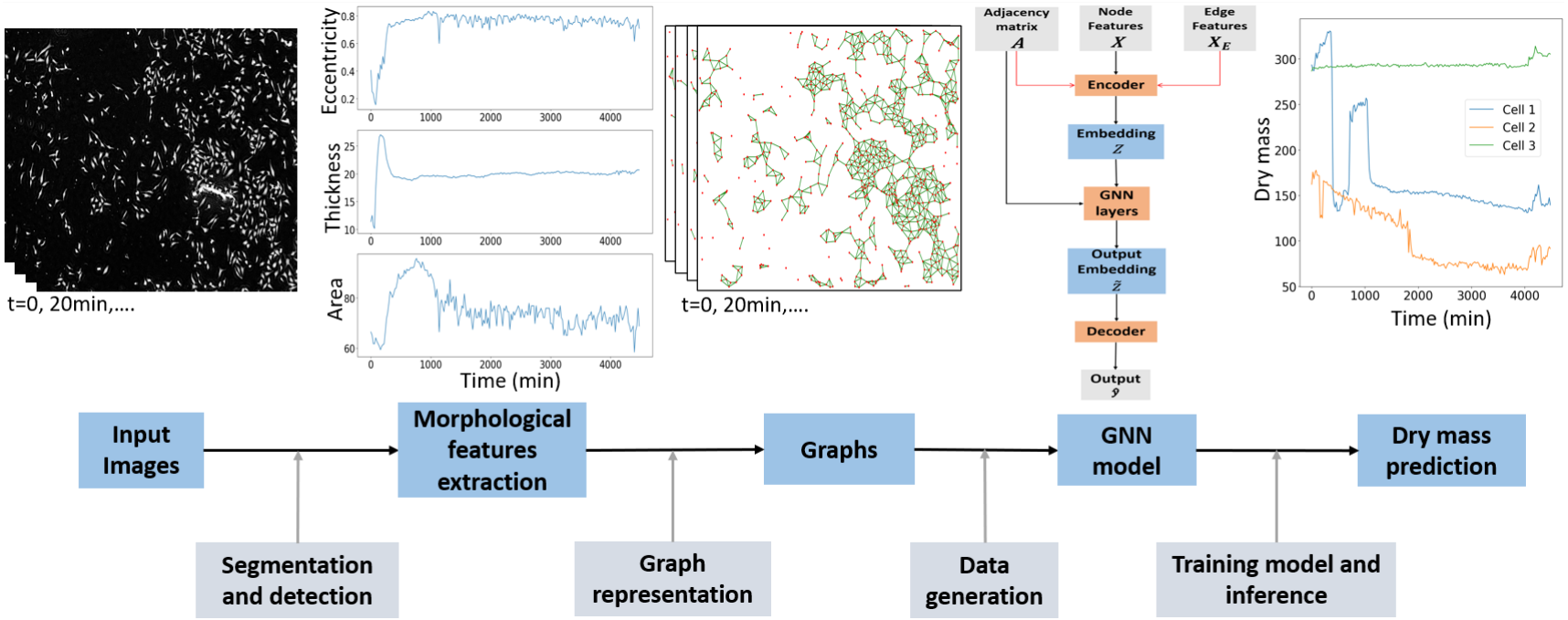


Figure 4.1: Proposed workflow for dynamic prediction of dry mass. Images acquised were segmented via Cellpose [251], which is predominantly used for cell segmentation and detection. A CNN model was employed to assess cellular morphological features from the segmented images. Graphs were created from the x-y cell positions, with nodes representing cells and edges depicting their spatial proximity. Node and edge features were calculated to define the input data for the GNN model. Finally, a dynamic method is employed to predict the dry mass once the training phase is complete.

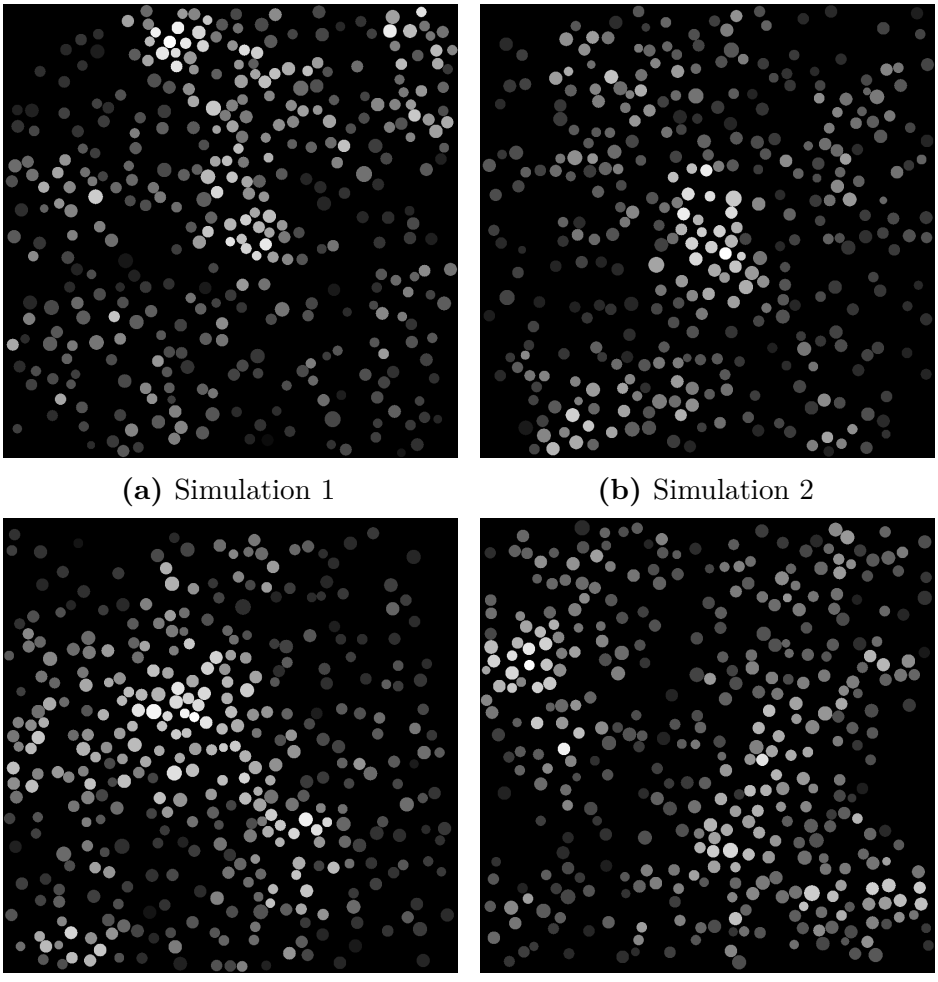
4.3.1 Datasets

For this study, two datasets consisting of 2D time-lapse microscopic videos of cell cultures were analyzed. The first dataset is simulated data designed to replicate the behavior of a cell culture over time. The second dataset comprises experimental data corresponding to HeLa cell cultures.

4.3.1.1 Simulated data

This dataset is designed to simulate the progression of a cell culture over time. At the initial time point ($t = 0$), a given number of cells are initialized with random positions and mass values. The displacements of each cell are governed by its spatial proximity to the other cells. A rule is then applied to increase the dry mass based on the proximity of cells where cell i is close to at least 3 neighboring cells, its mass is increased by a given value. Additionally, a rule for cell division is defined, where cell i divides itself if its mass exceeds a threshold value. In this case, 2 new cells are created, each with a mass equal to half of the original cell's mass.

Figure 4.2 illustrates a frame representation for 4 simulations conducted on the simulated dataset. These images show different behaviors under the proposed conditions.



(a) Simulation 1 **(b)** Simulation 2

(c) Simulation 3 **(d)** Simulation 4

Figure 4.2: Frame representation of 4 simulations from the simulated data

4.3.1.2 Experimental data

Experimental data consist of 2D time-lapse videos of HeLa cell cultures (Figure 4.3). Each frame has a resolution of 3840×2748 pixels, pitch $1.67\,\mu$m. The

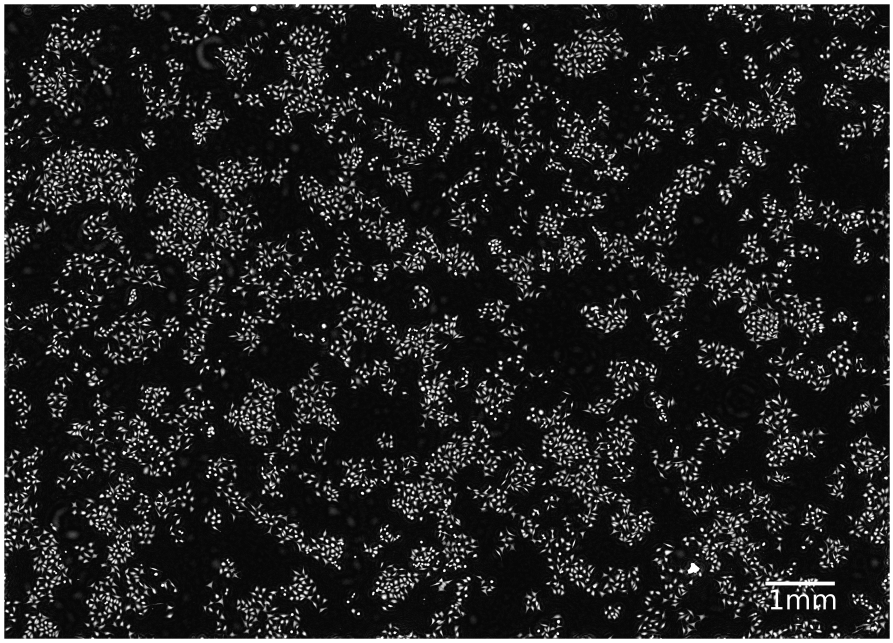


Figure 4.3: Lens-free image of HeLa cells

morphological features of cells are extracted from segmented images of the video. This structure contains information about the current frame of the video and, for each cell, its x-y pixel position, ID tracking, area, perimeter, length[2], width[3], eccentricity[4], orientation[5], thickness, and dry mass, which is proportional to the optical volume difference (OVD).

It was observed that some cells are close to the borders of the image. Consequently, these cells can enter or exit the image from one frame to the next, and the annotations made by the tracking algorithm do not account for these cells.

Once all the relevant information has been extracted from the time-lapse videos, the data is processed. Z-score normalization is then applied to the morphological characteristics, and a normalization between 0 and 1 is applied to the positions relative to the image size.

4.3.2 Prediction task

This study focuses on the prediction of dry mass, described as follows. Given a set of 2D images from a video, captured at regular time-steps from $t = 0, 1, 2, \ldots, T$, where T denotes the final frame, graph representations are constructed. These graphs include node and edge features, which are derived from cellular morphological features and cell positions. The objective is to forecast the dry mass value at $t + 1$, based on its past information and the graph representation at time t. This task

[2]**Length** is measured as the longest diameter of the masked cell (ellipse).

[3]**Width** is measured as the shortest diameter of the masked cell (ellipse).

[4]**Eccentricity** measures how much the cell deviates from being circular [15]. It calculates the offset focal spot on the main axis of the masked cell (ellipse).

[5]**Orientation** is the angle (in radians) formed by the position of the masked cell (ellipse) and the reference axes.

can be addressed through either a classic or a dynamic approach as defined in [232]. These two methods are summarized below:

- **One-step method** is the classic approach, which uses information at time t to predict the dry mass for $t+1$. The procedure can be repeated iteratively for $t+2, \ldots, T-1$.
- **Rollout method** is the dynamic approach, which is described as follows. Starting at $t=0$, the information available at $t=0$ is utilized to estimate the dry mass for $t=1$. The predicted value is then fed back into the model to predict the next value at $t=2$. This process is repeated until the end of the time-lapse video.

To achieve this, simulated data was initially considered, which includes cell diameter, cell positions, and cell dry mass. Subsequently, all available information from the experimental data was utilized, with features selected to predict the cell dry mass.

4.3.3 Data analysis

4.3.3.1 Spatial autocorrelation

First, the objective is to determine whether a correlation exists between the spatial proximity of cells and their dry mass growth. To statistically verify this assumption, a hypothesis test is formulated. The null hypothesis, $\boldsymbol{H_0}$, states that there is no spatial autocorrelation in dry mass, while the alternative hypothesis, $\boldsymbol{H_1}$, states that spatial autocorrelation is present. Moran's I [159, 192] is then employed to assess whether $\boldsymbol{H_0}$ should be accepted or rejected. This statistic measures spatial autocorrelation by considering both feature locations and feature values simultaneously [83]. Moran's I is defined as follows

$$I = \frac{N}{W} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}(m_i - \bar{m})(m_j - \bar{m})}{\sum_{i=1}^{N}(m_i - \bar{m})^2}.$$

where

- N is the number of cells;
- $m = m_1, \ldots, m_N$ is the variable of interest (i.e., dry mass);
- $\bar{m}$ is the mean of m;
- w_{ij} corresponds to the Euclidean distance between cells i and j based on their x-y positions;
- W is the sum of all w_{ij}, i.e., $W = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}$.

The $z-$score for this statistic is computed as

$$z_I = \frac{I - \mathbb{E}[I]}{\sqrt{\mathbb{V}[I]}}$$

where the expected value $\mathbb{E}$ and the variance $\mathbb{V}$ of I are given by

$$\mathbb{E}[I] = -\frac{1}{n-1}$$
$$\mathbb{V}[I] = \mathbb{E}[I^2] - (\mathbb{E}[I])^2 .$$

To determine whether $\boldsymbol{H_0}$ is statistically significant, the p-value is calculated. The p-value indicates how probable it is to obtain the observed effect—or one more extreme—if the null hypothesis were true [19, 278]. Then, $\boldsymbol{H_0}$ is rejected if

$$p\text{-value} \leqslant \alpha,$$

where α is the significance level, which is typically set at 5% [230].

The statistical significance of $\boldsymbol{H_0}$ was assessed using both simulations and the HeLa data. The weights $w_{i,j}$ were computed using the graph construction methods introduced in Section 1.3.2.4: Delaunay (without thresholding), Gabriel (without thresholding), Distance with connectivity radius $\tau = 300\,\mu\text{m}, 500\,\mu\text{m}, 650\,\mu\text{m}, 1000\,\mu\text{m}$, and k-NN with $k = 4, 5, 6, 8, 10$. The significance level α is set at 0.05.

Figures 4.4 and 4.5 display the Moran's Index (Moran's I) and the corresponding p-value for the best Moran's Index obtained over the time-lapse sequence in a set of simulated data with 71 frames. Observations indicate that Moran's I generally increases over time for most graph structures, with the exception of Distance graphs, which become flat or stationary towards the end of the time-lapse. Among the different graph construction methods, the Gabriel graph shows the highest Moran's I values. The p-value remains below $\alpha = 0.05$ throughout the entire time-lapse, implying that the null hypothesis $\boldsymbol{H_0}$ can be rejected. This indicates the statistically significant presence of spatial autocorrelation in dry mass. Additional outcomes for Moran's I and the corresponding p-value for the best Moran's I obtained from other five simulations of simulated data can be observed in Annex 6.4.

This analysis was also conducted using the HeLa data, as depicted in Figures 4.6 and 4.7. It is observed that Moran's I exhibits a growing trend with high variations over time for most graph structures, with the exception of Distance graphs, which demonstrate low fluctuations and a less pronounced growth trend. Initially, the p-value exceeds the chosen criterion, implying that there is no spatial autocorrelation in dry mass. However, this situation changes after frame 50, indicating a rejection of the null hypothesis $\boldsymbol{H_0}$. This can be attributed to the fact that there are insufficient interactions between cells at the beginning of the time-lapse video. As time goes by, cells begin to approach and interact with each other, which likely influences the behavior of dry mass growth.

4.3.3.2 Correlation analysis

Once all data have been normalized, it is proposed to analyze the correlation between all available cell morphological features. This analysis aims to determine which variables can be employed to construct the node features matrix, which is utilized in the models. This correlation analysis is applied to the HeLa data, as simulated data contain information only for x-y positions, diameter, and dry mass.

For HeLa data, the correlation between the variables—area, perimeter, length, width, eccentricity, orientation, and dry mass—is analyzed. First, the mean values of each variable are computed using all cells. Subsequently, the correlation between

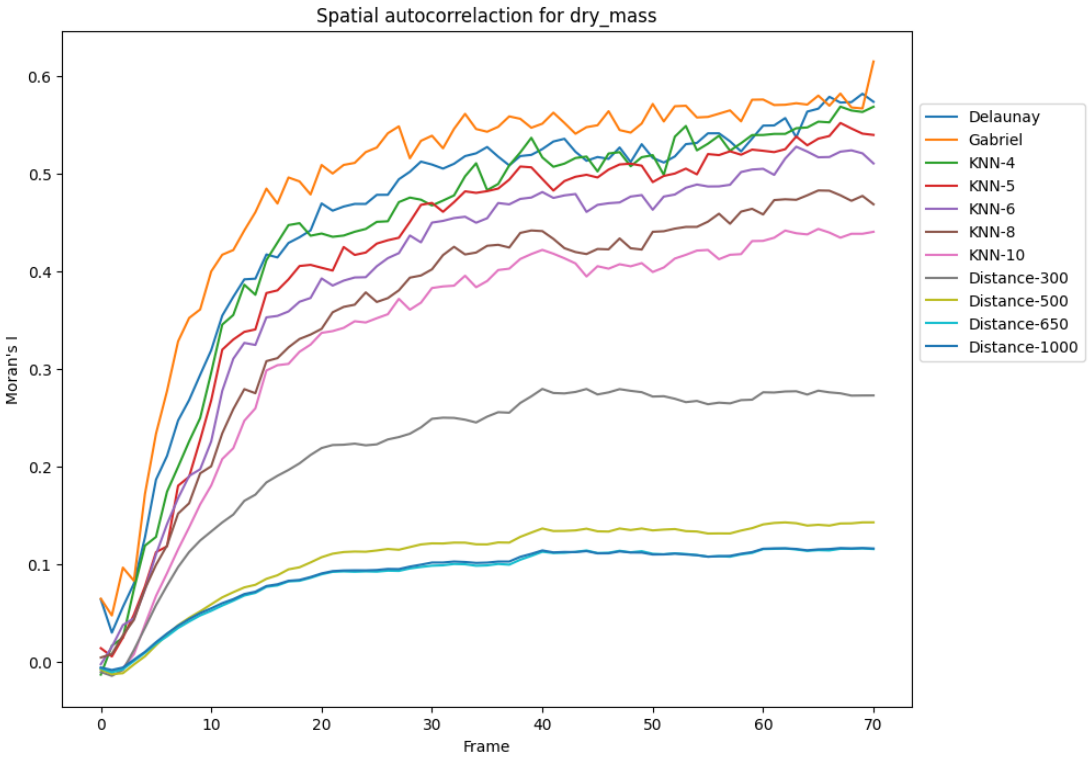


Figure 4.4: Moran's Index on a simulation

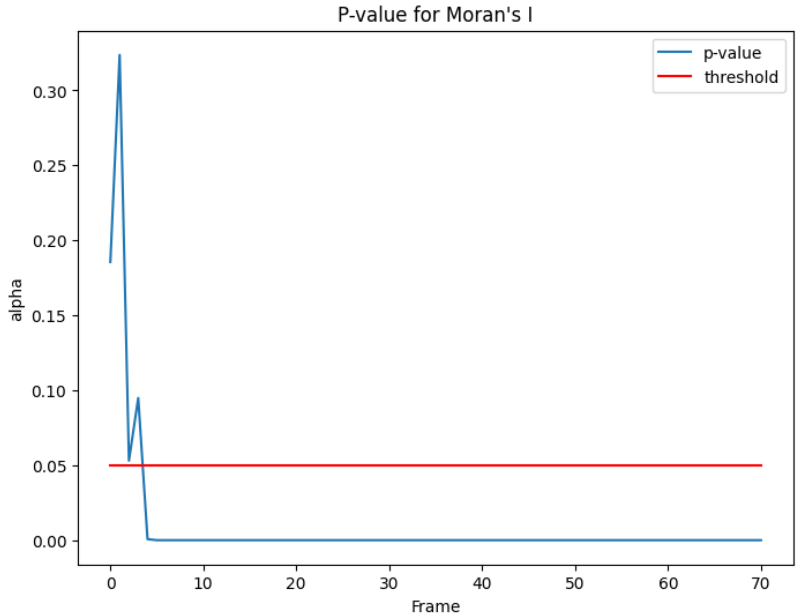


Figure 4.5: *P*-value from Moran's Index on a simulation

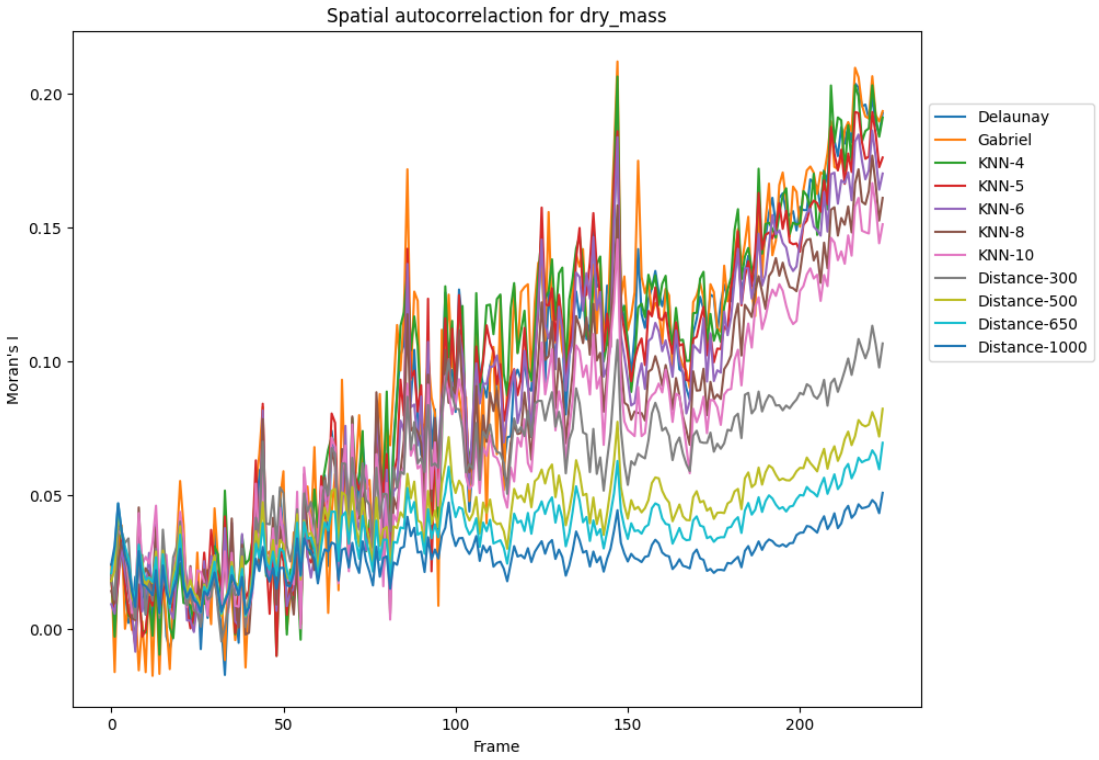


Figure 4.6: Moran's Index on HeLa data

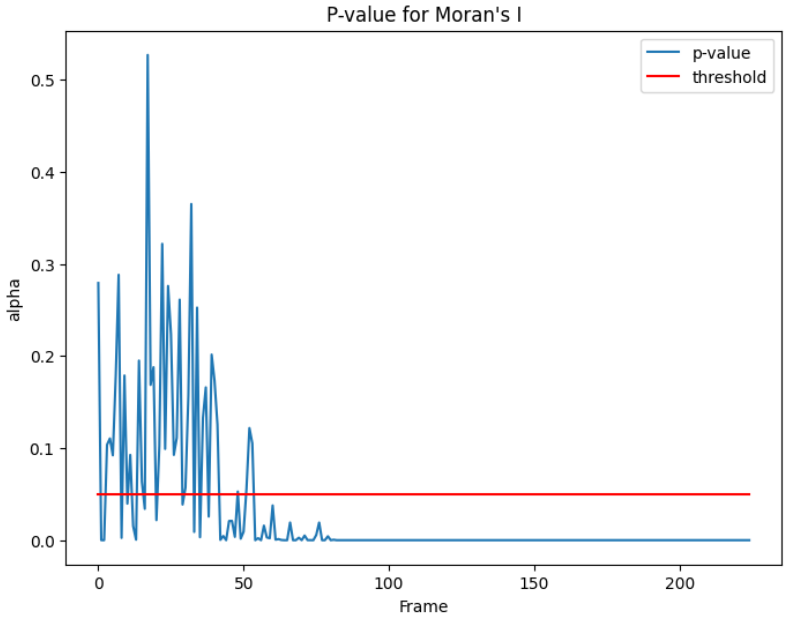


Figure 4.7: *P*-value from Moran's Index on HeLa data

all variables is calculated using their mean values derived from the available cell information for each frame (see Figure 4.8).

It is noticed that there are highly positive correlations between:

- Dry mass with area, perimeter, and length;
- Length and perimeter.
- Eccentricity with length and perimeter.

Then, this analysis can be used to construct the node features matrix for the graph.

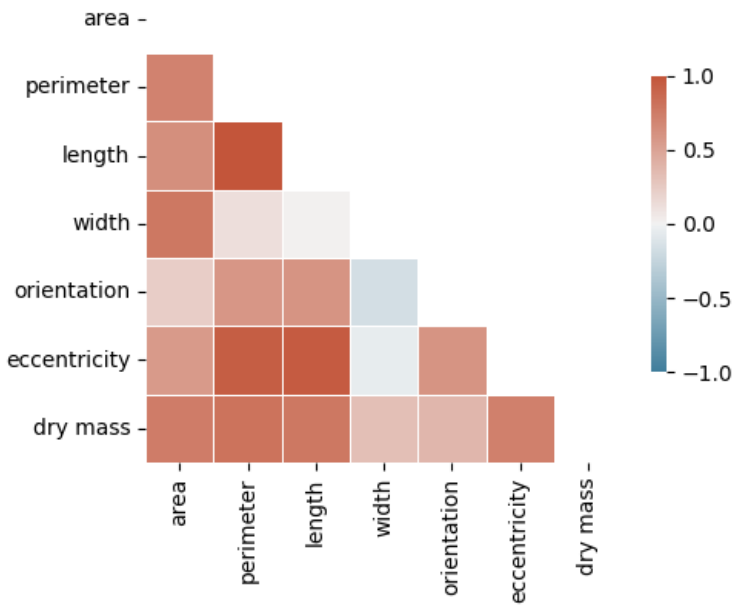


Figure 4.8: Correlation analysis on HeLa data

4.3.4 Graph construction

One important remark related to the different models mentioned above is that the input data plays an important role in model performance. Specifically, a suitable graph structure must be built as input for the models to achieve good performance on the proposed prediction task. Edges are constructed using the Distance graph (described in Section 1.3.2.4), where, for any two nodes i and j, the Euclidean distance d_{ij} is computed from their x-y positions. Then, an edge between i and j (noted by e_{ij}) is added to the graph if

$$d_{ij} \leqslant \tau$$

where τ is a threshold value chosen.

Additionally, two other graph structures (described in Section 1.3.2.4) are considered for the analysis: Delaunay and Gabriel graphs, as they exhibit high Moran's I values in the spatial autocorrelation analysis.

To apply the Rollout method for predicting the dry mass, it is necessary to use graph snapshots extracted from the sequence of 2D images. For reference, a graph snapshot represents the state of the graph, including its present nodes and edges, at a given point in time.

In this work, graph snapshots are constructed using fixed-length time intervals of duration C. Given a time t, the graph snapshot is obtained as follow. The graph edges are built by utilizing the x-y position provided at time t and the graph structure selected from one of the three previously described graph construction methods. The node features matrix $\boldsymbol{X}$ primarily consists of x-y position values and selected node features, derived from the earlier correlation analysis, at times $t, t-1, t-2, \ldots, t-C$. The edge features matrix $\boldsymbol{X_E}$ is consistent across all graph snapshots and consists of the measures defined in Section 1.3.2.2: relative displacement, its magnitude (Euclidean distance), and cosine similarity.

4.3.5 Models

In this section, the different models implemented are described. The first model, noted by $\boldsymbol{M_0}$, is the one implemented in [232]. This model is capable of learning to simulate diverse and complex physical systems, including fluids, rigid solids, and deformable materials that interact with each other [232]. The objective is to leverage this model to predict the dry mass. From now on, $\boldsymbol{M_0}$ will be referred as model LS (Learned Simulator). Figure 6.21 illustrates a general overview of the proposed model. It consists of the following architectures: an encoder, a processor and a decoder. The encoder and decoder are MLP models with ReLU activation functions for the hidden layers and Layer Normalization[6]. The processor includes GNN layers.

The GNN layer used in model $\boldsymbol{M_0}$ is described. This GNN layer is called Interaction Network (IN) and is proposed in [26].

For a graph G, let the feature of node i be v_i, and the feature of edge (i, j) be $e_{i,j}$. There are three stages for IN to generate new features of nodes and edges.

1. **Message generation.** If there is an edge pointing from node i to node j, node i sends a message to node j. The message carries the information of the edge and its two nodes, so it is generated by the following equation

$$\text{Msg}_{i,j} = \text{MLP}(v_i, v_j, e_{i,j}).$$

2. **Message aggregation.** In this stage, each node of the graph aggregates all the messages that it received to a fixed-sized representation. In the IN, aggregation means summing all the messages up, i.e.,

$$\text{Agg}_i = \sum_{(j,i)\in G} \text{Msg}_{i,j}.$$

3. **Update.** Finally, the features of nodes and edges are updated with the results of previous stages. For each edge, its new feature is simply the sum of its old feature and the corresponding message, i.e., $e'_{i,j} = e_{i,j} + \text{Msg}_{i,j}$. For each node, the new feature is determined by its old feature and the aggregated message, i.e.,

$$v'_i = v_i + \text{MLP}(v_i, \text{Agg}_i).$$

The second model, denoted by $\boldsymbol{M_1}$, is based on $\boldsymbol{M_0}$, where node features are updated by attention-based components from the MAGIK framework [213] in the GNN layers.

[6] https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html

It is important to note that MAGIK was not included in the performance analysis. This decision was made because the model was unable to predict dry mass values across the entire time-lapse sequence of the HeLa data.

For each model, different configurations of the following hyperparameters were analyzed: the number of hidden layers in the encoder and decoder, the number of GNN layers, the optimization method, the learning rate (LR), the weight decay, and the LR scheduler.

4.3.5.1 Loss function

For the regression task, it is supposed that there are n cells with their corresponding morphological features from $t = \{0, 1, 2, \ldots, T\}$, where T is the last time-step. The loss function $\mathcal{L}$ is defined by

$$\mathcal{L}(\boldsymbol{y_t}, \hat{\boldsymbol{y}}_{\boldsymbol{t}}) = \frac{1}{n}\sum_{i=1}^{n}(y_{i,t} - \hat{y}_{i,t})^2 \quad \text{for any } t \in \{1, 2, \ldots, T\} \tag{4.1}$$

where

- $\boldsymbol{y_t} = (y_{1,t}, \ldots, y_{n,t})$ is the vector with real values of target variable for all cells at time $t \in \{1, 2, \ldots, T\}$.
- $\hat{\boldsymbol{y}}_{\boldsymbol{t}} = (\hat{y}_{1,t}, \ldots, \hat{y}_{n,t})$ is the vector with estimated values of target variable for all cells at time $t \in \{1, 2, \ldots, T\}$.

4.3.5.2 Performance metrics

To evaluate the model performance, the mean squared error (MSE) is computed. This metric can be computed for the two prediction methods introduced in Section 4.3.2. Then, the MSE for One-step method, denoted by OS_{MSE}, and Rollout method, denoted by $Roll_{MSE}$, is defined as

$$OS_{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = Roll_{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{Tn}\sum_{i=1}^{n}\sum_{t=1}^{T}(y_{i,t} - \hat{y}_{i,t})^2, \tag{4.2}$$

where

- $\boldsymbol{y} = (y_{1,1}, \ldots, y_{1,T}, \ldots, y_{n,1}, \ldots, y_{n,T})$ is the vector of true values of target variable for all cells.
- $\hat{\boldsymbol{y}} = (\hat{y}_{1,1}, \ldots, \hat{y}_{1,T}, \ldots, \hat{y}_{n,1}, \ldots, \hat{y}_{n,T})$ is the vector of predicted values of target variable for all cells.

The main interest lies in the use of the Rollout method, as it enables dynamic forecasting of the target variable based on its previously predicted values.

In addition to the MSE based on the Rollout method, the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are also computed using this method to evaluate the model performance. These metrics are presented as follows

$$\text{RMSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sqrt{\frac{1}{Tn}\sum_{i=1}^{n}\sum_{t=1}^{T}(y_{i,t} - \hat{y}_{i,t})^2}. \tag{4.3}$$

$$\mathrm{MAE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{t=1}^{T} |y_{i,t} - \hat{y}_{i,t}|. \tag{4.4}$$

$$\mathrm{MAPE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{100}{nT} \sum_{i=1}^{n} \sum_{t=1}^{T} \left| \frac{y_{i,t} - \hat{y}_{i,t}}{y_{i,t}} \right|, \tag{4.5}$$

where $\boldsymbol{y}, \hat{\boldsymbol{y}}$ are defined as in (4.2).

4.4 Results and Discussion

Before presenting the results of the analysis, it is important to make some remarks. Predicted dry mass values were computed using the Rollout method previously described. These values are restored to their original magnitude through the inverse process of Z-score normalization. This is done to enable the computation of evaluation metrics.

For the analysis, it is supposed all cells do not have the same tracking length. This enables the creation of a dynamic input node features matrix, varying its size (number of rows/nodes) across time.

For the simulated data, x-y position (pos), dry mass, and diameter were considered as node features. Eleven simulations were used as training data. Four simulations were chosen as validation data, and three simulations were reserved for testing the models.

For the experimental data, the following features were employed as node attributes: x-y position (pos), dry mass, perimeter, length, area, and eccentricity. Two 2D microscopic videos were used for training and validation purposes. A single 2D microscopic video was utilized to test the models.

The models used for the prediction task were $\boldsymbol{M_0}$ (denoted LS) and $\boldsymbol{M_1}$ (denoted LS MAGIK). For each model, three different graph constructions were evaluated: Delaunay, Distance, and Gabriel graphs, with a threshold value of $\tau = 100\,\mu$m.

4.4.0.1 Simulated data

Tables 4.1, 4.2, and 4.3 present the evaluation metrics for all cells under different node feature configurations and graph structures, corresponding to three separate simulations from the simulated dataset. It is noticed that there are high values for all evaluation metrics across all graphs and node features tested. The model $\boldsymbol{M_1}$ generates the lowest values for all cases in the three datasets considered. For the first simulation, the best performance is obtained when considering only the dry mass along with the x-y position as node features and the Delaunay graph. This implies that models only need knowledge of past dry mass values of cells and their neighborhood to attempt to forecast their values in the following time step.

For the second simulation, the best performance is achieved by the model $\boldsymbol{M_0}$ when utilizing dry mass, diameter, and x-y position as node features and the Distance graph. This implies that the model attempts to predict future dry mass values by incorporating information from past diameter values. The model $\boldsymbol{M_1}$ achieves its lowest evaluation metric values when using dry mass and x-y position as node features and the Gabriel graph, which highlights the use of only past dry mass values for its next prediction, as before.

For the third simulation, the model $\boldsymbol{M_0}$, which uses dry mass, diameter, and x-y position as node features and a Distance graph, shows the best performance. The model $\boldsymbol{M_1}$ achieves its lowest best values when considering a Gabriel graph and using x-position and dry mass as node features. This holds true for almost all evaluation metrics, with the exception of the mean absolute error, which achieves its lowest score by a Delaunay graph with x-y position, dry mass, and diameter as node features.

Figures 4.9a and 4.9b, depict the true and predicted dry mass values across the two models considered for two cells from the first simulation data. In these figures, it is observed that the two models attempt to forecast the growth trend of dry mass. The predicted dry mass values obtained by the model $\boldsymbol{M_1}$ seem to be closer to the ground truth values than those generated by the model $\boldsymbol{M_0}$, which could underscore the importance of adding the attention mechanism to its structure.

Figures 4.10a, 4.10b, 4.11a and 4.11b depict scatter plots of true versus predicted dry mass values over time for two cells from the first simulation dataset using the two models, $\boldsymbol{M_0}$ and $\boldsymbol{M_1}$. These figures reveal that model $\boldsymbol{M_1}$ achieved a better linear regression fit between the ground truth and predicted values compared to $\boldsymbol{M_0}$. However, the available evidence is insufficient to demonstrate that the two proposed models can accurately predict the dry mass of cells based on their neighborhood over time. Although there is a correlation between the spatial proximity of cells and their dry mass, as previously discussed, this alone does not guarantee the models' predictive capability. This implies that either the graph structures considered or the architecture of the models utilized were not appropriate for the prediction task, suggesting the need to research better graph structures and models that could enhance the performance obtained.

Table 4.1: Comparison of models' performance per features and graphs from 1st simulation. The lowest value for each performance metric is highlighted in black.

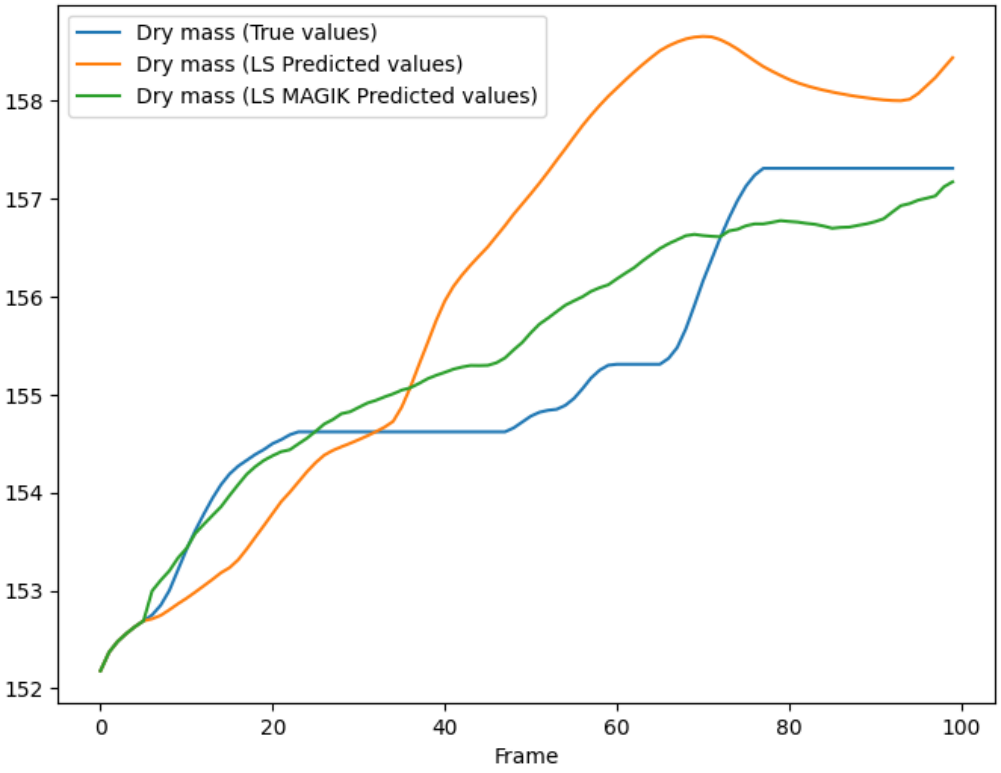
		MSE		RMSE		MAE		MAPE	
Graph	**Node features**	M_0	M_1	M_0	M_1	M_0	M_1	M_0	M_1
Delaunay	pos+dry mass	**555.1755**	**69.9300**	**23.5622**	**8.3624**	**9.2237**	**3.9415**	**0.0450**	**0.0206**
	pos+dry mass+ diameter	941.2333	99.5091	30.6795	9.9754	13.3664	4.5937	0.0647	0.0253
Distance	pos+dry mass	645.0998	228.9340	25.3988	15.1306	11.6570	7.0366	0.0594	0.0360
	pos+dry mass+ diameter	1252.2226	61.5153	35.3868	7.8432	14.7990	3.1899	0.0708	0.0169
Gabriel	pos+dry mass	899.4991	262.5254	29.9916	16.2026	12.0140	7.2046	0.0570	0.0358
	pos+dry mass+ diameter	608.5880	330.6879	24.6696	18.1848	13.0028	9.7860	0.0680	0.0549

Table 4.2: Comparison of models' performance per features and graphs from 2nd simulation. The lowest value for each performance metric is highlighted in black.

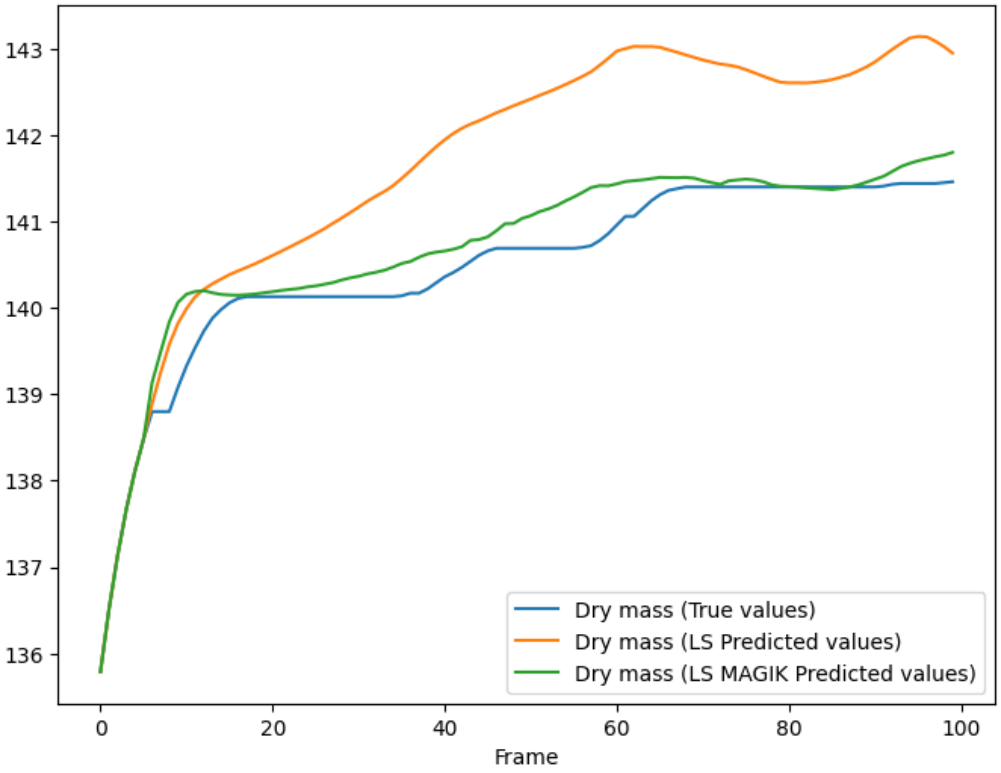
		MSE		RMSE		MAE		MAPE	
Graph	**Node features**	M_0	M_1	M_0	M_1	M_0	M_1	M_0	M_1
Delaunay	pos+dry mass	7697.1788	125692.9094	87.7336	354.5320	54.4472	246.0535	0.0936	0.3800
	pos+dry mass+ diameter	45679.0397	14981.7804	213.7266	122.4001	152.6239	76.6616	0.2540	0.1347
Distance	pos+dry mass	14435.0425	92300.5412	120.1459	303.8100	68.9653	190.4496	0.1108	0.2907
	pos+dry mass+ diameter	**3136.6527**	44680.6515	**56.0058**	211.3780	**36.7300**	135.2340	**0.0699**	0.2112
Gabriel	pos+dry mass	9104.0105	**11263.9685**	95.4149	**106.1318**	65.5802	**67.9974**	0.1182	**0.1156**
	pos+dry mass+ diameter	15902.2622	134329.9551	126.1042	366.5105	86.5014	222.9175	0.1498	0.3389

Table 4.3: Comparison of models' performance per features and graphs from 3rd simulation. The lowest value for each performance metric is highlighted in black.

		MSE		**RMSE**		**MAE**		**MAPE**	
Graph	**Node features**	M_0	M_1	M_0	M_1	M_0	M_1	M_0	M_1
Delaunay	pos+dry mass	8602.2188	154550.1678	92.7481	393.1287	60.3135	273.4477	0.0958	0.3878
	pos+dry mass+ diameter	62312.2560	18173.5230	249.6242	134.8092	180.5944	**84.2163**	0.2750	0.1391
Distance	pos+dry mass	11050.0424	145480.1189	105.1192	381.4186	61.5839	252.4445	0.0953	0.3551
	pos+dry mass+ diameter	**4225.2625**	63847.8685	**65.0020**	252.6814	**43.8038**	160.0039	**0.0777**	0.2172
Gabriel	pos+dry mass	10904.0880	**16725.0930**	104.4226	**129.3255**	71.4198	86.4600	0.1150	**0.1329**
	pos+dry mass+ diameter	19370.5186	195170.2022	139.1780	441.7807	95.1790	290.9992	0.1561	0.4106

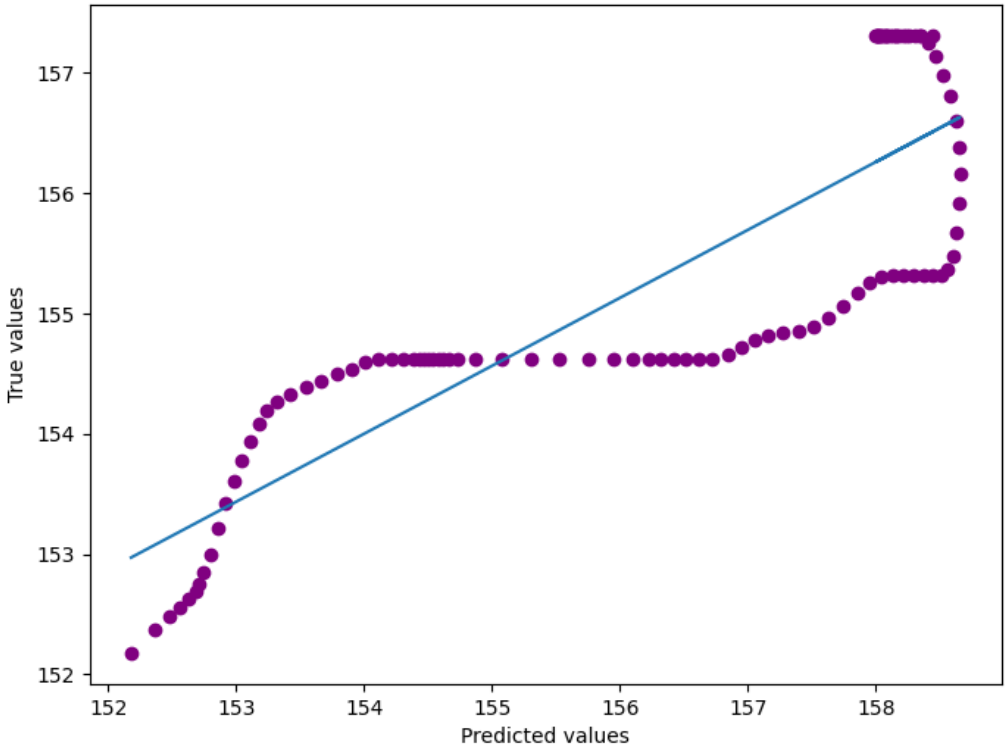


(a) Cell 1

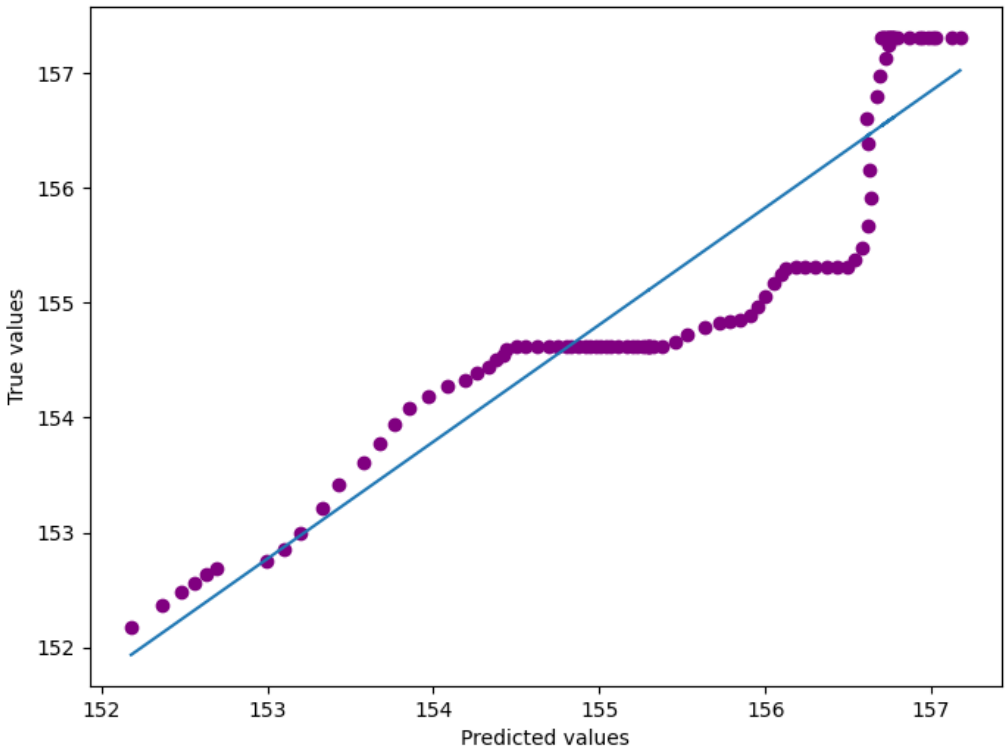


(b) Cell 2

Figure 4.9: Outcomes obtained for two cells on 1st simulation

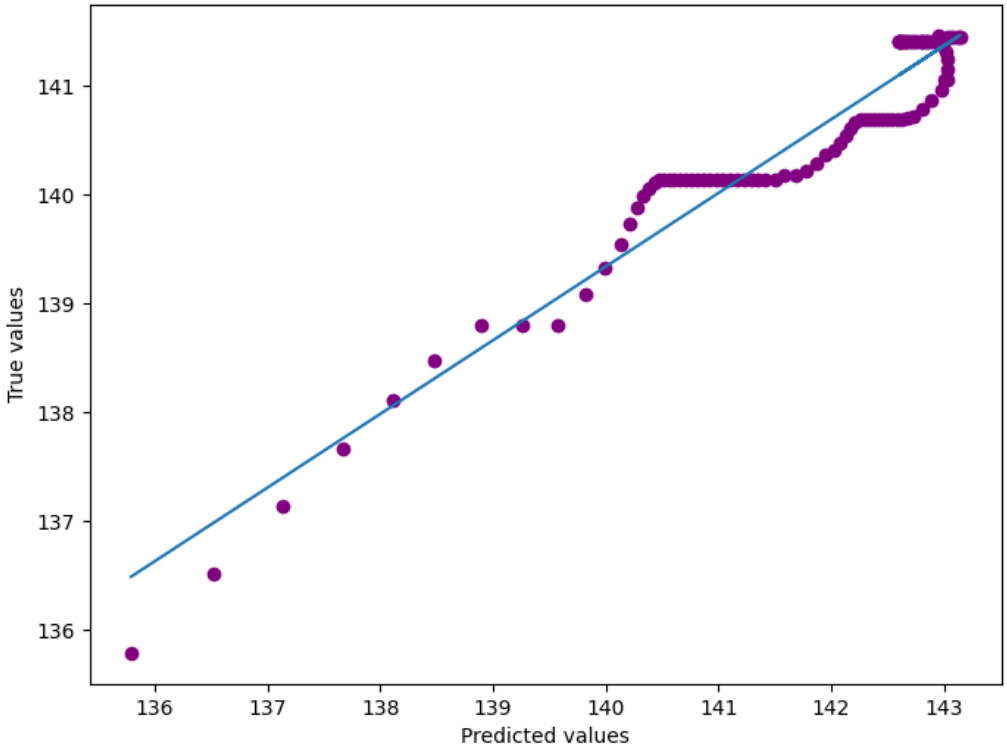


(a) Model LS

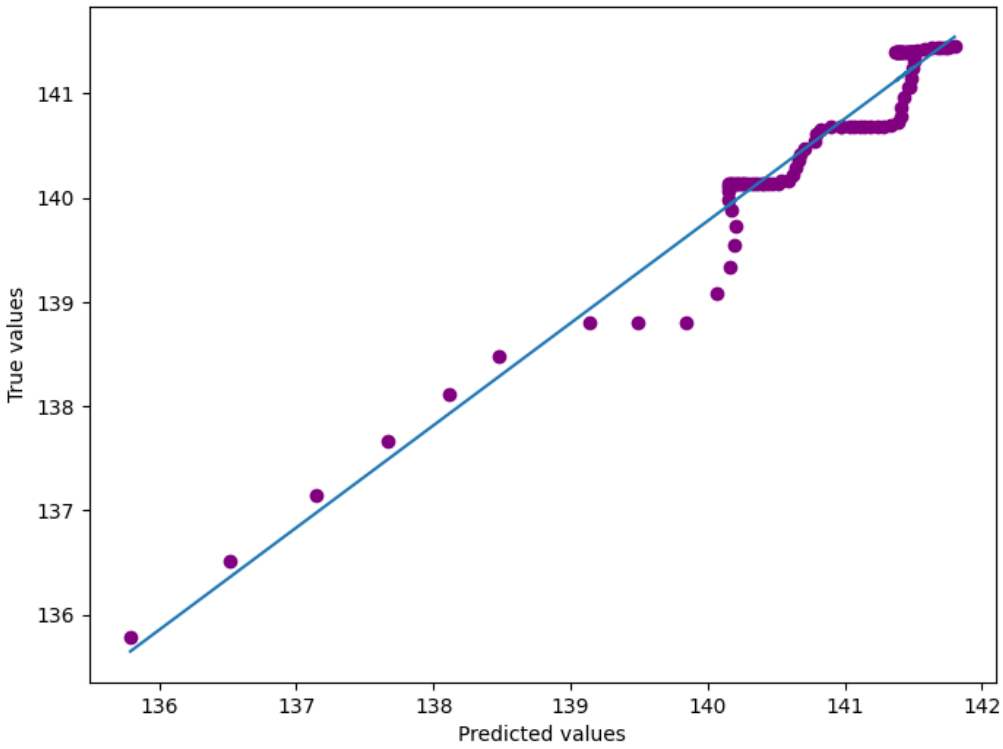


(b) Model LS MAGIK

Figure 4.10: Comparison between ground truth and predicted dry mass values for Cell 1 from simulated data



(a) Model LS



(b) Model LS MAGIK

Figure 4.11: Comparison between ground truth and predicted dry mass values for Cell 2 from simulated data

4.4.0.2 Experimental data

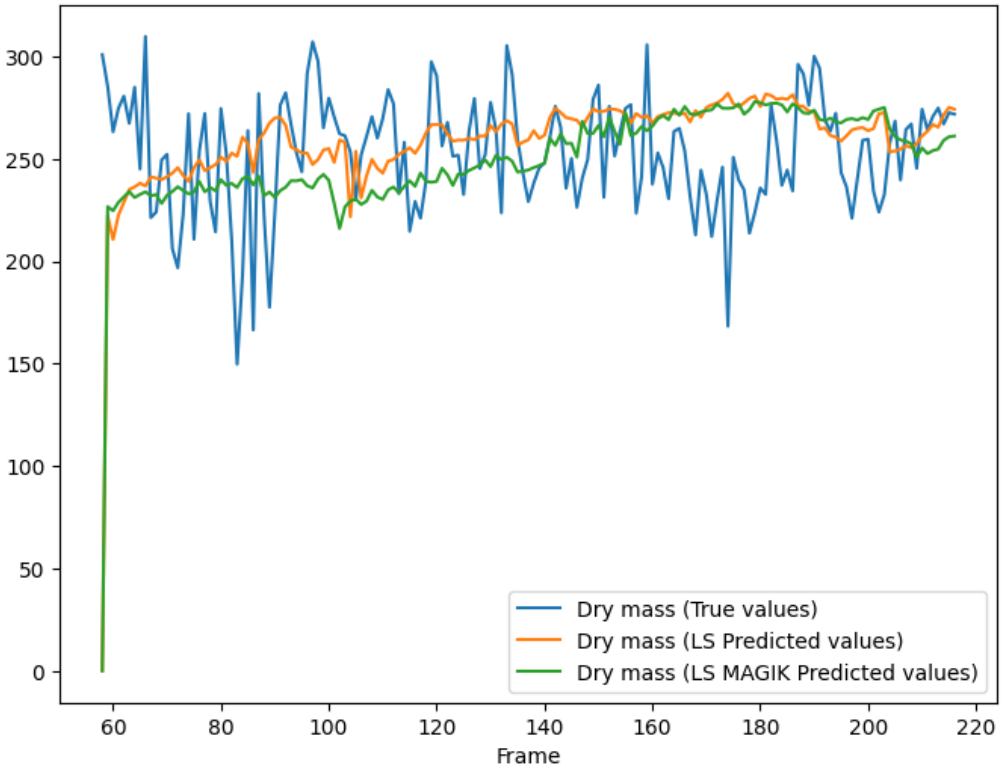
Table 4.4 illustrates the evaluation metrics for all cells, along with different node feature configurations and graph structures for HeLa test dataset. As seen before, it is noticed that there are still high values for all evaluation metrics across all graphs and node features tested. For the HeLa test dataset, the best performance for the models is achieved when considering x-y position, dry mass, perimeter, and eccentricity as node features and using the Gabriel graph. In this case, model $\boldsymbol{M_1}$ generates the lowest high values, with a slight difference from the values obtained from model $\boldsymbol{M_0}$. The outcomes obtained imply that the forecasting of dry mass values only depends on features related to the shape and position of cells.

Figures 4.12a and 4.12b illustrate the true and predicted dry mass values across the two models considered for two cells from the HeLa dataset. It is noted that the two selected cells exhibit high variation in their true dry mass values over time. Additionally, they do not display a clear growth trend for the dry mass, which could present a significant challenge for the models to learn and forecast this behavior in the cell culture. However, both models attempted to forecast the trend of the two cells without success, as the predictions obtained by the models do not visually align with the ground truth data or its trend.

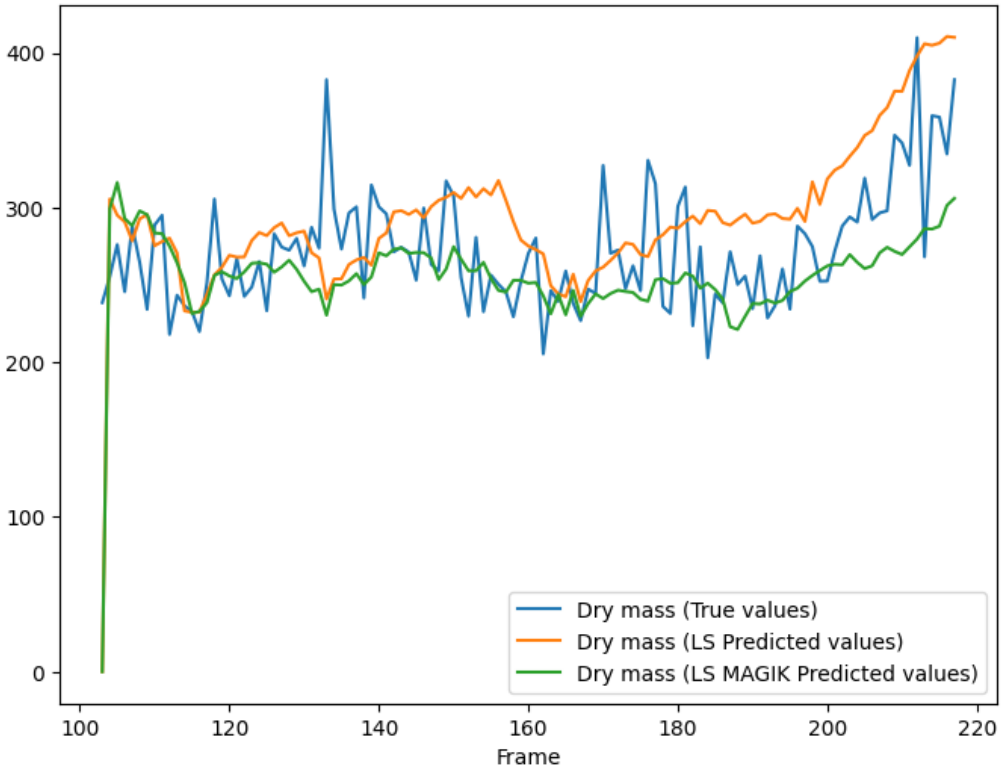
Figures 4.13a, 4.13b, 4.14a and 4.14b, depict scatter plots of the true versus predicted dry mass values over time for two cells using the two models, $\boldsymbol{M_0}$ and $\boldsymbol{M_1}$. These figures reveal that neither model was able to establish a linear regression between the ground truth and predicted values. Therefore, the proposed models did not adequately capture the behavior of the neighboring cells to predict the dry mass over time, despite evidence of spatial autocorrelation in the dry mass. This suggests that either the data utilized, the graph structures considered, or the architecture of the models employed were not suitable for addressing the prediction task. Consequently, further research into better graph structures and models is needed to improve the performance obtained.

Table 4.4: Comparison of models' performance per features and graphs from HeLa test data. The lowest value for each performance metric is highlighted in black.

		MSE		**RMSE**		**MAE**		**MAPE**	
Graph	**Node features**	M_0	M_1	M_0	M_1	M_0	M_1	M_0	M_1
Delaunay	pos+dry mass	1669.1697	2154.8358	40.8555	46.4202	11.8370	14.4216	0.0417	0.0506
	pos+dry mass+area	1126.6269	1065.3187	33.5653	32.6392	9.4632	9.1248	0.0324	0.0307
	pos+dry mass+perimeter	1067.4485	1113.0043	32.6718	33.3617	8.2578	8.6496	0.0265	0.0286
	pos+dry mass+perimeter+ eccentricity	808.6325	796.7184	28.4365	28.2262	6.6047	6.5849	0.0208	0.0205
	pos+dry mass+perimeter+ length	1346.8577	1473.9170	36.6996	38.3916	10.3985	11.0376	0.0305	0.0320
	pos+dry mass+perimeter+ area+length	1150.2383	1331.7541	33.9152	36.4932	9.5977	10.4436	0.0330	0.0365
Distance	pos+dry mass	1771.1144	2024.0829	42.0846	44.9898	12.2447	13.6152	0.0432	0.0480
	pos+dry mass+area	1149.4438	1354.5927	33.9034	36.8048	9.5649	10.5456	0.0329	0.0369
	pos+dry mass+perimeter	808.0696	900.3357	28.4266	30.0056	6.7242	7.4368	0.0203	0.0235
	pos+dry mass+perimeter+ eccentricity	843.0486	778.0984	29.0353	27.8944	6.9483	6.4478	0.0222	0.0196
	pos+dry mass+perimeter+ length	1117.1267	1289.5569	33.4234	35.9104	9.2928	10.1241	0.0284	0.0298
	pos+dry mass+perimeter+ area+length	1165.9154	1183.3163	34.1455	34.3994	9.6662	9.7091	0.0334	0.0334
Gabriel	pos+dry mass	1783.1223	1890.6472	42.2270	43.4816	12.2736	13.0530	0.0433	0.0459
	pos+dry mass+area	1120.2384	1191.6161	33.4700	34.5198	9.4032	9.7428	0.0321	0.0336
	pos+dry mass+perimeter	982.5910	810.7861	31.3463	28.4743	7.8601	6.7583	0.0252	0.0208
	pos+dry mass+perimeter+ eccentricity	**772.1416**	**757.5868**	**27.7874**	**27.5243**	**6.3828**	**6.3405**	**0.0200**	**0.0194**
	pos+dry mass+perimeter+ length	1013.3132	1126.4373	31.8326	33.5624	8.8314	9.3070	0.0285	0.0282
	pos+dry mass+perimeter+ area+length	1271.1357	1600.3261	35.6530	40.0041	10.1722	11.7152	0.0356	0.0413

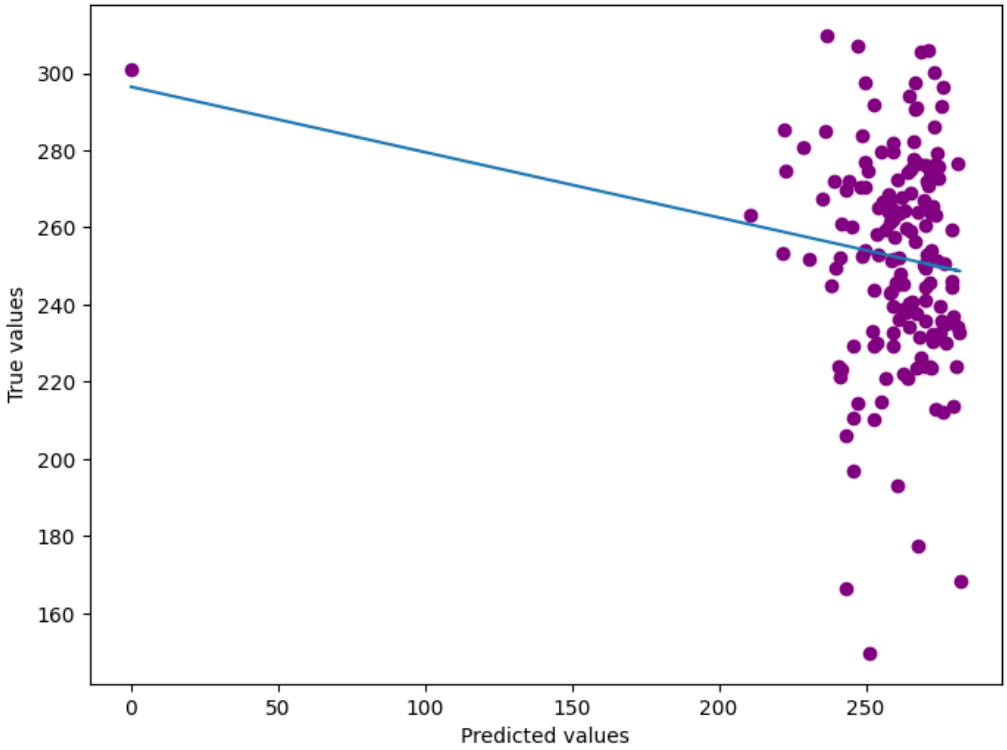


(a) Cell 1

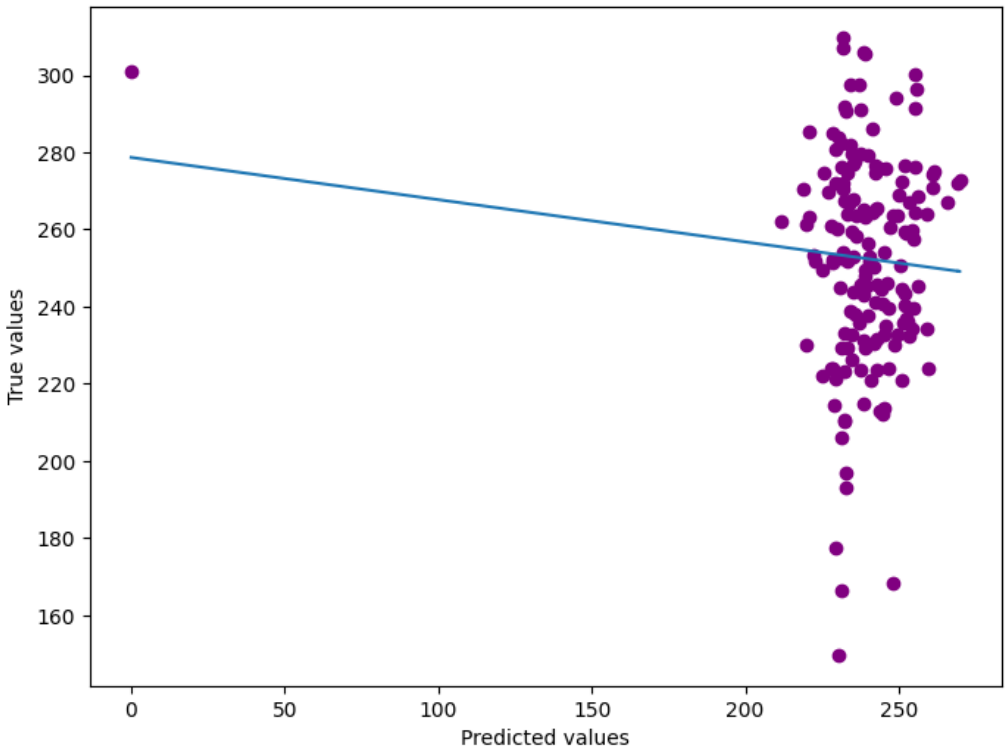


(b) Cell 2

Figure 4.12: Outcomes obtained for two cells on HeLa dataset

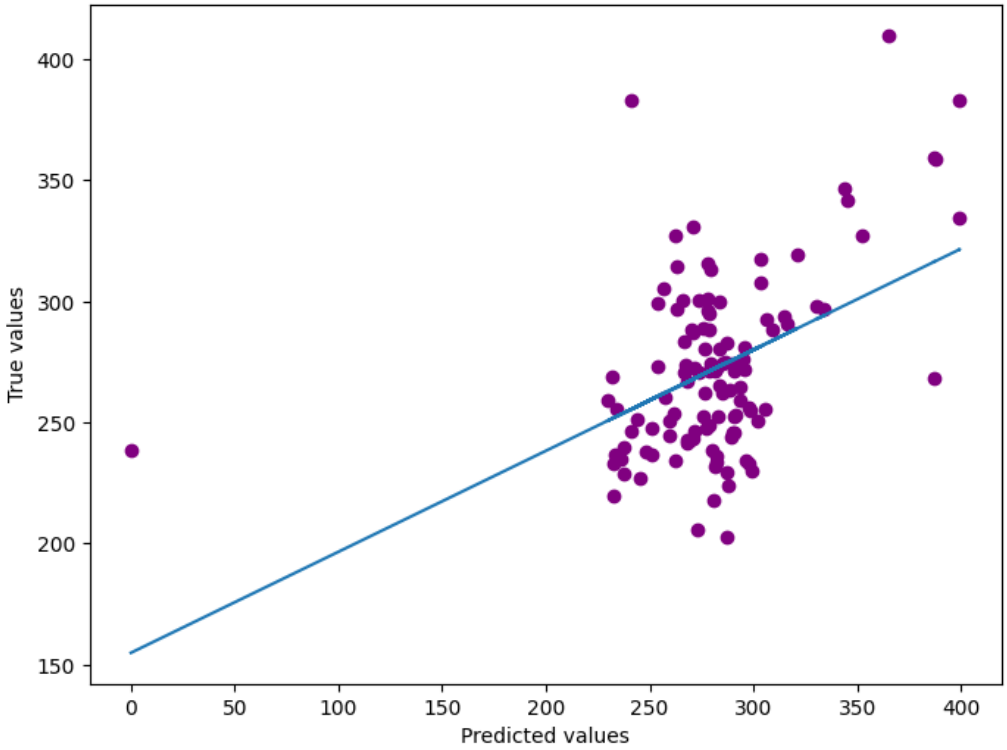


(a) Model LS

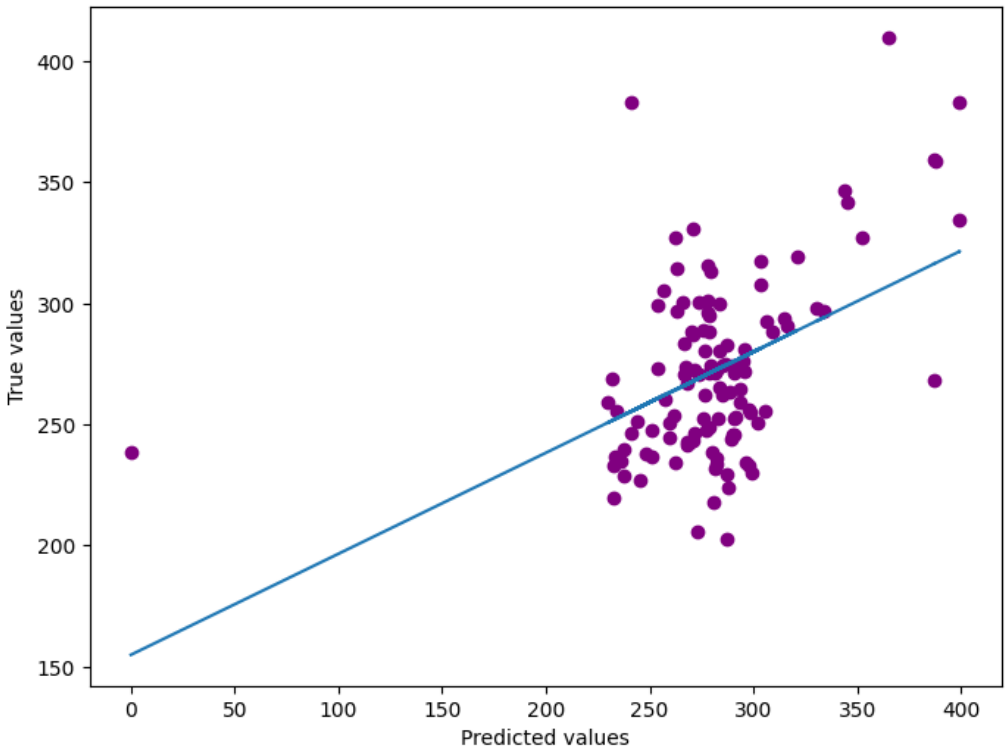


(b) Model LS MAGIK

Figure 4.13: Comparison between ground truth and predicted dry mass values for Cell 1



(a) Model LS



(b) Model LS MAGIK

Figure 4.14: Comparison between ground truth and predicted dry mass values for Cell 2

Conclusion

In summary, this study proposed an analysis of the influence of a cell's neighborhood on the growth of its dry mass by employing graphs through a regression task. Spatial autocorrelation confirmed the presence of a correlation between the neighborhood of cells and their dry mass values, which underscored the importance of utilizing graphs for this study. The results obtained did not provide clear and explicit evidence that a cell's neighborhood can help forecast its dry mass over time. However, it is not sufficient to draw a conclusion about the lack of utility of graph-based models for this task, suggesting further research on graph structures and the architecture of the models to better address the problem presented. One possible approach could be to utilize a Graph 3D structure that considers the entire time-lapse video. This could provide important insights on how to address the regression task from a different perspective to overcome the issues encountered.

Chapter 5

Conclusions and Perspectives

In conclusion, graphs were utilized as an alternative tool for analyzing 2D cell microscopic videos. Three different applications involving graph-based methods and machine learning approaches, including GNN models, were proposed. These graph-based methods and models aim to try to overcome the limitations presented in each addressed application. These limitations include the challenges associated with the segmentation and characterization of the dynamics of process formation in cellular networks, as well as the scarcity of data on neighboring cells and the difficulties in tracking cells during microscope image analysis.

Epithelial cells constitute fundamental tissues that are ubiquitous throughout the body. networks typically facilitate the growth of cellular tissues. The formation of these tissues results from the self-assembly processes of epithelial cells. Epithelial networks exhibited diverse structural behaviors in their development under different initial conditions considered.

Data processing of these cellular networks, which consist of 2D microscopic images, was not feasible through a single method for segmentation. This required the development of various methods for an initial segmentation on images. Consequently, a deep learning model was proposed to integrate all different methods into a single, unified approach. The deep learning-based segmentation model demonstrates significant performance during both the training and inference phases, exhibiting the ability to generalize over unseen data. This enabled a transition to the subsequent analysis, where graphs, which were constructed from the segmented images, played a key role through their metrics in distinguishing all different conditions and denoting at most four states that characterize the biological network formation process over the time-lapse sequence. In particular, graph metrics were identified to reveal and discern the actomyosin regulation of epithelial networks over time and under various drug treatments. Additionally, by pruning the graphs, it is possible to obtain simpler graphs that retain the primary shape and structures (meshes) of the networks.

An alternative approach was considered for the segmentation phase. This method involved retraining ERnet on the initial segmentation results. However, this approach was not pursued further due to insufficient GPU memory required to handle the model. Nevertheless, it would be worthwhile to explore this method and compare its outcomes with those obtained from the proposed model.

Single-cell analysis is a powerful approach for exploring heterogeneity within biological systems by examining individual cells and cell-to-cell variation across various contexts, such as organs, tissues, and cell cultures. This process involves genomic and transcriptomic profiling of cells. In microscope image analysis, single-cell analysis

includes detecting cells using tools like lens-free microscopy, segmenting them via deep-learning methods, extracting features from each cell, and applying these features to achieve specific biological objectives. However, single-cell analysis in imaging microscopy faces several limitations. Some of them, include the lack of data on neighboring cells, which could otherwise be necessary for better characterizing single cells, and the difficulty of tracking cells, often resulting in errors when identifying and monitoring cells over time.

Graphs have been proposed as a primary method to address these limitations. This methodology is facilitated by the development of a spatio-temporal graph, denoted Graph 3D, that exploits both spatial and temporal connections. This innovative graph structure combines spatial and temporal proximities to derive unique features at the node level. This application utilizes datasets comprising time-lapse 2D microscope images of cell cultures and corresponding simulations of objects exhibiting similar temporal evolution. Two learning-based prediction tasks were developed: identifying cell types in two distinct populations and two mixed populations. To tackle these tasks, both unsupervised and supervised methods were employed, utilizing extracted node-level attributes. The unsupervised methods included a Gaussian Mixture Model (GMM) and the He2CL algorithm, the latter designed specifically for the cell cultures utilized. For supervised learning, a specialized Graph Neural Network (GNN)-based model was used. The results demonstrate that unsupervised methods perform well by incorporating node-level attributes in both experimental and simulated data for the first prediction task, which involves identifying cell types in two distinct populations. This underscores the ability to identify cell types without the need for tracking individual cells.

For the second prediction task, which involves identifying cell types in two mixed cell populations, the Gaussian Mixture Model (GMM) exhibited poor performance on simulated data. Subsequently, a Graph Neural Network (GNN)-based model was tested on this simulated dataset and achieved significant performance in the second prediction task. However, this performance was not replicated on experimental data, where the GNN-based model showed poor results. This discrepancy may be attributed to the challenges associated with data acquisition, particularly in accurately detecting each type of cell. These difficulties limited the ability to discern the full potential of Graph 3D for the second prediction task.

In the final application, graphs were employed to analyze the influence of a cell's neighborhood on its dry mass over time. In this scenario, 2D cell microscopic videos and their associated tracking information were utilized. Graphs were constructed based on the spatial proximity between cells, and node features were derived from the tracking information, including 2D position and cellular morphological features. GNN-based have also been proposed to address this task. The performance of most of these models was evaluated using the predictions of dry mass over time, which were obtained from a dynamic approach known as Rollout. Results indicate that the proposed models estimated the trend of dry mass in simulations. However, they struggled to replicate this behavior with experimental data. In essence, it was not possible to successfully demonstrate this biological behavior, despite the existence of literature that supports it.

It is also crucial to emphasize the importance of accurately constructing graphs, particularly through the creation of edges, for precise analysis in the prediction tasks addressed. This aspect was not a concern in the study of epithelial networks, where graphs were automatically extracted from segmented images. In contrast, for the

other two applications examined, edges in the graphs were constructed by leveraging the spatial (and temporal, for Graph 3D) proximity between cells.

In an overall view, graphs have been demonstrated to be a powerful tool for conducting exploratory and quantitative analyses derived from 2D videos of cell cultures. When combined with image processing techniques, these analyses highlight the efficacy of the proposed methodology for the applications described. The potential of graphs and the models derived from them is far from being fully exploited and extends even further, motivating continuous research into the utilization of this mathematical tool across diverse domains.

Chapter 6

Annexes

6.1 Segmentation process on cellular tissues

The procedure employed to segment cellular tissues-based image data through a deep learning approach is described here. The ECMstiff dataset was used to train the model.

6.1.1 Initial segmentation

To treat the image data, five segmentation types were defined. The ERnet model used is based on a pre-trained SwinIR network [171], which is available for download on the corresponding GitHub repository[1]. For two conditions, the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm [303] was utilized to enhance the contrast of images.

An adaptive thresholding algorithm was also applied, which determines the threshold for a pixel based on a small region around it. This means that different thresholds are used for different regions of the same image. The threshold value is computed as the mean of the neighborhood area minus a constant C, with the size of the neighborhood area set at 349 pixels and $C = 3$.

Five segmentation types are described as follows:

1. **Otsu:** For this segmentation, images are first normalized using the min-max method. They are then segmented through a threshold value determined by Otsu's algorithm [204].

2. **ERnet:** The images are first downsampled to half of their original size (1920×1374 pixels). A bilateral filter is then applied to them. The images are normalized using the min-max method and their intensity is rescaled. Subsequently, the pre-trained ERnet model is used to segment the images, followed by a closing operation. Finally, the images are upsampled to their original size using nearest-neighbor interpolation.

3. **Adaptive threshold:** Images are first normalized using the min-max method and their intensity is rescaled. An adaptive threshold method is then applied to segment them, followed by the removal of isolated pixels. Finally, a dilation operation is applied three times.

[1] https://github.com/charlesnchr/ERnet-v2/releases/tag/v2.0

4. **Adaptive threshold+:** A bilateral filter is first applied to the images. The images are then normalized using the min-max method and their intensity is rescaled. An adaptive threshold method is applied to segment the images, followed by the removal of isolated pixels. Finally, a dilation operation is performed three times, followed by a closing operation three times.

5. **CLAHE+rescaling:** A bilateral filter is first applied to the images. Subsequently, the images are normalized using the min-max method. The CLAHE algorithm is then applied to enhance the contrast and rescale the intensity of the images. Following this, the images are segmented using a threshold value determined by Otsu's algorithm. Finally, an opening operation is performed.

These segmentation types were applied for each video (or condition) according to the process described below, where the index i corresponds to the i-th frame image:

- **Condition 1:**
  - For $i \leqslant 150$: **Otsu**
  - For $150 < i \leqslant 519$: **ERnet**
  - For $519 < i \leqslant 900$: **Adaptive threshold**
  - For $i > 900$: **Adaptive threshold+**
- **Condition 2:**
  - For $i \leqslant 150$: **Otsu**
  - For $150 < i \leqslant 579$: **ERnet**
  - For $579 < i \leqslant 900$: **Adaptive threshold**
  - For $i > 900$: **Adaptive threshold+**
- **Condition 3:**
  - For $i \leqslant 100$: **Otsu**
  - For $100 < i \leqslant 399$: **ERnet**
  - For $399 < i \leqslant 739$: **Adaptive threshold**
  - For $i > 739$: **Adaptive threshold+**
- **Condition 4:**
  - For $i \leqslant 109$: **Otsu**
  - For $109 < i \leqslant 449$: **ERnet**
  - For $449 < i \leqslant 999$: **Adaptive threshold**
  - For $i > 999$: **Adaptive threshold+**
- **Conditions 5 and 6:**
  - For $i \leqslant 100$: **Otsu**
  - For $100 < i \leqslant 400$: **ERnet**
  - For $i > 400$: **CLAHE + rescaling**

6.1.2 Training-validation data split

After image segmentation for each condition, some of the images are selected as ground-truth data to build the training and validation datasets, respectively. This process is illustrated below

- **Condition 1:** frames 150 to 900.
- **Conditions 2, 3 and 4:** frames 150 to 1100.
- **Conditions 5 and 6:** frames 150 to 1557.

For all conditions, images were taken from frames: 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Subsequently, a training-validation dataset split was generated. Data from videos 1, 2, 3, 5, and 6 were used as **training data**, while data from video 4 were used as **validation data**.

Additionally, cropped images of size 512×512 pixels (48 crops per image) were generated from the training and validation data as input data during the training phase. The model's performance was evaluated on the original-sized images. A total of 1050 images from the training data and 210 images from the validation data were selected, resulting in 50400 cropped images for training and 10080 cropped images for validation.

6.1.3 Model architecture

This part describes the network employed to achieve the final segmentation. The model is based on a U-Net [227] architecture, which is a classical neural network for image segmentation tasks. In Figure 6.1, an overview of this deep neural network is provided. It is composed of two parts: the encoder, which encodes input images into a lower-dimensional latent space to retain relevant information, and the decoder structure, which retrieves the original-sized images from the latent space. The encoder network is a contracting path, composed of a series of convolutional layers (followed by ReLU activation functions) and pooling layers that gradually decrease the resolution of the feature map in the image. The decoder network is an expansive route that consists of a succession of up-sampling (transposed convolution) and convolutional layers (followed by ReLU activation functions) that gradually enhance the resolution of the feature map. At each step in the decoder, the relevant feature map from the encoder is concatenated with the current feature map. Finally, a sigmoid activation function is applied on the last output feature map in the decoder.

6.1.4 Model training

To train the model, the loss function chosen is the Binary Cross-Entropy (BCE), which is defined as:

$$BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} [y_{i,j} \log \hat{y}_{i,j} + (1 - y_{i,j}) \log(1 - \hat{y}_{i,j})],$$

where m and n correspond to the size of 2D images ($m \times n$ pixels), $\boldsymbol{y} = (y_{1,1}, \ldots, y_{m,n})$ is the ground-truth image segmentation mask, and $\hat{\boldsymbol{y}} = (\hat{y}_{1,1}, \ldots, \hat{y}_{m,n})$ is the predicted mask from the deep learning model. Then, the model hyperparameters involved

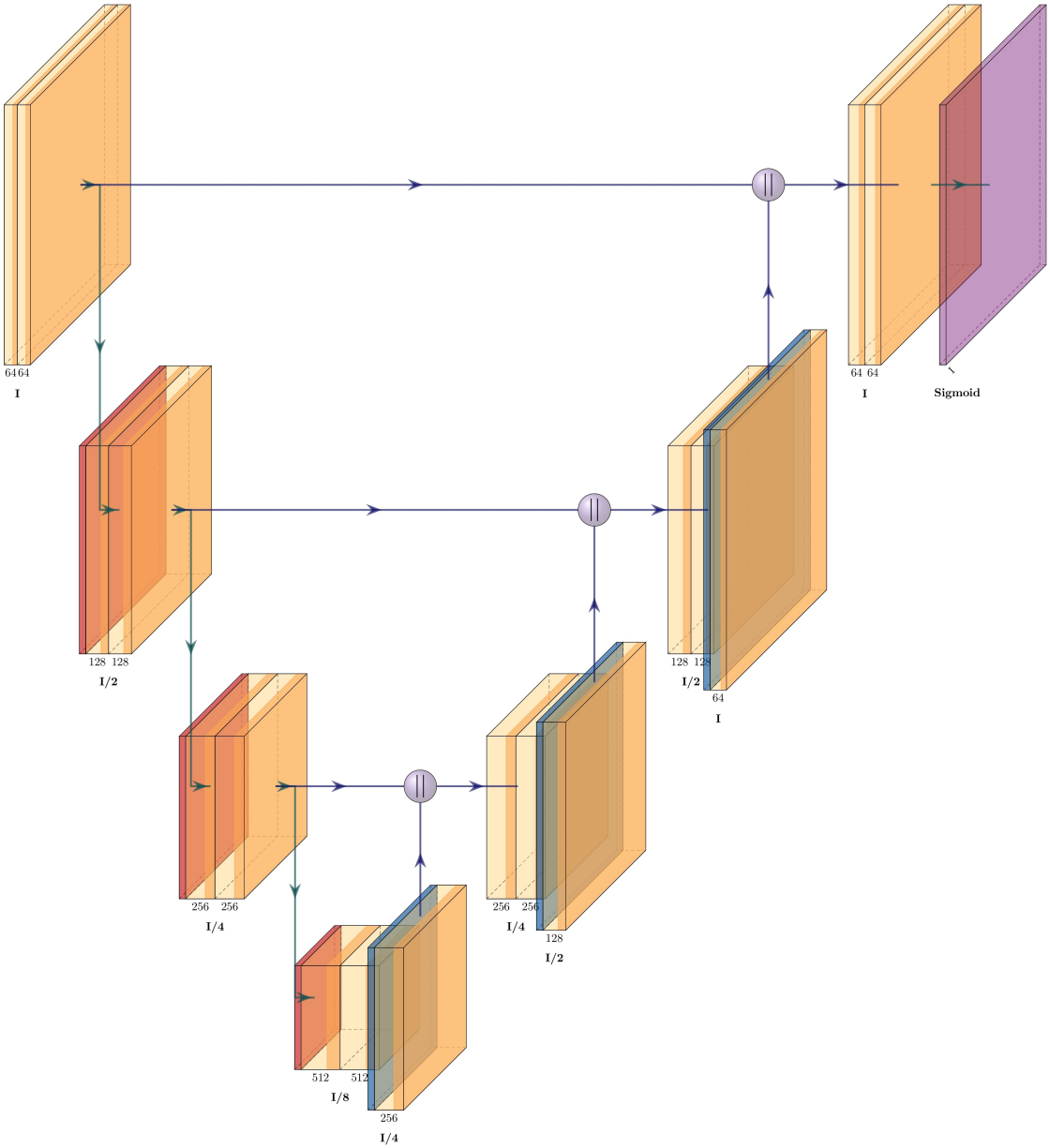


Figure 6.1: U-Net model architecture. The model is composed of two parts: the encoder and the decoder. Each step in the encoder and the decoder part have two 3×3 convolutional layers (orange boxes) followed by ReLU activation functions. A 2×2 max pooling layer (red box) is applied to down-sample the feature map at each step in the encoder. While the decoder uses a 2×2 up-convolution, called transposed convolution (dark blue-green box), at each step to up-sample the feature map. Additionally, at each step in the decoder part the corresponding feature map of the encoder is concatenated with the current feature map. Finally, a sigmoid activation function (magenta box) is applied on the last output feature map in the decoder.

during the training phase should be defined. The method AdamW [177] was chosen as the optimizer algorithm during the training process, with a learning rate (LR) of $1e-3$ and a weight decay of $5e-4$. Furthermore, an exponential LR scheduler was applied to improve the update of model parameters during the backpropagation phase. At each training epoch, image batches of size 8 were taken from the total cropped images. The training phase was stopped by defining an early stopping criterion: if the validation loss did not improve after 10 epochs, the model stopped training. Finally, the model was trained over 100 epochs, and the best model with respect to the lowest validation loss acquired was kept.

6.2 Graph clustering on simulated data

6.2.1 Analysis of two distinct simulations

First, the graph structures employed in the analysis of two distinct simulations, derived from simulated data, are presented. This elucidation aims to facilitate an understanding of the terminology employed for the graphs utilized and presented in the subsequent tables within this section.

It is important to note that τ_{sp} refers to the threshold value set for spatial connections, and τ_{temp} corresponds to the threshold value for temporal connections between two consecutive frames. The following graph structures were considered:

- **Delaunay:** refers to Delaunay graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 30\,\mu$m.
- **Gabriel:** refers to Gabriel graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 30\,\mu$m.
- **Urquhart:** refers to Urquhart graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 30\,\mu$m.
- **dist_50:** refers to Distance graph with $\tau_{sp} = \tau_{temp} = 50\,\mu$m.
- **dist_100:** refers to Distance graph with $\tau_{sp} = \tau_{temp} = 100\,\mu$m.
- **dist_150:** refers to Distance graph with $\tau_{sp} = \tau_{temp} = 150\,\mu$m.
- **dist_100_temp_20:** refers to Distance graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 20\,\mu$m.
- **dist_100_temp_30:** refers to Distance graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 30\,\mu$m.
- **dist_100_temp_50:** refers to Distance graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 50\,\mu$m.

Table 6.1: Mean accuracy score (10 times) on GMM for all possible combinations of two distinct simulations with regard to the graph structures

Data	gabriel	delaunay	urquhart	dist_50	dist_100	dist_150	dist_100 _temp_20	dist_100 _temp_30	dist_100 _temp_50
sim 1-2	0.6279	0.5916	0.6534	0.6879	0.5306	0.5493	0.5069	0.5560	0.5252
sim 1-3	0.5290	0.5161	0.5323	0.5412	0.5133	0.5133	0.5277	0.5136	0.5023
sim 1-4	0.5306	0.5228	0.5361	0.5238	0.5072	0.5081	0.5238	0.5336	0.5177
sim 1-5	0.6913	0.6992	0.7370	0.5915	0.5415	0.5030	0.9585	0.6675	0.7061
sim 1-6	0.9330	0.9314	0.9360	0.5695	0.7058	0.5475	0.9091	0.8932	0.8851
sim 1-7	0.5693	0.5632	0.5817	0.5146	0.5383	0.5029	0.9676	0.5877	0.6441
sim 1-8	0.7510	0.5983	0.7627	0.5128	0.5976	0.5117	0.9075	0.5952	0.5063
sim 2-3	0.5529	0.5564	0.5693	0.5188	0.5530	0.5230	0.5208	0.5459	0.5479
sim 2-4	0.5859	0.5611	0.5812	0.6086	0.5094	0.5511	0.5173	0.5525	0.5143
sim 2-5	0.6364	0.6511	0.6506	0.6094	0.5371	0.5628	0.9715	0.5045	0.5703
sim 2-6	0.7205	0.7363	0.7198	0.6227	0.6348	0.5932	0.8822	0.6414	0.6653
sim 2-7	0.6992	0.6769	0.6980	0.6489	0.6151	0.5691	0.9676	0.5428	0.5091
sim 2-8	0.6602	0.5359	0.8227	0.5904	0.5348	0.5184	0.9075	0.5526	0.5150
sim 3-4	0.5202	0.5011	0.5203	0.5631	0.5072	0.5111	0.5167	0.5079	0.5286
sim 3-5	0.7040	0.7042	0.6839	0.5750	0.5452	0.5102	0.9804	0.5641	0.6643
sim 3-6	0.7709	0.8003	0.8511	0.8004	0.6889	0.5556	0.9274	0.8324	0.8322
sim 3-7	0.6189	0.5673	0.6461	0.5621	0.5482	0.5125	0.9676	0.6465	0.5246
sim 3-8	0.6993	0.5846	0.7123	0.5256	0.5862	0.5005	0.9075	0.6456	0.6613
sim 4-5	0.7501	0.7327	0.7561	0.5101	0.5109	0.5070	0.9784	0.5367	0.5996
sim 4-6	0.8163	0.7975	0.9095	0.7231	0.6639	0.5355	0.8952	0.9074	0.8411
sim 4-7	0.5912	0.5815	0.5846	0.5171	0.5510	0.5086	0.9677	0.5079	0.5979
sim 4-8	0.7495	0.7184	0.7568	0.5230	0.5809	0.5062	0.9077	0.5820	0.6654
sim 5-6	0.5953	0.5963	0.5990	0.5239	0.5709	0.5226	0.6865	0.5996	0.5986
sim 5-7	0.6240	0.6230	0.6281	0.5318	0.5841	0.5077	0.5434	0.6147	0.5304
sim 5-8	0.5580	0.5533	0.5558	0.5237	0.5314	0.5024	0.5198	0.5504	0.5247
sim 6-7	0.7189	0.7256	0.7361	0.5608	0.5488	0.5496	0.8049	0.7425	0.7074
sim 6-8	0.6441	0.6462	0.6508	0.5348	0.5759	0.5348	0.6559	0.6383	0.6290
sim 7-8	0.5176	0.5153	0.6064	0.5234	0.5425	0.5019	0.5164	0.5835	0.5153

Table 6.2: Best mean accuracy score (10 times) on GMM for all possible combinations of two distinct simulations with regard to all possible node features combinations

Data	**Spatial and temporal node features**	**Spatial node features**	**Temporal node features**
sim 1-2	0.6879	0.6382	0.6675
sim 1-3	0.5412	0.5747	0.5900
sim 1-4	0.5361	0.5550	0.5948
sim 1-5	0.9585	0.6682	0.9560
sim 1-6	0.9360	0.8050	0.9632
sim 1-7	0.9676	0.6078	0.9676
sim 1-8	0.9075	0.6783	0.9075
sim 2-3	0.5693	0.5584	0.5769
sim 2-4	0.6086	0.5888	0.6409
sim 2-5	0.9715	0.6266	0.9553
sim 2-6	0.8822	0.6893	0.8806
sim 2-7	0.9676	0.6335	0.9676
sim 2-8	0.9075	0.6012	0.9075
sim 3-4	0.5631	0.5430	0.5685
sim 3-5	0.9804	0.6493	0.9780
sim 3-6	0.9274	0.7363	0.9021
sim 3-7	0.9676	0.6062	0.9676
sim 3-8	0.9075	0.6834	0.9075
sim 4-5	0.9784	0.6514	0.9743
sim 4-6	0.9095	0.7665	0.9256
sim 4-7	0.9677	0.5927	0.9677
sim 4-8	0.9077	0.6889	0.9077
sim 5-6	0.6865	0.5826	0.7271
sim 5-7	0.6281	0.6247	0.6954
sim 5-8	0.5580	0.5788	0.5893
sim 6-7	0.8049	0.6917	0.8393
sim 6-8	0.6559	0.6301	0.7808
sim 7-8	0.6064	0.5811	0.6082

Table 6.3: Best mean accuracy score (10 times) on a supervised method (SVM) for all possible combinations of two distinct simulations with regard to the possible combinations of features

Data	x-y position	Node features	x-y position and node features
sim 1-2	0.6819	0.9991	0.9994
sim 1-3	0.6905	0.9996	0.9999
sim 1-4	0.6374	0.9960	0.9975
sim 1-5	0.6571	0.9973	0.9987
sim 1-6	0.6932	0.9999	0.9999
sim 1-7	0.6556	0.9834	0.9899
sim 1-8	0.6437	0.9896	0.9952
sim 2-3	0.6950	0.9830	0.9953
sim 2-4	0.6698	0.9967	0.9996
sim 2-5	0.6995	0.9869	0.9975
sim 2-6	0.7178	0.9614	0.9974
sim 2-7	0.7033	0.9969	0.9979
sim 2-8	0.7297	0.9906	0.9994
sim 3-4	0.6566	0.9873	0.9954
sim 3-5	0.6797	0.9849	0.9957
sim 3-6	0.6808	0.9818	0.9953
sim 3-7	0.6565	0.9975	0.9972
sim 3-8	0.7271	0.9785	0.9945
sim 4-5	0.6572	0.9864	0.9958
sim 4-6	0.7218	0.9852	0.9993
sim 4-7	0.6327	0.9967	0.9945
sim 4-8	0.6932	0.9787	0.9874
sim 5-6	0.6870	0.9730	0.9944
sim 5-7	0.6669	0.9941	0.9986
sim 5-8	0.6870	0.9741	0.9933
sim 6-7	0.7056	0.9974	0.9975
sim 6-8	0.7336	0.9909	0.9988
sim 7-8	0.6974	0.9902	0.9935

6.2.2 Analysis of two mixed cell types

6.2.2.1 Results on GMM and SVM methods

Table 6.4: Kolmogorov-Smirnov test for simulated data with regard to the graph structures

Node feature	**gabriel**	**delaunay**	**urquhart**	**dist_50**	**dist_100**	**dist_150**	**dist_100 _temp_20**	**dist_100 _temp_30**	**dist_100 _temp_50**
spatial degree									
spatial betweenness centrality									
spatial closeness centrality						X			
spatial mean neighbor degree			X						
spatial mean shortest path	X	X	X		X	X	X	X	X
spatial clustering coefficient			X						
temporal degree									
temporal betweenness centrality									
temporal closeness centrality					X	X			
temporal mean neighbor degree									
temporal mean shortest path					X	X			

Table 6.5: Mean accuracy score (10 times) on GMM for the simulated data containing two cell types over all possible node features combinations

Graph	Spatial and temporal node features	Spatial node features	Temporal node features
gabriel	0.5082	0.5047	0.5150
delaunay	0.5449	0.5068	0.5175
urquhart	0.5324	0.5023	0.5177
dist_50	0.5524	0.5478	0.5108
dist_100	0.5040	0.5107	0.5039
dist_150	0.5065	0.5178	0.5060
dist_100_temp_20	0.5384	0.5107	0.5100
dist_100_temp_30	0.5788	0.5107	0.5178
dist_100_temp_50	0.5381	0.5107	0.5108

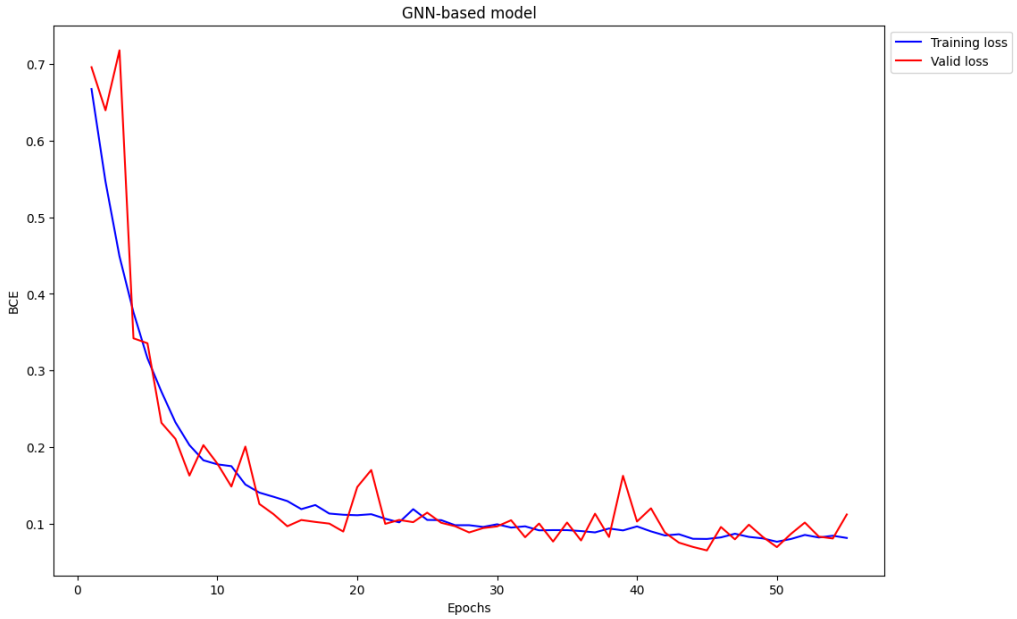
Table 6.6: Mean accuracy score (10 times) on a supervised method (SVM) for the simulated data, containing two cell types, regarding to all graph structures and the possible combinations of features

Graph	x-y position	Node features	x-y position and node features
gabriel	0.6176	0.6218	0.6557
delaunay	0.6117	0.6356	0.6704
urquhart	0.6113	0.6155	0.6507
dist_50	0.6116	0.5947	0.6311
dist_100	0.6091	0.5964	0.6323
dist_150	0.6108	0.6197	0.6440
dist_100_temp_20	0.6093	0.6739	0.7071
dist_100_temp_30	0.6139	0.6330	0.6614
dist_100_temp_50	0.6103	0.6236	0.6500

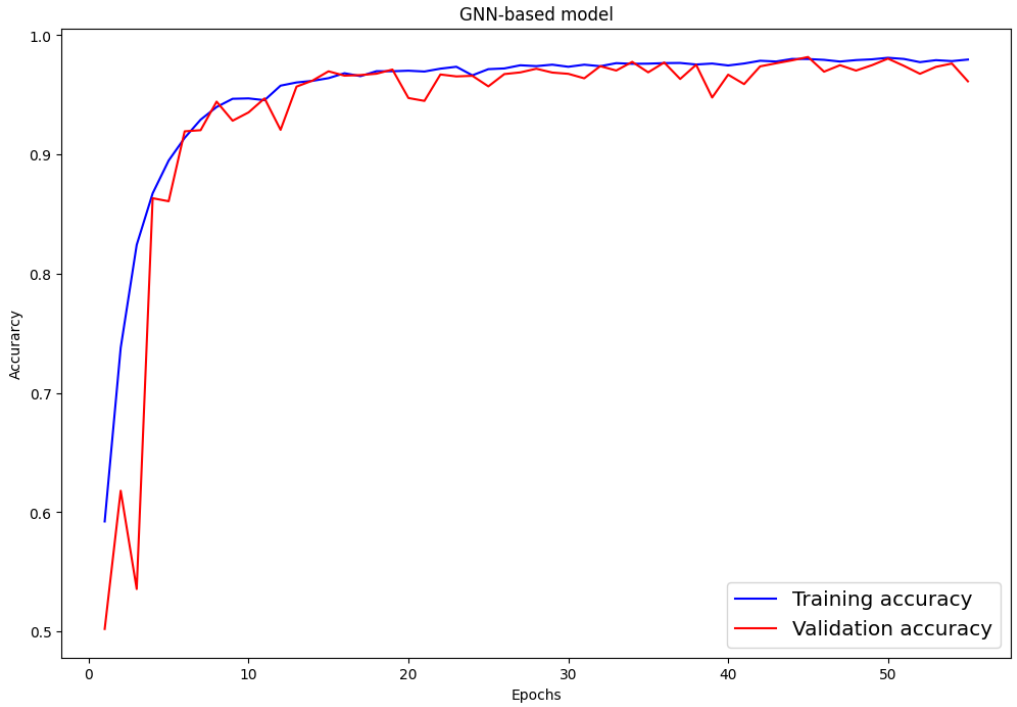
6.2.2.2 Results on GNN-based supervised classification

In addition, another set of graph structures was considered to test the proposed GNN-based model for the supervised classification task. These graph structures are described below:

- Delaunay graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 50\,\mu$m.
- Gabriel graph with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} = 50\,\mu$m.
- Distance graph with $\tau_{sp}, \tau_{temp} \in \{50\,\mu\text{m}, 100\,\mu\text{m}\}$.
- Distance graph with with $\tau_{sp} = 100\,\mu$m and $\tau_{temp} \in \{20\,\mu\text{m}, 30\,\mu\text{m}, 50\,\mu\text{m}\}$.



(a) Loss



(b) Accuracy

Figure 6.2: Loss and accuracy over epochs

Table 6.7: Model performance per GNN layers

GNN layers	**Mean accuracy**
1	0.9774
2	0.9815
3	0.9741
4	0.9769
5	0.9665

6.3 Graph clustering on experimental data

6.3.1 Histograms of node features for two distinct populations

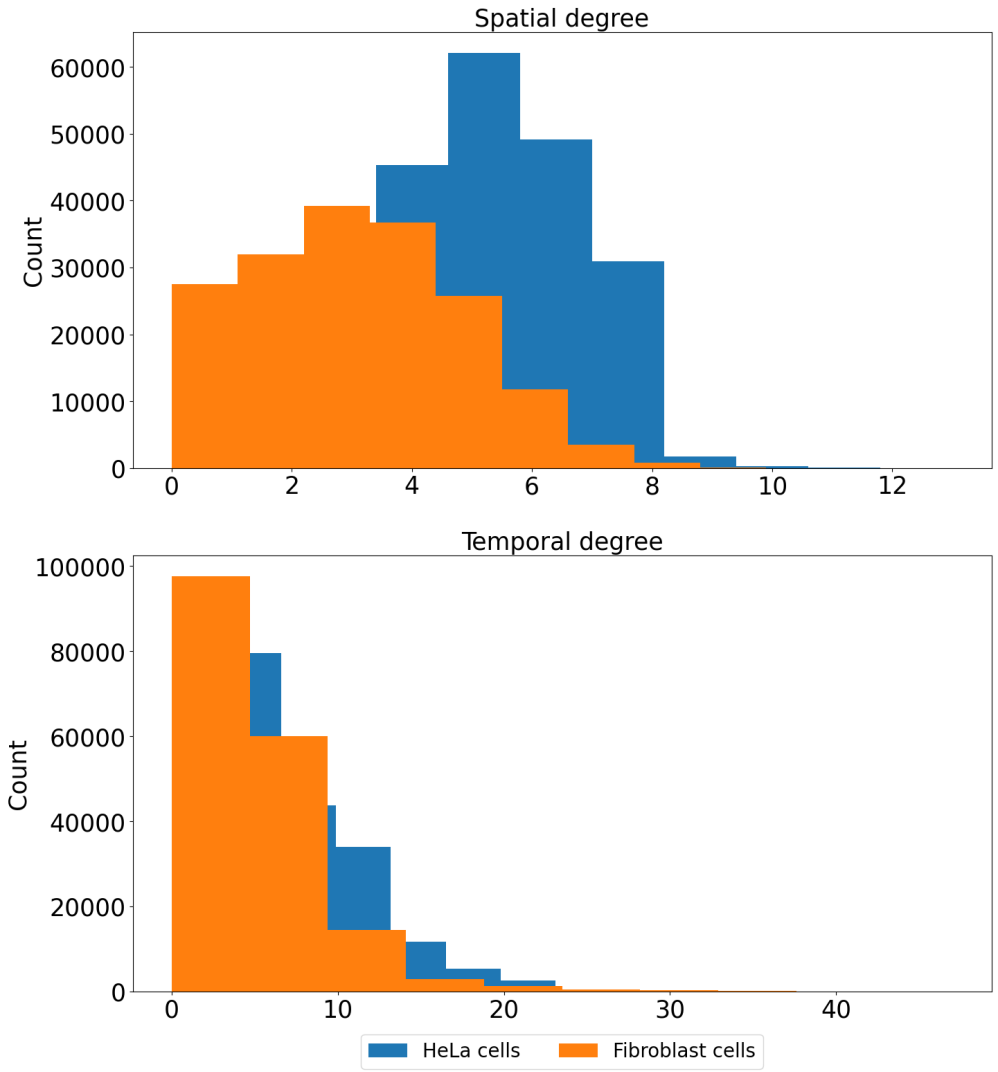


Figure 6.3: Histograms for spatial and temporal degree

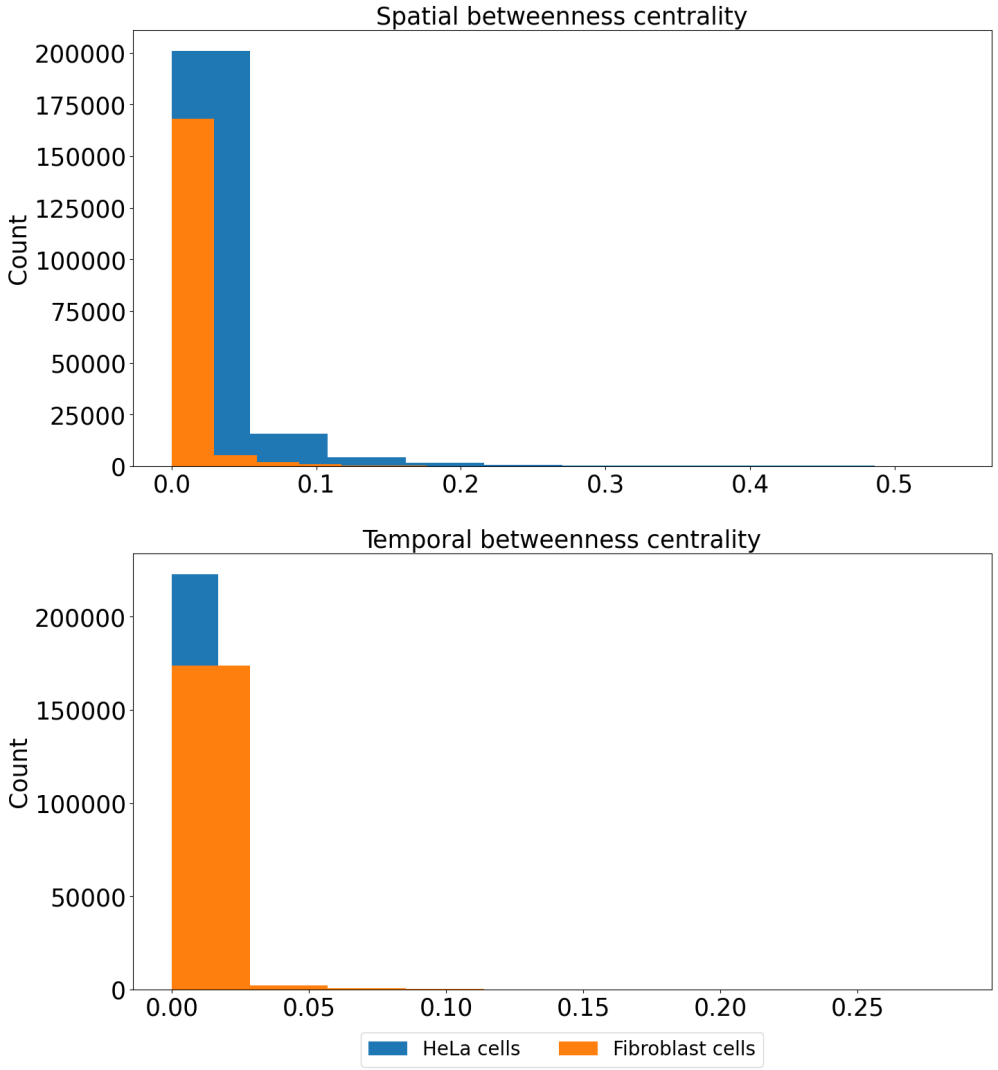


Figure 6.4: Histograms for spatial and temporal betweenness centrality

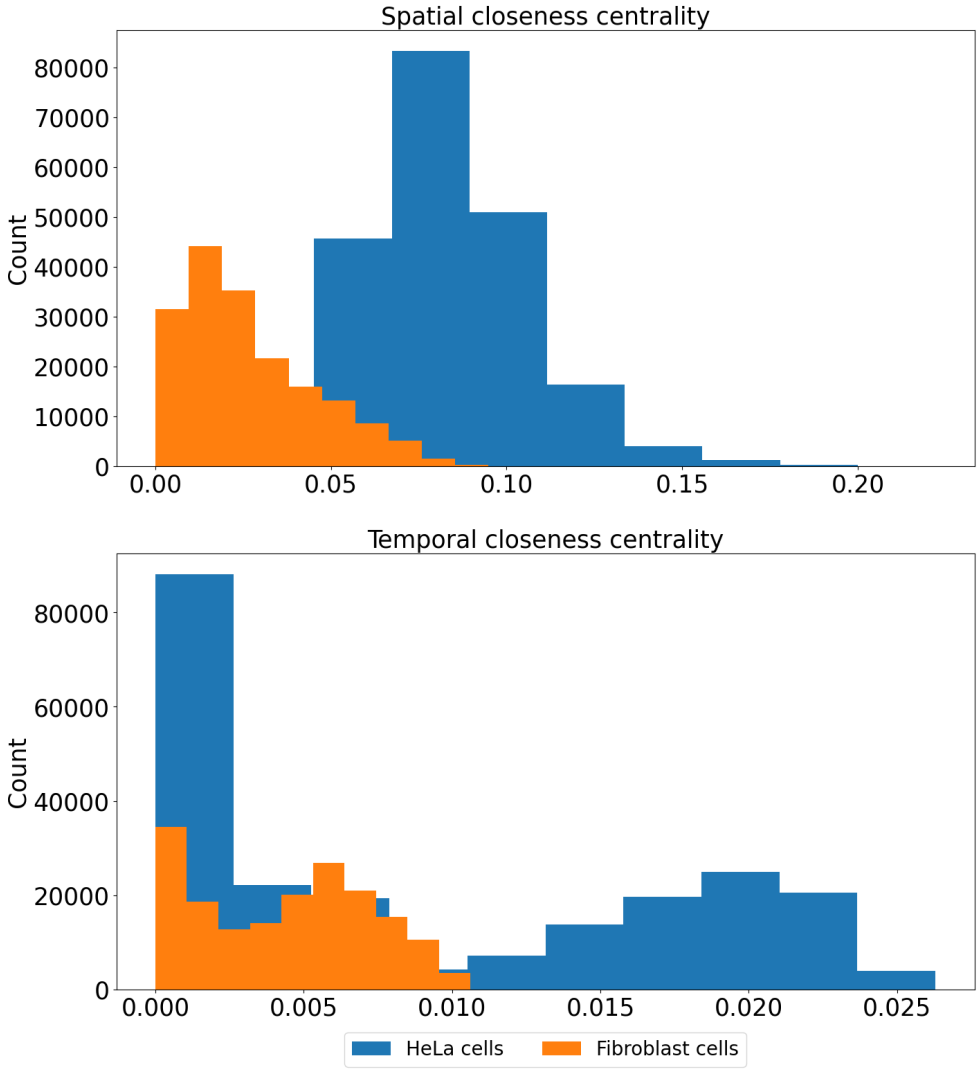


Figure 6.5: Histograms for spatial and temporal closeness centrality

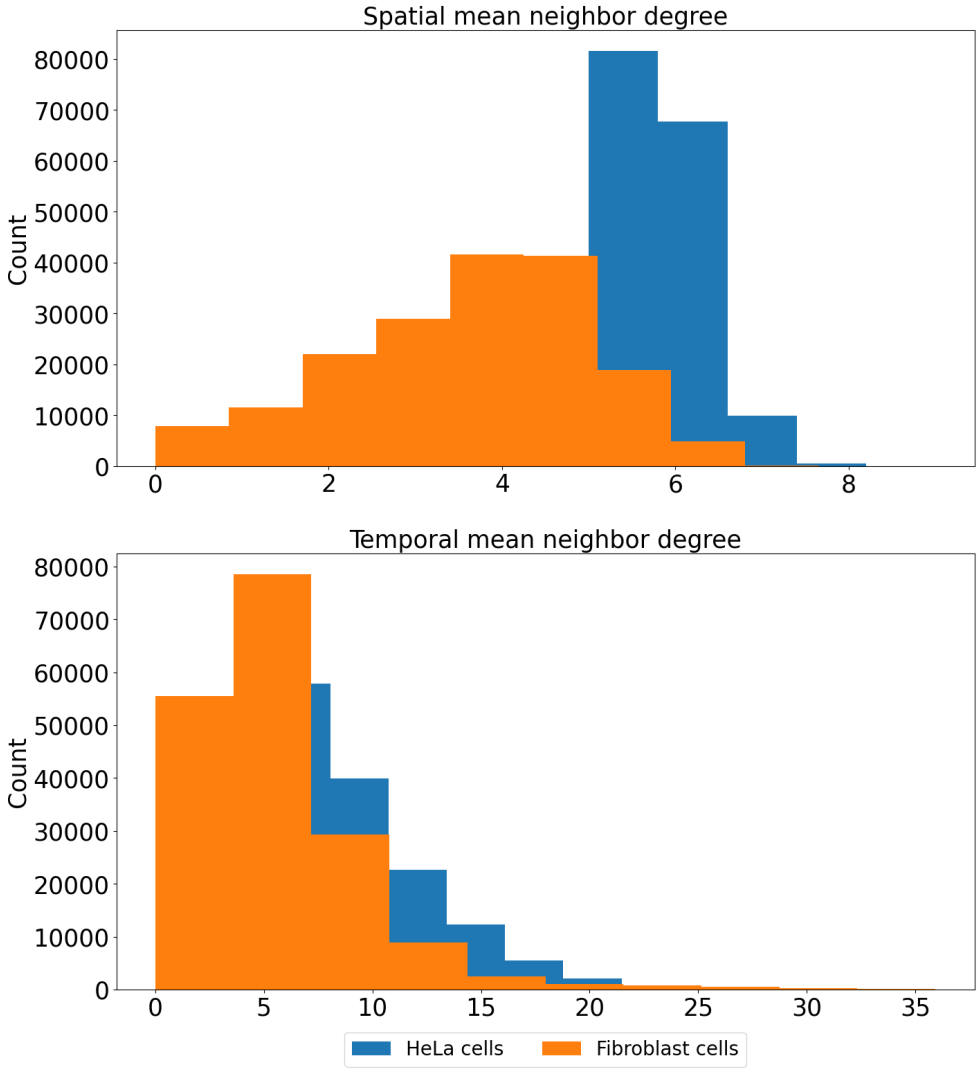


Figure 6.6: Histograms for spatial and temporal mean neighbor degree

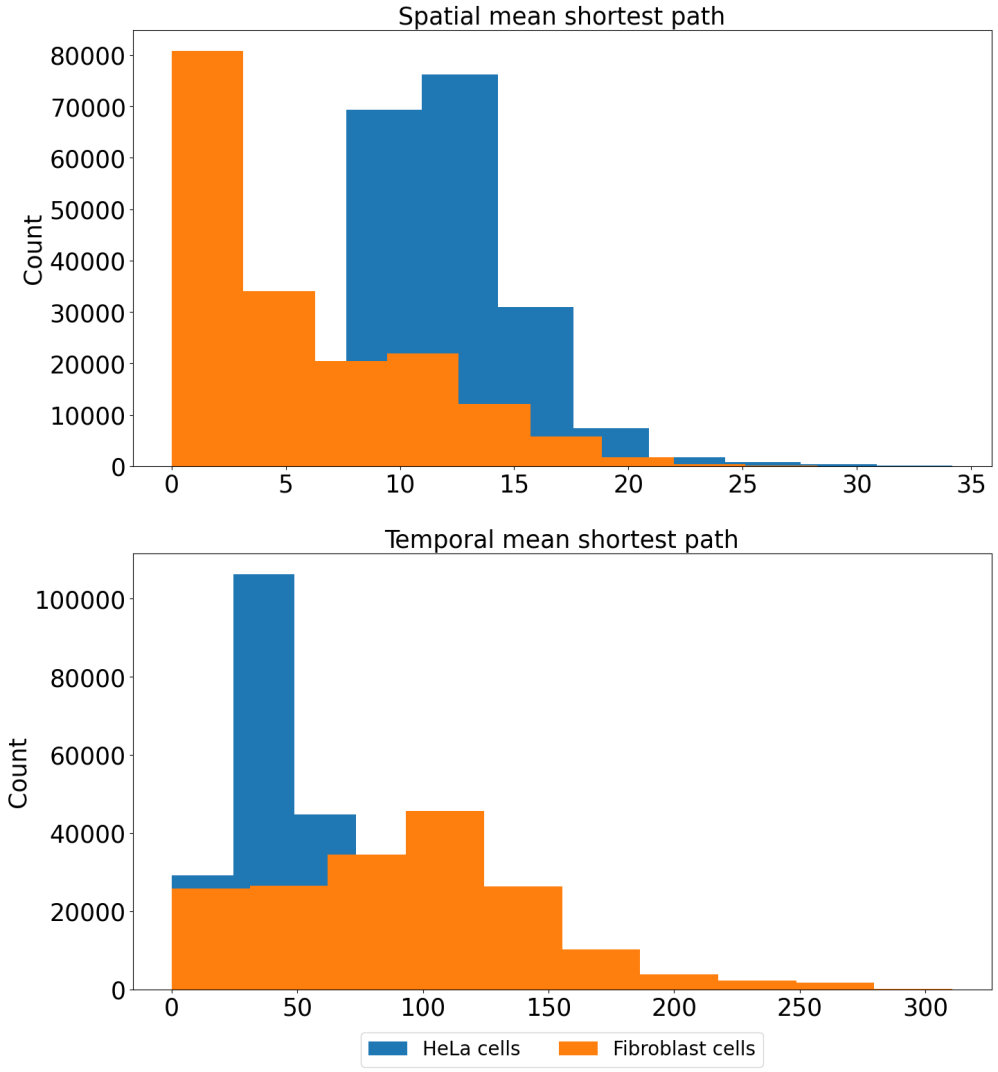


Figure 6.7: Histograms for spatial and temporal mean shortest path

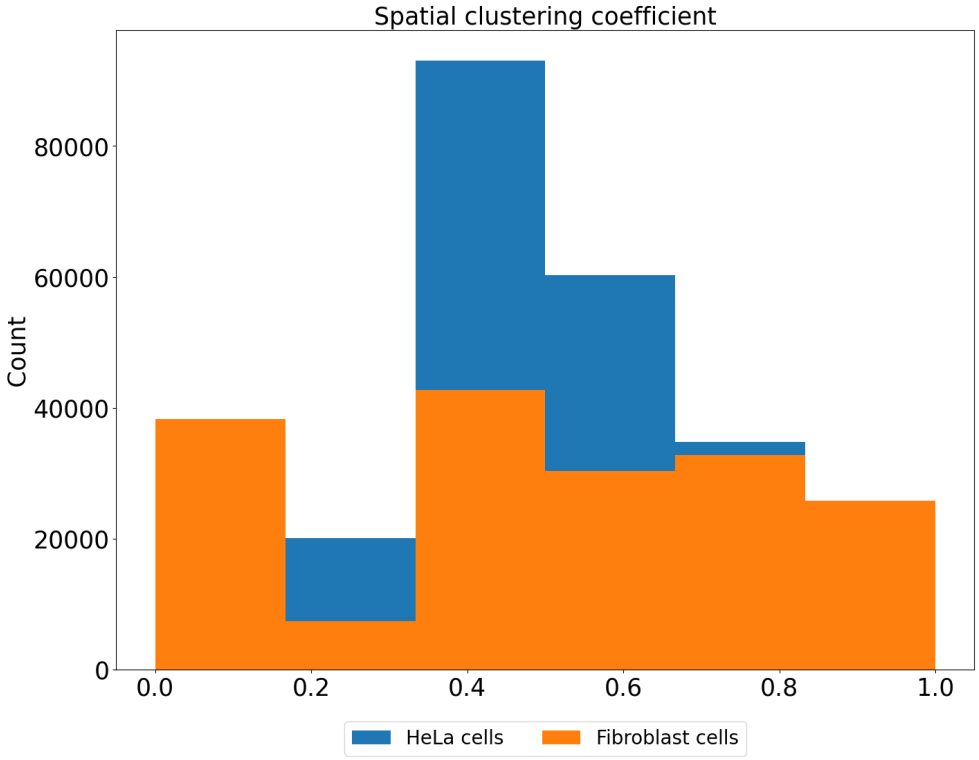


Figure 6.8: Histograms for spatial clustering coefficient

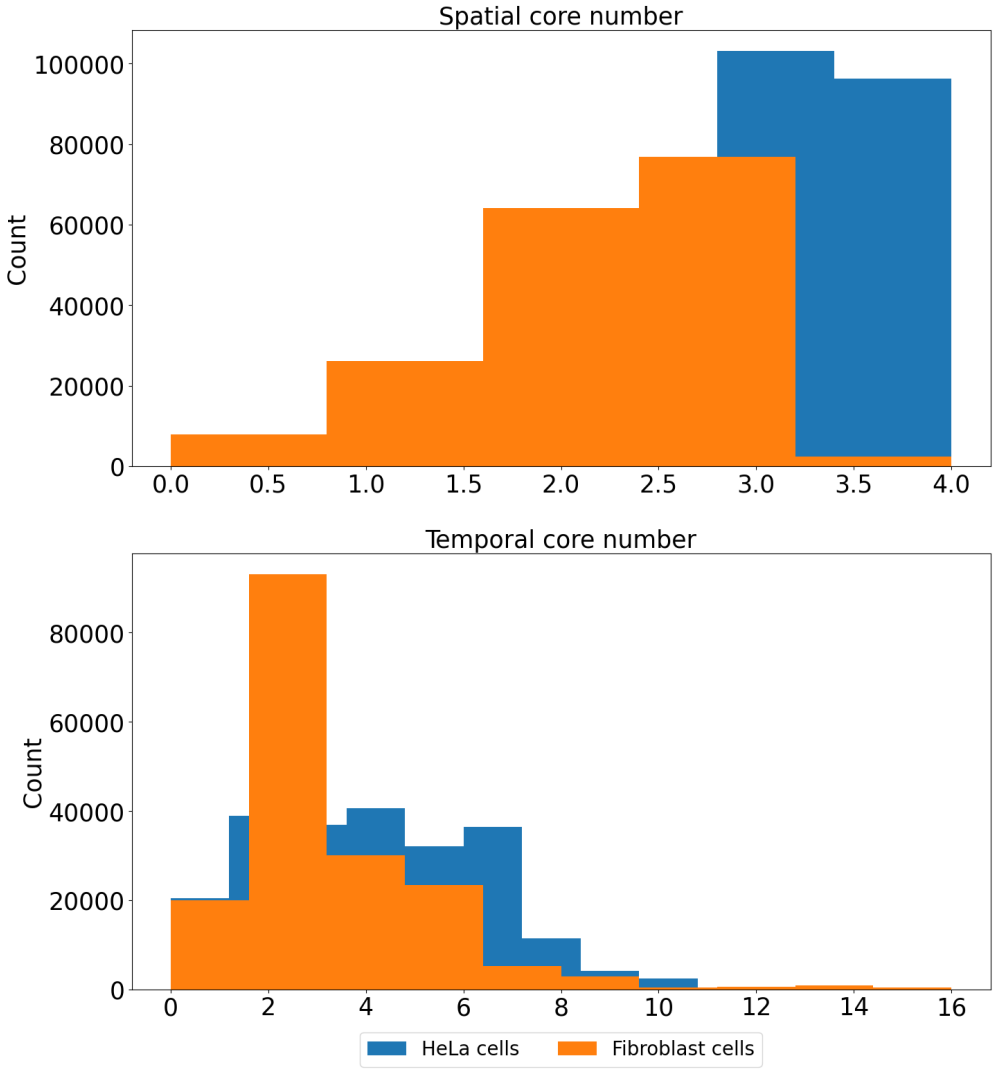


Figure 6.9: Histograms for spatial and temporal core number

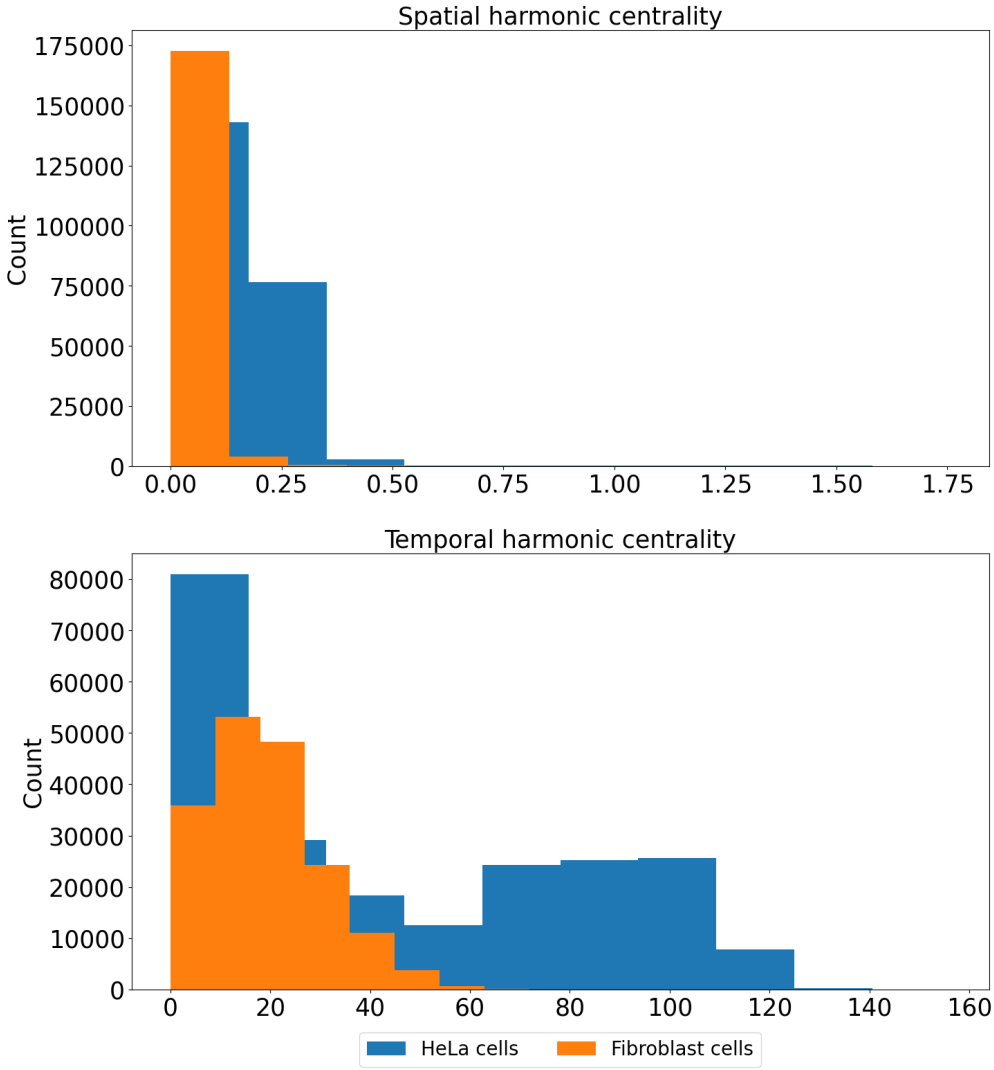


Figure 6.10: Histograms for spatial and temporal harmonic centrality

6.3.2 Results on GNN-based supervised classification

First, the graph structures employed to address the second prediction task in the experimental data are presented.

It is important to note that τ_{sp} refers to the threshold value set for spatial connections, and τ_{temp} corresponds to the threshold value for temporal connections between two consecutive frames. The following graph structures were considered: Delaunay, Gabriel and Distance with $\tau_{sp} \in \{100\,\mu\text{m}, 120\,\mu\text{m}, 130\,\mu\text{m}, 150\,\mu\text{m}, 160\,\mu\text{m}, 180\,\mu\text{m}\}$ and $\tau_{temp} \in \{50\,\mu\text{m}, 60\,\mu\text{m}, 80\,\mu\text{m}\}$.

Table 6.8: Comparison of model performance per node features and GNN layers for experimental data

	Node Features				
GNN layers	**position**	**position + spatial**	**position + temporal**	**position + spatial + temporal**	**position + spatio-temporal**
1	0.5110	0.4993	0.5221	0.5227	0.5036
2	0.4890	0.5077	0.5190	0.5167	0.5214
3	0.5143	0.5070	0.5178	0.5129	0.5141
4	0.5107	0.5028	0.5149	0.5182	0.5266
5	0.5101	0.4920	0.5148	0.5095	0.5135

Table 6.9: Comparison of model performance per node features for experimental data

	Node Features				
Model	**position**	**position + spatial**	**position + temporal**	**position + spatial + temporal**	**position + spatio-temporal**
GNN_{sp}	0.5110	0.5007	0.4969	0.5129	0.4995
GNN_{temp}	0.5037	0.5271	0.5113	0.5148	0.5186
GNN_{sp_temp}	0.5110	0.4990	0.5111	0.5110	0.4893

6.4 Spatial autocorrelation analysis

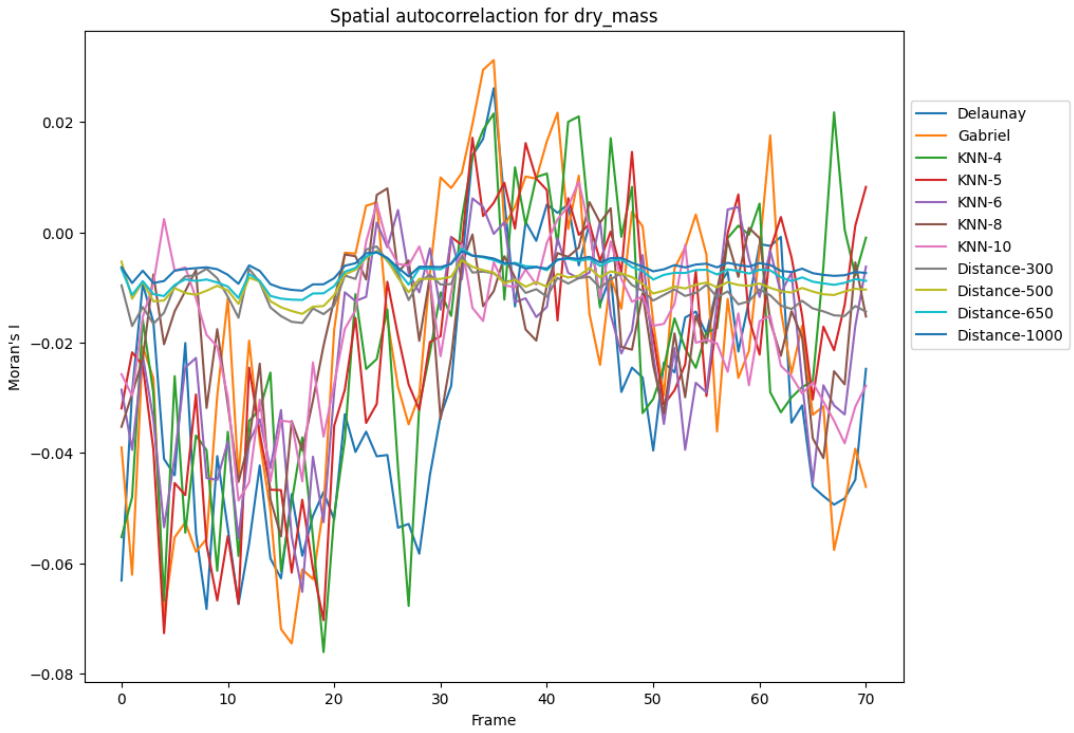


Figure 6.11: Moran's Index for the 2nd simulation of simulated data

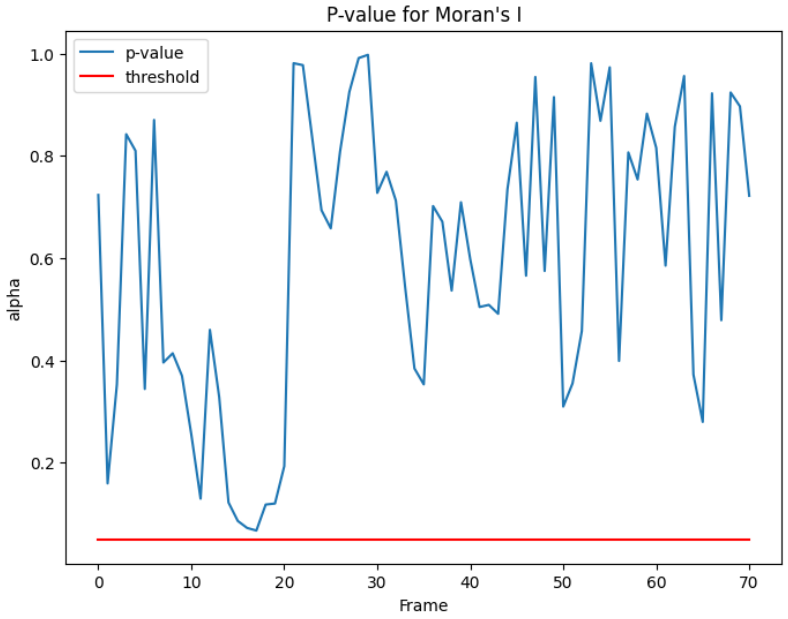


Figure 6.12: *P*-value from Moran's Index for the 2nd simulation of simulated data

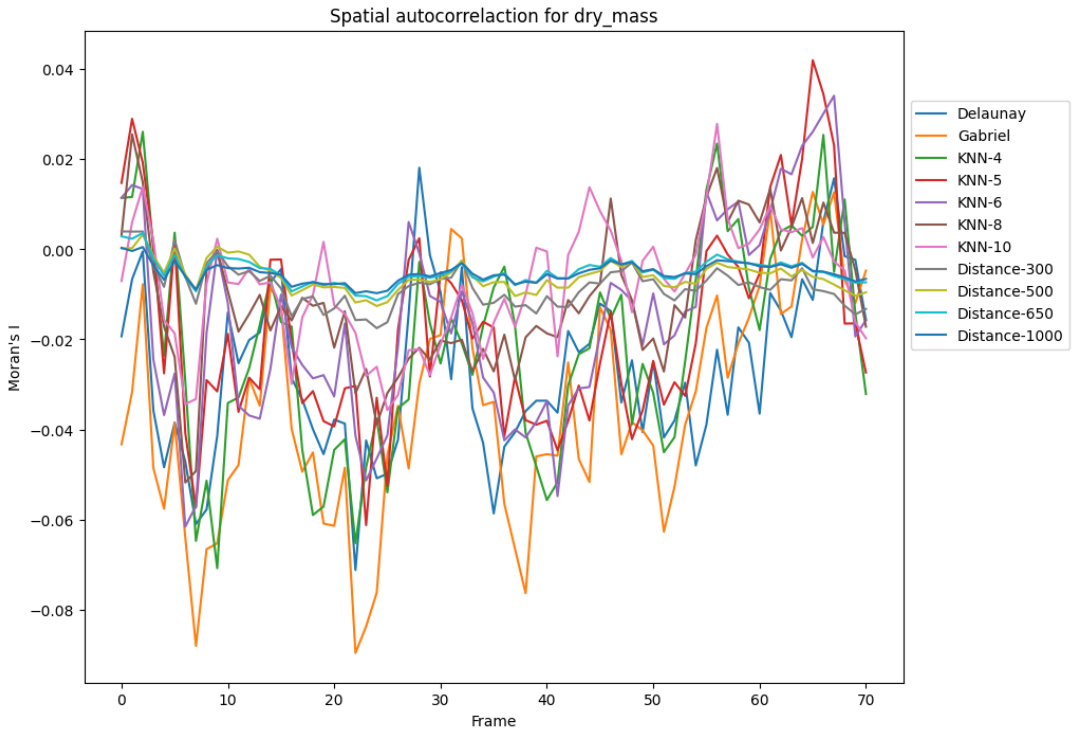


Figure 6.13: Moran's Index for the 3rd simulation of simulated data

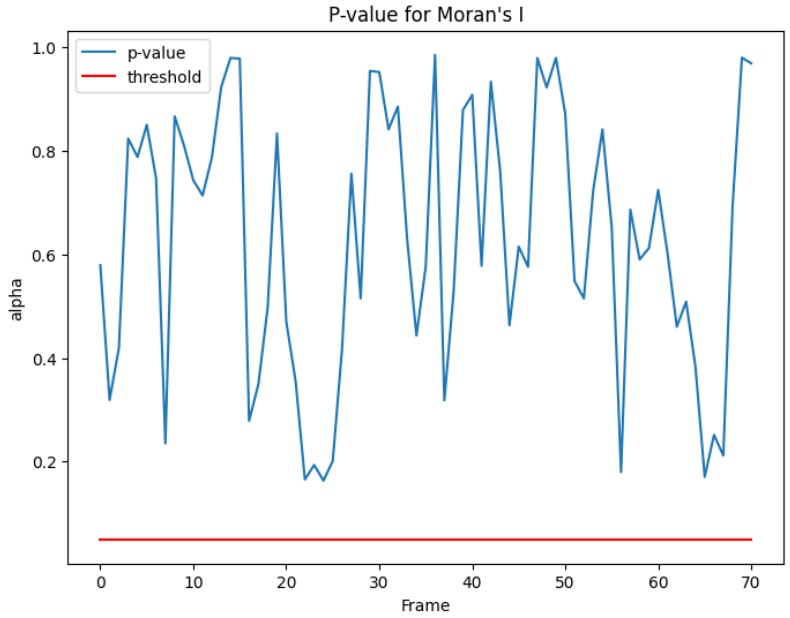


Figure 6.14: *P*-value from Moran's Index for the 3rd simulation of simulated data

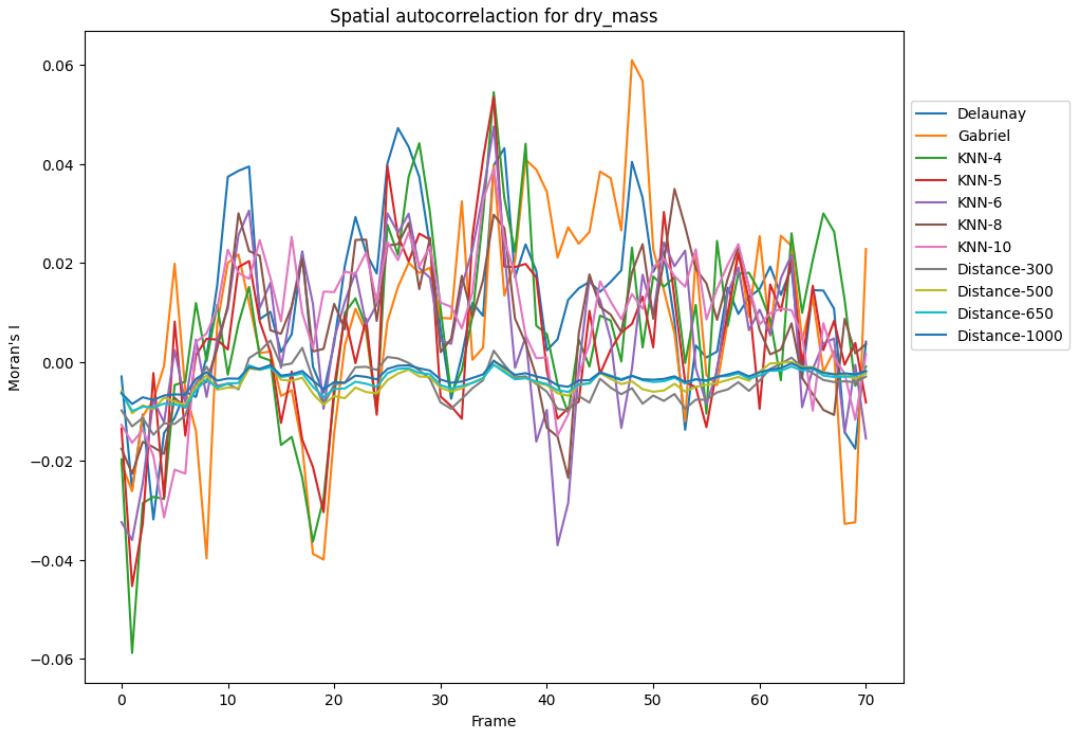


Figure 6.15: Moran's Index for the 4th simulation of simulated data

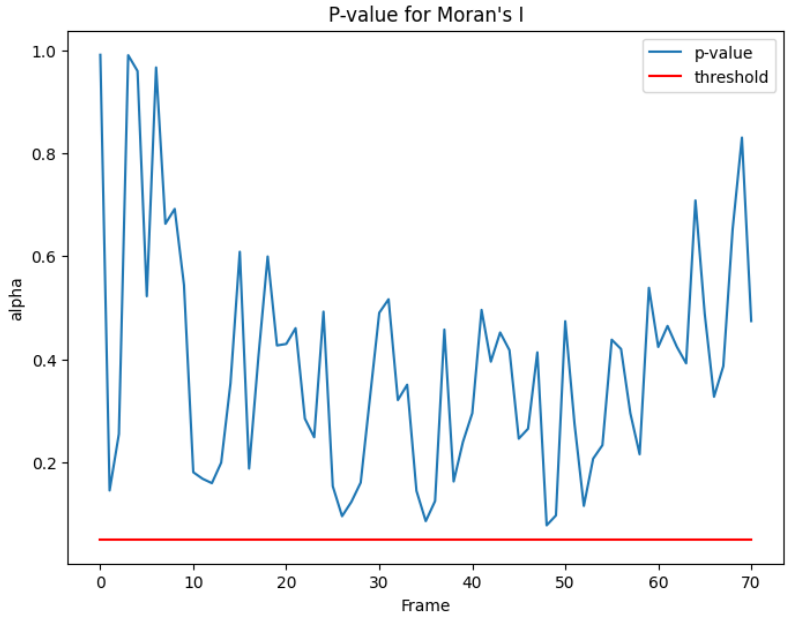


Figure 6.16: *P*-value from Moran's Index for the 4th simulation of simulated data

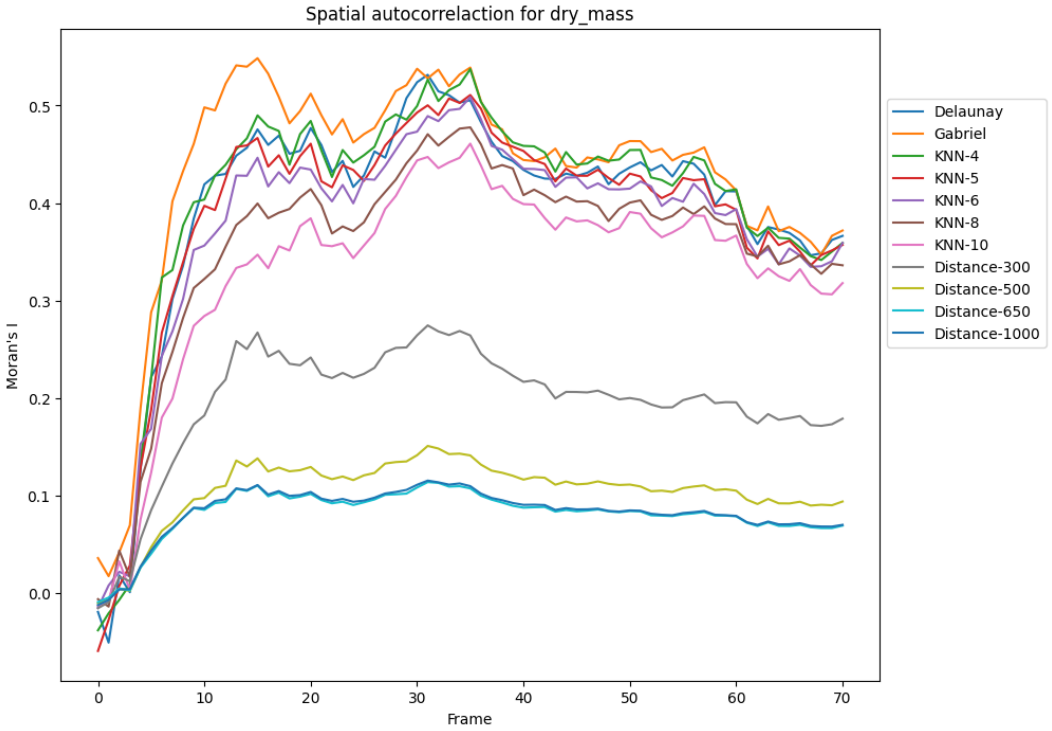


Figure 6.17: Moran's Index for the 5th simulation of simulated data

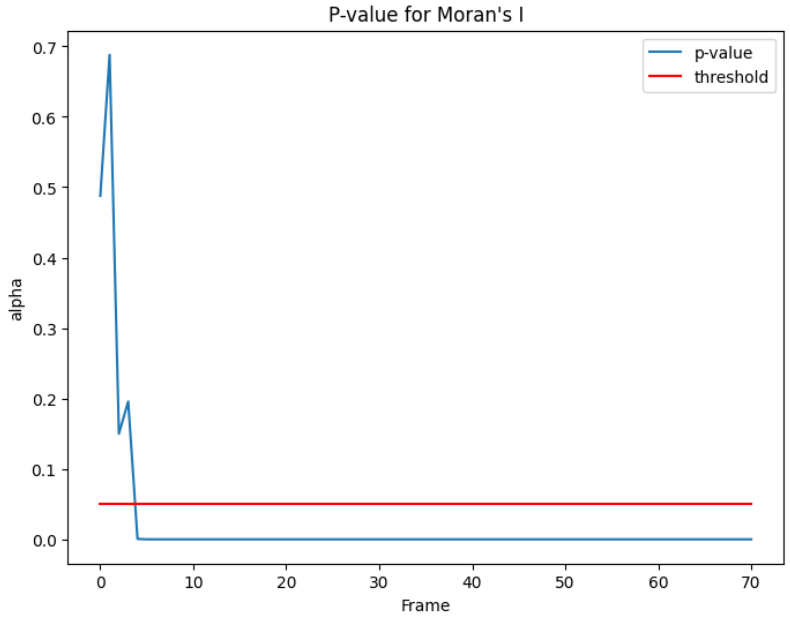


Figure 6.18: *P*-value from Moran's Index for the 5th simulation of simulated data

6.5 GNN architectures

6.5.1 Supervised classification model

6.5.1.1 GNN encoder

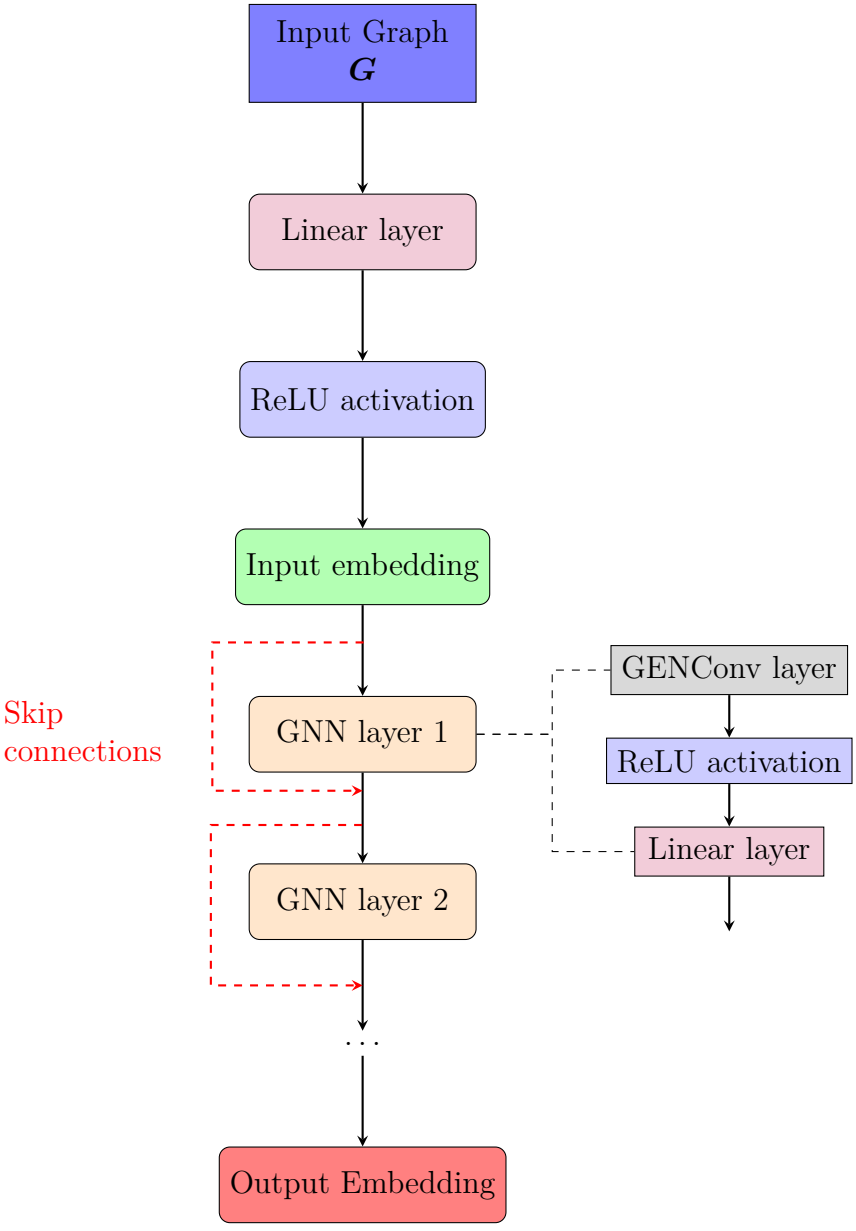


Figure 6.19: GNN encoder

6.5.1.2 MLP Classifier

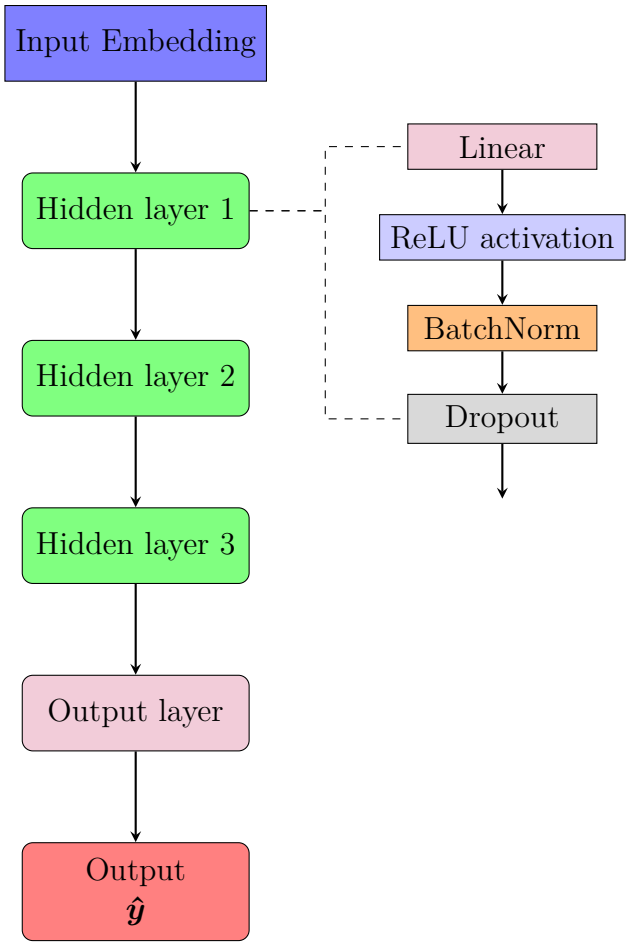


Figure 6.20: MLP Classifier

6.5.2 Dynamic prediction

6.5.2.1 GNN-based model

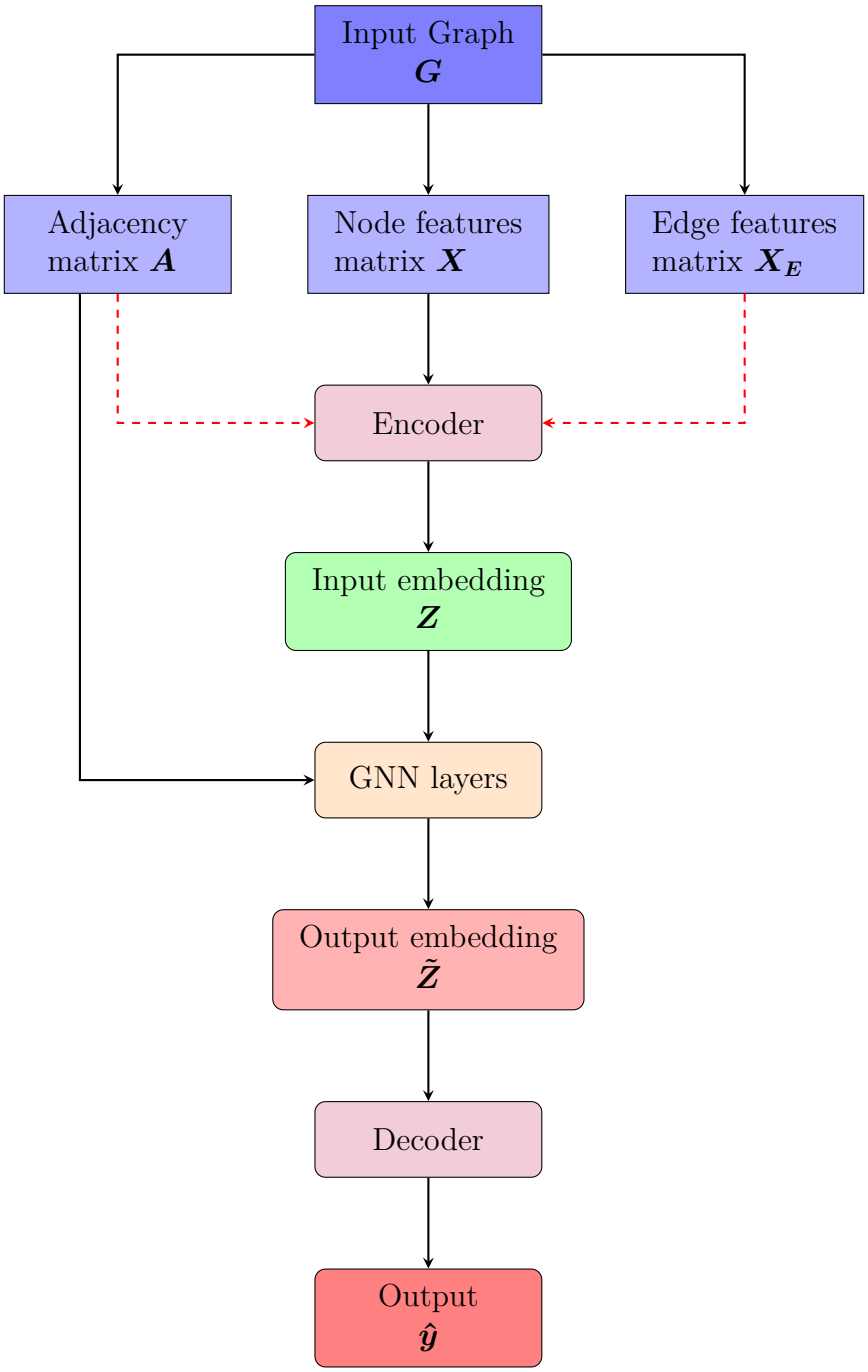


Figure 6.21: GNN-based model

References

[1] Mechanobiology Institute (MBI). *What is Actomyosin?* 2022. URL: https://www.mbi.nus.edu.sg/mbinfo/what-is-actomyosin/ (visited on 16/06/2025).

[2] Mechanobiology Institute (MBI). *What is cell polarity?* 2022. URL: https://www.mbi.nus.edu.sg/mbinfo/what-is-cell-polarity/ (visited on 16/06/2025).

[3] Sayan Acharya et al. "Cell Cycle State Prediction Using Graph Neural Networks". In: *2024 IEEE International Conference on Image Processing (ICIP)*. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 2916–2922.

[4] Priti Agarwal and Ronen Zaidel-Bar. "Principles of actomyosin regulation in vivo". In: *Trends Cell Biol.* 29.2 (Feb. 2019), pp. 150–163.

[5] R. Albert. "Scale-free networks in cell biology". In: *J. Cell Sci* 118 (2005), pp. 4947–4957.

[6] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks". In: *Rev. Mod. Phys.* 74.1 (Jan. 2002), pp. 47–97.

[7] B. Alberts et al. *Essential Cell Biology*. CRC Press, 2015. ISBN: 9781317806271. URL: https://books.google.fr/books?id=Cg4WAgAAQBAJ.

[8] Bruce Alberts et al. *Molecular Biology of the Cell*. 6th ed. Garland Science, 2017.

[9] Daniel B. Allan et al. *soft-matter/trackpy: v0.6.4*. 2024.

[10] C. Allier et al. "Imaging of dense cell cultures by multiwavelength lens-free video microscopy". In: *Cytometry A* 91.5 (May 2017), pp. 433–442.

[11] C. P. Allier et al. "Lensfree video microscopy: high throughput monitoring and cell tracking of 2D cell cultures". In: *Advanced Microscopy Techniques IV; and Neurophotonics II*. Optica Publishing Group, 2015, 95360J. DOI: 10.1364/ECBO.2015.95360J. URL: https://opg.optica.org/abstract.cfm?URI=ECBO-2015-95360J.

[12] Cedric Allier et al. "Lens-free video microscopy for the dynamic and quantitative analysis of adherent cell culture". In: *J. Vis. Exp.* 132 (Feb. 2018).

[13] Cédric Allier et al. "Label-free cell viability assay using lens-free microscopy". In: *Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues XVI*. Ed. by Daniel L Farkas, Dan V Nicolau, and Robert C Leif. San Francisco, United States: SPIE, Feb. 2018.

[14] Cédric Allier et al. "CNN-based cell analysis: From image to quantitative representation". In: *Front. Phys.* 9 (Jan. 2022).

[15] Laith Alzubaidi et al. “Robust and Efficient Approach to Diagnose Sickle Cell Anemia in Blood”. In: Jan. 2020, pp. 560–570. ISBN: 978-3-030-10435-1. DOI: 10.1007/978-3-030-16657-1_52.

[16] Mutsuki Amano, Masanori Nakayama, and Kozo Kaibuchi. “Rho-kinase/ROCK: A key regulator of the cytoskeleton and cell polarity”. In: *Cytoskeleton (Hoboken)* 67.9 (Sept. 2010), pp. 545–554.

[17] Erick Armingol et al. “Deciphering cell-cell interactions and communication from gene expression”. In: *Nat. Rev. Genet.* 22.2 (Feb. 2021), pp. 71–88.

[18] Sunil Arya et al. “An optimal algorithm for approximate nearest neighbor searching fixed dimensions”. In: *J. ACM* 45.6 (Nov. 1998), pp. 891–923.

[19] Christie Aschwanden. *Not Even Scientists Can Easily Explain P-values*. 2015. URL: https://web.archive.org/web/20190925221600/https://fivethirtyeight.com/features/not-even-scientists-can-easily-explain-p-values/ (visited on 08/06/2025).

[20] Armen S. Asratian, Tristan M. J. Denley, and Roland Haggkvist. *Bipartite Graphs and their Applications*. Vol. 131. Cambridge Tracts in Mathematics. Cambridge University Press, July 1998.

[21] Mihir Bafna, Hechen Li, and Xiuwei Zhang. “CLARIFY: cell–cell interaction and gene regulatory network refinement from spatially resolved transcriptomics”. In: *Bioinformatics* 39.39 Suppl 1 (June 2023), pp. i484–i493.

[22] Romain Bailly et al. “Detecting abnormal cell behaviors from dry mass time series”. In: *Sci. Rep.* 14.1 (Mar. 2024), p. 7053.

[23] A Barrat et al. “The architecture of complex weighted networks”. In: *Proc. Natl. Acad. Sci. U. S. A.* 101.11 (Mar. 2004), pp. 3747–3752.

[24] Nikolaos Bastas et al. “evolve2vec: Learning Network Representations Using Temporal Unfolding”. In: *MultiMedia Modeling*. Cham: Springer International Publishing, 2019, pp. 447–458. ISBN: 978-3-030-05710-7.

[25] V. Batagelj and M. Zaversnik. *An O(m) Algorithm for Cores Decomposition of Networks*. 2003. arXiv: cs/0310049 [cs.DS]. URL: https://arxiv.org/abs/cs/0310049.

[26] Peter W. Battaglia et al. *Interaction Networks for Learning about Objects, Relations and Physics*. 2016. arXiv: 1612.00222 [cs.AI].

[27] Richard Bellman. “On a routing problem”. In: *Quart. Appl. Math.* 16.1 (1958), pp. 87–90.

[28] Tal Ben-Haim and Tammy Riklin-Raviv. *Graph Neural Network for Cell Tracking in Microscopy Videos*. 2022. arXiv: 2202.04731 [cs.CV].

[29] Anthony Berdeu et al. “Lens-free microscopy for 3D + time acquisitions of 3D cell culture”. In: *Sci. Rep.* 8.1 (Oct. 2018), p. 16135.

[30] Serge Beucher. “Use of watersheds in contour detection”. In: *Proc. Int. Workshop on Image Processing, Sept. 1979*. 1979, pp. 17–21.

[31] Subhayan Bhattacharya et al. “Online social-network sensing models”. In: *Computational Intelligence Applications for Text and Sentiment Data Analysis*. Elsevier, 2023, pp. 113–140.

[32] Uttaran Bhattacharya et al. “STEP: Spatial Temporal Graph Convolutional Networks for Emotion Perception from Gaits”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.02 (Apr. 2020), pp. 1342–1350. DOI: 10.1609/aaai.v34i02.5490. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5490.

[33] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Ed. by M Jordan, J Kleinberg, and B Schölkopf. Vol. 4. Information science and statistics 4. Springer, 2006. Chap. Graphical, p. 738. ISBN: 9780387310732. DOI: 10.1117/1.2819119. arXiv: 0-387-31073-8. URL: http://www.library.wisc.edu/selectedtocs/bg0137.pdf.

[34] Avishek Biswas. *The History of Convolutional Neural Networks for Image Classification (1989- Today)*. 2024. URL: https://towardsdatascience.com/the-history-of-convolutional-neural-networks-for-image-classification-1989-today-5ea8a5c5fe20/ (visited on 16/06/2025).

[35] AlgoCademy Blog. *Critical Connections in a Network: Understanding Bridges in Graph Theory*. URL: https://algocademy.com/blog/critical-connections-in-a-network-understanding-bridges-in-graph-theory/ (visited on 16/06/2025).

[36] Vincent D. Blondel et al. “Fast unfolding of communities in large networks”. In: *J. Stat. Mech.* 2008.10 (Oct. 2008), P10008.

[37] S. E. M. Boas et al. “Computational Modeling of Angiogenesis: Towards a Multi-Scale Understanding of Cell–Cell and Cell–Matrix Interaction”. In: *In: Reinhart-King, C. (eds) Mechanical and Chemical Signaling in Angiogenesis, Berlin, Heidelberg: Springer* 12 (2013), pp. 161–183.

[38] Geoff Boeing. *Topological Graph Simplification Solutions to the Street Intersection Miscount Problem*. 2025. arXiv: 2407.00258 [physics.soc-ph]. URL: https://arxiv.org/abs/2407.00258.

[39] Paolo Boldi and Sebastiano Vigna. *Axioms for Centrality*. 2013. arXiv: 1308.2140 [cs.SI]. URL: https://arxiv.org/abs/1308.2140.

[40] Y. Y. Boykov and M-P Jolly. “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images”. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vancouver, BC, Canada: IEEE Comput. Soc, 2002.

[41] Ulrik Brande. “On variants of shortest-path betweenness centrality and their generic computation”. In: *Social Networks* 30.2 (2008), pp. 136–145. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2007.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0378873307000731.

[42] Ulrik Brandes. “A faster algorithm for betweenness centrality”. In: *The Journal of Mathematical Sociology* 25.2 (2001), pp. 163–177. DOI: 10.1080/0022250X.2001.9990249. URL: https://doi.org/10.1080/0022250X.2001.9990249.

[43] Ann-Kristin Brauns et al. “A defect in the peroxisomal biogenesis in germ cells induces a spermatogenic arrest at the round spermatid stage in mice”. In: *Sci. Rep.* 9.1 (July 2019), p. 9553.

[44] Maria Brbić et al. “Annotation of Spatially Resolved Single-cell Data with STELLAR”. In: *Nature Methods* 19 (2022), pp. 1411–1418. DOI: 10.1038/s41592-022-01651-8.

[45] Cláudia Brito and Sandra Sousa. “Non-muscle myosin 2A (NM2A): Structure, regulation and function”. In: *Cells* 9.7 (July 2020), p. 1590.

[46] Michael M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. 2021. arXiv: 2104.13478 [cs.LG].

[47] Joan Bruna et al. *Spectral Networks and Locally Connected Networks on Graphs*. 2014. arXiv: 1312.6203 [cs.LG].

[48] H Bunke and A Sanfeliu. *Syntactic And Structural Pattern Recognition - Theory And Applications*. Ed. by Horst Bunke and Alberto Sanfeliu. World Scientific Series In Computer Science. Singapore, Singapore: World Scientific Publishing, July 1989.

[49] Peter A. Burrough et al. *Principles of Geographical Information Systems*. Oxford University Pres, 2015. Chap. 8.11 Nearest neighbours: Thiessen (Dirichlet/Voroni) polygons, pp. 160–162.

[50] G. Carpentier et al. “Angiogenesis Analyzer for ImageJ–A comparative morphometric analysis of Endothelial Tube Formation Assay and Fibrin Bead Assay”. In: *Sci Rep* 10.1 (2020), pp. 1–13.

[51] G. W. Carter. “Inferring network interactions within a cell”. In: *Brief Bioinformatics* 6.4 (2005), pp. 1–13.

[52] F. Castellano et al. “Inducible recruitment of Cdc42 or WASP to a cell-surface receptor triggers actin polymerization and filopodium formation”. In: *Curr. Biol.* 9.7 (Apr. 1999), pp. 351–360.

[53] Ines Chami et al. “Machine Learning on Graphs: A Model and Comprehensive Taxonomy”. In: *Journal of Machine Learning Research* 23.89 (2022), pp. 1–64. URL: http://jmlr.org/papers/v23/20-852.html.

[54] Lucas von Chamier et al. “Democratising deep learning for microscopy with ZeroCostDL4Mic”. In: *Nature communications* 12.1 (2021), p. 2276.

[55] William Chan et al. *SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network*. 2021. arXiv: 2104.02133 [cs.CL]. URL: https://arxiv.org/abs/2104.02133.

[56] Douglas A Chapnick, Jeremy Jacobsen, and Xuedong Liu. “The development of a novel high throughput computational tool for studying individual and collective cellular migration”. In: *PLoS One* 8.12 (Dec. 2013), e82444.

[57] Cen Chen et al. “Gated Residual Recurrent Graph Neural Networks for Traffic Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 485–492. DOI: 10.1609/aaai.v33i01.3301485. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3821.

[58] Danian Chen et al. “Cell cycle duration determines oncogenic transformation capacity”. In: *Nature* 641.8065 (May 2025), pp. 1309–1318.

[59] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Springer International Publishing, 2018, pp. 833–851. ISBN: 978-3-030-01234-2.

[60] Hui-Jun Cheng et al. “A review for cell and particle tracking on microscopy images using algorithms and deep learning technologies”. In: *Biomedical Journal* 45.3 (2022), pp. 465–471. ISSN: 2319-4170. DOI: https://doi.org/10.1016/j.bj.2021.10.001. URL: https://www.sciencedirect.com/science/article/pii/S2319417021001359.

[61] Nicolas Chenouard, Isabelle Bloch, and Jean-Christophe Olivo-Marin. “Multiple Hypothesis Tracking for Cluttered Biological Image Sequences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.11 (2013), pp. 2736–3750. DOI: 10.1109/TPAMI.2013.97.

[62] Ananda S. Chowdhury et al. “Cell Tracking in Video Microscopy Using Bipartite Graph Matching”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2456–2459. DOI: 10.1109/ICPR.2010.601.

[63] Peng Chu et al. *TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking*. 2021. arXiv: 2104.00194 [cs.CV].

[64] Dmitriy M Chudakov, Sergey Lukyanov, and Konstantin A Lukyanov. “Fluorescent proteins as a toolkit for in vivo imaging”. In: *Trends Biotechnol.* 23.12 (Dec. 2005), pp. 605–613.

[65] Yu-An Chung et al. “w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training”. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2021, pp. 244–250. DOI: 10.1109/ASRU51503.2021.9688253.

[66] Geoffrey M Cooper. *The Cell: A Molecular Approach*. 2nd ed. Sinauer Associates 2000, 2000.

[67] Courtney Cox, James Doroghazi, and Douglas Mitchell. “The genomic landscape of ribosomal peptides containing thiazole and oxazole heterocycles”. In: *BMC Genomics* 16 (Oct. 2015). DOI: 10.1186/s12864-015-2008-0.

[68] Lauren Dalton and Robin Young. *Fundamentals of Cell Biology*. Corvallis, Oregon: Oregon State University, Jan. 2024.

[69] G. Dantzig, R. Fulkerson, and S Johnson. “Solution of a Large-Scale Traveling-Salesman Problem”. In: *Journal of the Operations Research Society of America* 2.4 (1954), pp. 393–410. URL: http://www.jstor.org/stable/166695 (visited on 14/05/2025).

[70] G. B. Dantzig and J. H. Ramser. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91. URL: http://www.jstor.org/stable/2627477 (visited on 14/05/2025).

[71] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. “HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2001–2011. DOI: 10.18653/v1/D18-1225. URL: https://aclanthology.org/D18-1225/.

[72] B. Delaunay. “Sur la sphère vide. A la mémoire de Georges Voronoï”. In: *Bulletin de l’Académie des Sciences de l’URSS. Classe des Sciences Mathématiques et Naturelle* 6 (1934), pp. 793–800.

[73] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[74] Reinhard Diestel. *Graph Theory*. 4th ed. Graduate Texts in Mathematics. Berlin, Germany: Springer, July 2010.

[75] E. W Dijkstra. "A note on two problems in connexion with graphs". In: *Numer. Math. (Heidelb.)* 1.1 (Dec. 1959), pp. 269–271.

[76] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2016), pp. 295–307. DOI: 10.1109/TPAMI.2015.2439281.

[77] Mike J. Downey et al. "Extracting fluorescent reporter time courses of cell lineages from high-throughput microscopy at low temporal resolution". In: *PLoS One* 6.12 (Dec. 2011), e27886.

[78] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. "Temporal Link Prediction Using Matrix and Tensor Factorizations". In: *ACM Transactions on Knowledge Discovery from Data* 5.2 (Feb. 2011), pp. 1–27. ISSN: 1556-472X. DOI: 10.1145/1921632.1921636. URL: http://dx.doi.org/10.1145/1921632.1921636.

[79] JH Eckert and R Erdmann. "Peroxisome biogenesis". In: *Reviews of physiology, biochemistry and pharmacology* (2003), pp. 75–121.

[80] Leo Egghe and Ronald Rousseau. *Introduction to informetrics*. London, England: Elsevier Science, May 1990.

[81] Isabella Ellinger and Adolf Ellinger. "Smallest unit of life: Cell biology". In: *Comparative Medicine*. Vienna: Springer Vienna, 2014, pp. 19–33.

[82] Luis M. Escudero et al. "Epithelial organisation revealed by a network of cellular contacts". In: *Nat. Commun.* 2.1 (Nov. 2011), p. 526.

[83] Esri. *Spatial Autocorrelation (Global Moran's I)*. URL: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/spatial-autocorrelation.htm (visited on 08/06/2025).

[84] Antonio Fabregat et al. "The Reactome Pathway Knowledgebase". In: *Nucleic Acids Res.* 46.D1 (Jan. 2018), pp. D649–D655.

[85] Pedro F. Felzenszwalb and Daniel P Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *Int. J. Comput. Vis.* 59.2 (Sept. 2004), pp. 167–181.

[86] Pedro Miguel Raminhos Ferreira. "Microscopy image segmentation by active contour models". PhD thesis. Universidade Nova de Lisboa, 2014.

[87] Celeste C. Finnerty et al. "Cytokine expression profile over time in severely burned pediatric patients". In: *Shock* 26.1 (2006), pp. 13–19.

[88] Sabine C. Fischer, George W. Bassel, and Philip Kollmannsberger. "Tissues as networks of cells: towards generative rules of complex organ development". In: *J. R. Soc. Interface* 20.204 (July 2023), p. 20230115.

[89] L. R. Ford. *Network Flow Theory*. Santa Monica, CA: RAND Corporation, 1956.

[90] Linton C. Freeman. "Centrality in social networks conceptual clarification". In: *Social Networks* 1.3 (1978), pp. 215–239. ISSN: 0378-8733. DOI: https://doi.org/10.1016/0378-8733(78)90021-7. URL: https://www.sciencedirect.com/science/article/pii/0378873378900217.

[91] Yohsuke T Fukai and Kyogo Kawaguchi. "LapTrack: linear assignment particle tracking with tunable metrics". In: *Bioinformatics* 39.1 (Dec. 2022), btac799. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btac799. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/1/btac799/48520757/btac799.pdf. URL: https://doi.org/10.1093/bioinformatics/btac799.

[92] K. Ruben Gabriel and Robert R. Sokal. "A New Statistical Approach to Geographic Variation Analysis". In: *Systematic Biology* 18.3 (1969), pp. 259–278.

[93] Wei Gao et al. "Study of biological networks using graph theory". In: *Saudi J. Biol. Sci.* 25.6 (Sept. 2018), pp. 1212–1219.

[94] Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. "Learning Sequence Encoders for Temporal Knowledge Graph Completion". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4816–4821. DOI: 10.18653/v1/D18-1516. URL: https://aclanthology.org/D18-1516/.

[95] Lamya Ghenim et al. "A new ultradian rhythm in mammalian cell dry mass observed by holography". In: *Sci. Rep.* 11.1 (Jan. 2021), p. 1290.

[96] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG].

[97] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *International Conference on Artificial Intelligence and Statistics*. 2010. URL: https://api.semanticscholar.org/CorpusID:5575601.

[98] Rishab Goel et al. *Diachronic Embedding for Temporal Knowledge Graph Completion*. 2019. arXiv: 1907.03143 [cs.LG]. URL: https://arxiv.org/abs/1907.03143.

[99] Simon Graham et al. "HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images". In: *Medical image analysis* 58 (2019), p. 101563.

[100] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. 2016. arXiv: 1607.00653 [cs.SI]. URL: https://arxiv.org/abs/1607.00653.

[101] Atika Gupta, Priya Matta, and Bhasker Pant. "Graph neural network: Current state of Art, challenges and applications". In: *Mater. Today* 46 (2021), pp. 10927–10932.

[102] A. Hagberg, D. Schult, and P. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX in Proceedings of the 7th Python in Science conference (SciPy 2008)". In: *G Varoquaux, T Vaught, J Millman (Eds.)* (2008), pp. 11–15.

[103] Katarzyna Hajdowska, Sebastian Student, and Damian Borys. "Graph based method for cell segmentation and detection in live-cell fluorescence microscope imaging". In: *Biomedical Signal Processing and Control* 71 (2022), pp. 1–13. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.103071. URL: https://www.sciencedirect.com/science/article/pii/S1746809421006686.

[104] Takuo Hamaguchi et al. "Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach". In: *arXiv preprint arXiv:1706.05674* (2017).

[105] Anne Hamik, Baiqiu Wang, and Mukesh K Jain. "Transcriptional regulators of angiogenesis". In: *Arteriosclerosis, thrombosis, and vascular biology* 26.9 (2006), pp. 1936–1947.

[106] Kerstin Hammernik et al. *Learning a Variational Network for Reconstruction of Accelerated MRI Data*. 2017. arXiv: 1704.00447 [cs.CV]. URL: https://arxiv.org/abs/1704.00447.

[107] E. Hannezo and B. D. Simons. "Multiscale dynamics of branching morphogenesis". In: *Current Opinion in Cell Biology* 60 (2019), pp. 99–105.

[108] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[109] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[110] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[111] Tao He et al. "Cell tracking using deep neural networks with multi-task learning". In: *Image and Vision Computing* 60 (2017), pp. 142–153.

[112] Yong He, Zhang J Chen, and Alan C Evans. "Small-world anatomical networks in the human brain revealed by cortical thickness from MRI". In: *Cerebral cortex* 17.10 (2007), pp. 2407–2419.

[113] National Institutes of Health. *Significant Research Advances Enabled by HeLa Cells*. URL: https://osp.od.nih.gov/hela-cells/significant-research-advances-enabled-by-hela-cells/ (visited on 16/06/2025).

[114] U.S. National Institutes of Health-National Cancer Institute. *Cell Structure*. URL: https://training.seer.cancer.gov/anatomy/cells_tissues_membranes/cells/structure.html (visited on 16/06/2025).

[115] Davide Heller et al. "EpiTools: An open-source image analysis toolkit for quantifying epithelial growth dynamics". In: *Dev. Cell* 36.1 (Jan. 2016), pp. 103–116.

[116] Mikael Henaff, Joan Bruna, and Yann LeCun. *Deep Convolutional Networks on Graph-Structured Data*. 2015. arXiv: 1506.05163 [cs.LG].

[117] L. Hervé et al. "2D and 3D Lensless Microscopes : Easy Tools for Visualizing Cell Culture Time-Lapses". In: *Imaging and Applied Optics Congress*. Optica Publishing Group, 2020, HW1C.2. DOI: 10.1364/DH.2020.HW1C.2. URL: https://opg.optica.org/abstract.cfm?URI=DH-2020-HW1C.2.

[118] L. Hervé et al. “Alternation of inverse problem approach and deep learning for lens-free microscopy image reconstruction”. In: *Sci. Rep.* 10 (Nov. 2020), p. 20207.

[119] Lionel Hervé et al. “Multispectral total-variation reconstruction applied to lens-free microscopy”. In: *Biomedical Optics Express* 9 (Oct. 2018), pp. 5828–5836. DOI: 10.1364/BOE.9.005828.

[120] Ryohei Hisano. “Semi-supervised Graph Embedding Approach to Dynamic Link Prediction”. In: *Complex Networks IX*. Cham: Springer International Publishing, 2018, pp. 109–121. ISBN: 978-3-319-73198-8. DOI: 10.1007/978-3-319-73198-8_10.

[121] Emir Hodzic. “Single-cell analysis: Advances and future perspectives”. In: *Bosn. J. Basic Med. Sci.* 16.4 (Nov. 2016), pp. 313–314.

[122] Petter Holme and Jari Saramäki. “Temporal networks”. In: *Physics reports* 519.3 (2012), pp. 97–125.

[123] R. Hooke. *Micrographia*. History of microscopy series. Science Heritage, 1665. ISBN: 9780940095076. URL: https://books.google.fr/books?id=LsbBada4VVYC.

[124] Ádám I Horváth et al. “Effect of allosteric inhibition of non-muscle myosin 2 on its intracellular diffusion”. In: *Sci. Rep.* 10.1 (Aug. 2020), p. 13341.

[125] Andrew Howard et al. “Searching for mobilenetv3”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1314–1324.

[126] Tao Hu et al. “CellTracker: an automated toolbox for single-cell segmentation and tracking of time-lapse microscopy images”. In: *Bioinformatics* 37.2 (2021), pp. 285–287.

[127] Zhongyu Huang et al. *Going Deeper into Permutation-Sensitive Graph Neural Networks*. 2022. arXiv: 2205.14368 [cs.LG]. URL: https://arxiv.org/abs/2205.14368.

[128] Geoffrey Hummelke. *Unveiling the Inner Lives of Cells: A Spatial Look at Cell-Cell Interaction*. URL: https://nanostring.com/blog/a-spatial-look-at-cell-cell-interaction/ (visited on 16/06/2025).

[129] Beckman Coulter Inc. *Epithelial Cells*. 2000. URL: https://www.beckman.fr/resources/cell-types/epithelial-cells (visited on 12/06/2025).

[130] Phase Holographic Imaging PHI Inc. *Single Cell Tracking Assay*. 2012. URL: https://phiab.com/applications/single-cell-tracking/ (visited on 10/06/2025).

[131] Shinya Inoue and Kenneth R Spring. *Video microscopy*. 2nd ed. New York, NY: Springer, Oct. 2012.

[132] National Human Genome Research Institute. *Fibroblast*. URL: https://www.genome.gov/genetics-glossary/Fibroblast (visited on 16/06/2025).

[133] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456.

[134] Marie-Paule Jacob. “Extracellular matrix and vascular ageing”. In: *Med. Sci. (Paris)* 22.3 (Mar. 2006), pp. 273–278.

[135] Míriam Javier-Torrent and Carlos A Saura. “Conventional and non-conventional roles of non-muscle myosin II-actin in neuronal development and degeneration”. In: *Cells* 9.9 (Aug. 2020), p. 1926.

[136] Ellen C Jensen. “Types of imaging, Part 2: an overview of fluorescence microscopy”. In: *Anat. Rec. (Hoboken)* 295.10 (Oct. 2012), pp. 1621–1627.

[137] Di Jin et al. *A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning*. 2021. arXiv: 2101.01669 [cs.SI]. URL: https://arxiv.org/abs/2101.01669.

[138] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (Aug. 2021), pp. 1–11. DOI: 10.1038/s41586-021-03819-2.

[139] Mikhail E Kandel et al. “Cell-to-cell influence on growth in large populations”. In: *Biomed. Opt. Express* 10.9 (Sept. 2019), pp. 4664–4675.

[140] Amol Kapoor et al. *Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks*. 2020. arXiv: 2007.03113 [cs.LG].

[141] Navneet Kaur, Sigrun Reumann, and Jianping Hu. “Peroxisome biogenesis and function”. In: *Arabidopsis Book* 7 (Sept. 2009), e0123.

[142] Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. *Learning Dynamic Graph Representation of Brain Connectome with Spatio-Temporal Attention*. 2021. arXiv: 2105.13495 [cs.CV].

[143] Dayoung Kim, Jonathan A. Cooper, and David M Helfman. “Loss of myosin light chain kinase induces the cellular senescence associated secretory phenotype to promote breast epithelial cell migration”. In: *Sci. Rep.* 14.1 (Oct. 2024), p. 25786.

[144] R.B. King and United States. Office of Naval Research. *Chemical Applications of Topology and Graph Theory: A Collection of Papers from a Symposium Held at the University of Georgia, Athens, Georgia, U.S.A., 18-22 April 1983*. Developments in Geotectonics. Elsevier, 1983. ISBN: 9780444416995. URL: https://books.google.fr/books?id=SBrwAAAAMAAJ.

[145] Thomas N. Kipf and Max Welling. *Variational Graph Auto-Encoders*. 2016. arXiv: 1611.07308 [stat.ML].

[146] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[147] Mikaela Koutrouli et al. “A Guide to Conquer the Biological Network Era Using Graph Theory”. In: *Front. Bioeng. Biotechnol.* 8 (Jan. 2020), p. 34.

[148] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.

[149] S. Krugmann et al. “Cdc42 induces filopodia by promoting the formation of an IRSp53:Mena complex”. In: *Curr. Biol.* 11.21 (Oct. 2001), pp. 1645–1655.

[150] Joseph B Kruskal. “On the shortest spanning subtree of a graph and the traveling salesman problem”. In: *Proc. Am. Math. Soc.* 7.1 (1956), pp. 48–50.

[151] Sonja Kühn et al. “The structure of FMNL2-Cdc42 yields insights into the mechanism of lamellipodia and filopodia formation”. In: *Nat. Commun.* 6.1 (May 2015), p. 7088.

[152] Srijan Kumar, Xikun Zhang, and Jure Leskovec. “Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1269–1278. ISBN: 9781450362016. DOI: 10.1145/3292500.3330895. URL: https://doi.org/10.1145/3292500.3330895.

[153] Vito Latora and Massimo Marchiori. “Efficient Behavior of Small-World Networks”. In: *Phys. Rev. Lett.* 87 (19 Oct. 2001), p. 198701. DOI: 10.1103/PhysRevLett.87.198701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.198701.

[154] E. L. Lawler. *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley-Interscience series in discrete mathematics and optimization. John Wiley & Sons, 1985. URL: https://books.google.fr/books?id=qbFlMwEACAAJ.

[155] Mengyuan Lee, Guanding Yu, and Huaiyu Dai. “Decentralized inference with graph neural networks in wireless communication systems”. In: *IEEE Transactions on Mobile Computing* 22.5 (2021), pp. 2582–2598.

[156] Mike Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL]. URL: https://arxiv.org/abs/1910.13461.

[157] Yi-Fan Li et al. “Time Series Forecasting Using a Unified SpatialTemporal Graph Convolutional Network”. In: *Proceedings of Preregister Workshop in 34th Conference on Neural Information Processing Systems* (2020).

[158] Guohao Li et al. *DeeperGCN: All You Need to Train Deeper GCNs*. 2022. URL: https://openreview.net/forum?id=qOcf6HgSmRH.

[159] Hongfei Li, Catherine A Calder, and Noel Cressie. “Beyond Moran’sI: Testing for spatial dependence based on the spatial autoregressive model”. In: *Geogr. Anal.* 39.4 (Oct. 2007), pp. 357–375.

[160] Tianpeng Li et al. “Exploring Temporal Community Structure via Network Embedding”. In: *IEEE Transactions on Cybernetics* 53.11 (2023), pp. 7021–7033. DOI: 10.1109/TCYB.2022.3168343.

[161] Yujia Li et al. *Gated Graph Sequence Neural Networks*. 2017. arXiv: 1511.05493 [cs.LG].

[162] Zhen Li et al. *Feedback Network for Image Super-Resolution*. 2019. arXiv: 1903.09814 [cs.CV]. URL: https://arxiv.org/abs/1903.09814.

[163] David Liben-Nowell and Jon Kleinberg. “The Link Prediction Problem for Social Networks”. In: *Journal of the American Society for Information Science and Technology* 58 (Nov. 2003), pp. 1019–1031. DOI: 10.1002/asi.20591.

[164] Jeff W. Lichtman and José-Angel Conchello. “Fluorescence microscopy”. In: *Nat. Methods* 2.12 (Dec. 2005), pp. 910–919.

[165] Congping Lin et al. “Structure and dynamics of ER: minimal networks and biophysical constraints”. In: *Biophysical Journal* 107.3 (2014), pp. 763–772.

[166] Meng Liu et al. *Deep Temporal Graph Clustering*. 2024. arXiv: 2305.10738 [cs.LG]. URL: https://arxiv.org/abs/2305.10738.

[167] Xili Liu, Jiawei Yan, and Marc W Kirschner. "Beyond G1/S regulation: how cell size homeostasis is tightly controlled throughout the cell cycle?" Feb. 2022.

[168] Yue Liu et al. "A Survey of Deep Graph Clustering: Taxonomy, Challenge, and Application". In: *ArXiv* abs/2211.12875 (2022). URL: https://api.semanticscholar.org/CorpusID:253801768.

[169] Yue Liu et al. "Deep graph clustering via dual correlation reduction". In: *Proc. Conf. AAAI Artif. Intell.* 36.7 (June 2022), pp. 7603–7611.

[170] Yue Liu et al. "Simple Contrastive Graph Clustering". In: *IEEE Transactions on Neural Networks and Learning Systems* 35.10 (2024), pp. 13789–13800. DOI: 10.1109/TNNLS.2023.3271871.

[171] Ze Liu et al. "Swin Transformer V2: Scaling Up Capacity and Resolution". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11999–12009. DOI: 10.1109/CVPR52688.2022.01170.

[172] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[173] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[174] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[175] Paul A. Longley et al. *Geographic Information Systems and Science*. Wiley, 2005. Chap. 14.4.4.1 Thiessen polygons, p. 333.

[176] Júlio Castro Lopes and Rui Pedro Lopes. "Computer Vision in Augmented, Virtual, Mixed and Extended Reality environments—A bibliometric review". In: *Visual Informatics* 8.4 (2024), pp. 13–22. ISSN: 2468-502X. DOI: https://doi.org/10.1016/j.visinf.2024.11.002. URL: https://www.sciencedirect.com/science/article/pii/S2468502X24000676.

[177] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2017.

[178] Xinghua Lou, Martin Schiegg, and Fred A Hamprecht. "Active structured learning for cell tracking: algorithm, framework, and usability". In: *IEEE transactions on medical imaging* 33.4 (2014), pp. 849–860.

[179] Lark Technologies Pte. Ltd. *Computer Vision In Self-Driving Cars*. URL: https://www.meegle.com/en_us/topics/computer-vision/computer-vision-in-self-driving-cars (visited on 16/06/2025).

[180] M. Lu et al. "ERnet: a tool for the semantic segmentation and quantitative analysis of endoplasmic reticulum topology". In: *Nature Methods* 20 (2023), pp. 569–579.

[181] Meng Lu et al. "The structure and global distribution of the endoplasmic reticulum network are actively regulated by lysosomes". In: *Sci. Adv.* 6.51 (Dec. 2020), eabc7209.

[182] Yuchao Lu et al. “Integrative microRNA-gene expression network analysis in genetic hypercalciuric stone-forming rat kidney”. In: *PeerJ* 4.e1884 (Mar. 2016), e1884.

[183] Yao Ma et al. *Streaming Graph Neural Networks*. 2018. arXiv: 1810.10627 [cs.LG]. URL: https://arxiv.org/abs/1810.10627.

[184] Yunpu Ma, Volker Tresp, and Erik Daxberger. *Embedding Models for Episodic Knowledge Graphs*. 2018. arXiv: 1807.00228 [cs.AI]. URL: https://arxiv.org/abs/1807.00228.

[185] J MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*. 1967.

[186] Sedigheh Mahdavi, Shima Khoshraftar, and Aijun An. “dynnode2vec: Scalable Dynamic Network Embedding”. In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 3762–3765. DOI: 10.1109/BigData.2018.8621910.

[187] Massimo Marchiori and Vito Latora. “Harmony in the small-world”. In: *Physica A: Statistical Mechanics and its Applications* 285.3-4 (2000), pp. 539–546.

[188] Alice Martin et al. “He2Cl: a 2-step clustering algorithm to characterize cellular heterogeneity from cell morpho-dynamics behaviors”. Oct. 2024.

[189] Elizabeth Martin and Robert Hine. *A dictionary of biology*. Ed. by Elizabeth A Martin and Robert S Hine. 6th ed. Oxford Quick Reference. London, England: Oxford University Press, Feb. 2008.

[190] Geoffrey J. McLachlan and David Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Nashville, TN: John Wiley & Sons, Sept. 2000.

[191] C. E. Miller, A. W. Tucker, and R. A Zemlin. “Integer programming formulation of traveling salesman problems”. In: *J. ACM* 7.4 (Oct. 1960), pp. 326–329.

[192] P. A. P Moran. “Notes on continuous stochastic phenomena”. In: *Biometrika* 37.1/2 (June 1950), p. 17.

[193] Nairouz Mrabah et al. *Rethinking Graph Auto-Encoder Models for Attributed Graph Clustering*. 2021. arXiv: 2107.08562 [cs.LG]. URL: https://arxiv.org/abs/2107.08562.

[194] Ryan Murphy et al. “Relational pooling for graph representations”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4663–4673.

[195] M. E. J. Newman. “The structure and function of complex networks”. In: *SIAM Rev* 45.2 (2003), pp. 167–256.

[196] Mark Newman. *Networks: An Introduction*. London, England: Oxford University Press, Mar. 2010.

[197] Giang H. Nguyen et al. “Dynamic Network Embeddings: From Random Walks to Temporal Random Walks”. In: *2018 IEEE International Conference on Big Data (Big Data)*. Dec. 2018, pp. 1085–1092. DOI: 10.1109/BigData.2018.8622109.

[198] Giang Hoang Nguyen et al. “Continuous-Time Dynamic Network Embeddings”. In: *Companion Proceedings of the The Web Conference 2018*. WWW ’18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 969–976. ISBN: 9781450356404. DOI: 10.1145/3184558.3191526. URL: https://doi.org/10.1145/3184558.3191526.

[199] Thang L. Nguyen et al. “Quantitative phase imaging: Recent advances and expanding potential in biomedicine”. In: *ACS Nano* 16.8 (Aug. 2022), pp. 11516–11544.

[200] Frank Nielsen. “Hierarchical Clustering”. In: Feb. 2016, pp. 195–211. ISBN: 978-3-319-21902-8. DOI: 10.1007/978-3-319-21903-5_8.

[201] Lutz Oettershagen and Petra Mutzel. “Computing top-k temporal closeness in temporal networks”. In: *Knowl. Inf. Syst.* 64.2 (Feb. 2022), pp. 507–535.

[202] Eric Olson. *Particle Shape Factors and Their Use in Image Analysis – Part 1: Theory*. 2011. URL: https://particletechlabs.com/ptl-press/particle-shape-factors-and-their-use-in-image-analysis-part-1-theory/ (visited on 08/06/2025).

[203] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[204] Nobuyuki Otsu. “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[205] Ryohei Oya et al. “Influence of myosin regulatory light chain and myosin light chain kinase on the physiological function of inner ear hair cells”. In: *J. Assoc. Res. Otolaryngol.* 26.3 (June 2025), pp. 225–238.

[206] D. Paez and Y Filion. “Use of network theory and reliability indexes for the validation of synthetic water distribution systems case studies”. In: 24 (Dec. 2017), pp. 2–7.

[207] Charlotte Pain et al. “Quantitative analysis of plant ER architecture and dynamics”. In: *Nat. Commun.* 10.1 (Feb. 2019), p. 984.

[208] Jiahao Pang and Jin Zeng. *Graph spectral image restoration*. en. Aug. 2021.

[209] Ho-Jin Park et al. “Human umbilical vein endothelial cells and human dermal microvascular endothelial cells offer new insights into the relationship between lipid metabolism and angiogenesis”. In: *Stem Cell Rev.* 2.2 (2006), pp. 93–102.

[210] Jiwoong Park et al. *Symmetric Graph Convolutional Autoencoder for Unsupervised Graph Representation Learning*. 2019. arXiv: 1908.02441 [cs.LG]. URL: https://arxiv.org/abs/1908.02441.

[211] Namyong Park et al. “CGC: Contrastive Graph Clustering for Community Detection and Tracking”. In: *Proceedings of the ACM Web Conference 2022*. WWW’22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 1115–1126. ISBN: 9781450390965. DOI: 10.1145/3485447.3512160. URL: https://doi.org/10.1145/3485447.3512160.

[212] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '14. ACM, Aug. 2014, pp. 701–710. DOI: 10.1145/2623330.2623732. URL: http://dx.doi.org/10.1145/2623330.2623732.

[213] Jesús Pineda et al. "Geometric deep learning reveals the spatiotemporal features of microscopic motion". In: *Nature Machine Intelligence* 5 (2023), pp. 71–82. DOI: 10.1038/s42256-022-00595-0.

[214] John Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Adv. Large Margin Classif.* 10 (June 2000).

[215] Laras Pratiwi, Elisa Elisa, and Henry Sutanto. "Probing the protrusions: lamellipodia and filopodia in cancer invasion and beyond". In: *Mechanobiol. Med.* 2.2 (June 2024), p. 100064.

[216] R. C. Prim. "Shortest connection networks and some generalizations". In: *The Bell System Technical Journal* 36.6 (1957), pp. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x.

[217] Ante Pusic. *Graph Clustering Algorithms: Usage and Comparison*. 2023. URL: memgraph.com/blog/graph-clustering-algorithms-usage-comparison (visited on 08/06/2025).

[218] Xuhui Qin et al. "Identification and validation of a PEX5-dependent signature for prognostic prediction in glioma". In: *Biomolecules* 14.3 (Mar. 2024), p. 314.

[219] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG]. URL: https://arxiv.org/abs/1910.10683.

[220] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical Review E* 76.3 (Sept. 2007). ISSN: 1550–2376. DOI: 10.1103/physreve.76.036106. URL: http://dx.doi.org/10.1103/PhysRevE.76.036106.

[221] Beatriz Remeseiro and Veronica Bolon-Canedo. "A review of feature selection methods in medical applications". In: *Computers in Biology and Medicine* 112 (2019), p. 103375. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2019.103375. URL: https://www.sciencedirect.com/science/article/pii/S0010482519302525.

[222] Markus Rempfler et al. "Tracing cell lineages in videos of lens-free microscopy". In: *Med. Image Anal.* 48 (Aug. 2018), pp. 147–161.

[223] Elisabeth G. Rens and Roeland M.H. Merks. "Cell Shape and Durotaxis Explained from Cell-Extracellular Matrix Forces and Focal Adhesion Dynamics". In: *iScience* 23.9 (2020), p. 101488. ISSN: 2589-0042. DOI: https://doi.org/10.1016/j.isci.2020.101488. URL: https://www.sciencedirect.com/science/article/pii/S2589004220306805.

[224] Anne J. Ridley et al. "Cell migration: integrating signals from front to back". In: *Science* 302.5651 (2003), pp. 1704–1709.

[225] Florian Robert et al. “Enhancing cell instance segmentation in scanning electron microscopy images via a deep contour closing operator”. In: *Computers in Biology and Medicine* 190 (2025), p. 109972. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2025.109972. URL: https://www.sciencedirect.com/science/article/pii/S0010482525003233.

[226] Julia Jean Robinson. “On the Hamiltonian Game (A Traveling Salesman Problem)”. In: 1949. URL: https://api.semanticscholar.org/CorpusID:122464330.

[227] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[228] Frank Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychol. Rev.* 65.6 (Nov. 1958), pp. 386–408.

[229] Emanuele Rossi et al. *Temporal Graph Networks for Deep Learning on Dynamic Graphs*. 2020. arXiv: 2006.10637 [cs.LG]. URL: https://arxiv.org/abs/2006.10637.

[230] Neil Salkind. *Encyclopedia of measurement and statistics*. 2455 Teller Road, Thousand Oaks California 91320 United States of America, 2007.

[231] Jose M Sallan. *Efficiency of Unweighted Networks*. 2023. URL: https://jmsallan.netlify.app/blog/efficiency-of-unweighted-networks/ (visited on 16/06/2025).

[232] Alvaro Sanchez-Gonzalez et al. *Learning to Simulate Complex Physics with Graph Networks*. 2020. arXiv: 2002.09405 [cs.LG].

[233] A. Sanfeliu et al. “Graph-based representations and techniques for image processing and image analysis”. In: *Pattern Recognition* 35 (Mar. 2002), pp. 639–650. DOI: 10.1016/S0031-3203(01)00066-8.

[234] Kavish Sanghvi. *Image Classification Techniques*. 2020. URL: https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac (visited on 16/06/2025).

[235] Aravind Sankar et al. “DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks”. In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. WSDM ’20. Houston, TX, USA: Association for Computing Machinery, 2020, pp. 519–527. ISBN: 9781450368223. DOI: 10.1145/3336191.3371845. URL: https://doi.org/10.1145/3336191.3371845.

[236] Gaetano Santulli, ed. *Angiogenesis*. Cell biology research progress. Hauppauge, NY: Nova Science, July 2013.

[237] Y. Sasai. “Cytosystems dynamics in self-organization of tissue architecture”. In: *Nature* 493.7432 (2013), pp. 318–326.

[238] Franco Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[239] Francoise Schoentgen. “Insight on the role of RKIP in cancer through key protein partners and cellular protrusions”. In: *Prognostic and Therapeutic Applications of RKIP in Cancer*. Elsevier, 2020, pp. 3–35.

[240] Alexander Schrijver. “On the History of Combinatorial Optimization (Till 1960)”. In: *Discrete Optimization*. Ed. by K. Aardal, G. L. Nemhauser, and R. Weismantel. Vol. 12. Handbooks in Operations Research and Management Science. Elsevier, 2005, pp. 1–68. DOI: https://doi.org/10.1016/S0927-0507(05)12001-5. URL: https://www.sciencedirect.com/science/article/pii/S0927050705120015.

[241] QIMA Life Sciences. *Normal Human Dermal Fibroblasts (NHDF) in in vitro Research*. URL: https://qima-lifesciences.com/bioanalysis-bioengineering/cell-and-tissue-engineering/normal-human-dermal-fibroblasts/ (visited on 16/06/2025).

[242] Arindrajit Seal, Arunava Das, and Prasad Sen. “Watershed: an image segmentation approach”. In: *International Journal of Computer Science and Information Technologies* 6.3 (2015), pp. 2295–2297.

[243] Zekai Sen. *Spatial Modeling Principles in Earth Sciences*. Springer, 2016. Chap. 2.8.1 Delaney, Varoni, and Thiessen Polygons, pp. 57–59.

[244] Francesc Serratosa and Alberto Sanfeliu. “Function-described graphs applied to 3D object representation”. In: *Image Analysis and Processing*. Ed. by Alberto Del Bimbo. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 701–708. ISBN: 978-3-540-69585-1.

[245] Jean de Seze et al. “Optogenetic control of a GEF of RhoA uncovers a signaling switch from retraction to protrusion”. In: *Elife* 12 (May 2025).

[246] Mirvat El-Sibai et al. “Cdc42 is required for EGF-stimulated protrusion and motility in MTLn3 carcinoma cells”. In: *J. Cell Sci.* 120.Pt 19 (Oct. 2007), pp. 3465–3474.

[247] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

[248] Abhijeet R. Sonawane et al. “Network medicine in the age of biomedical big data”. In: *Front. Genet.* 10 (Apr. 2019), p. 294.

[249] L. Andrew Staehelin and Harvey F. Lodish. *Cell*. URL: https://www.britannica.com/science/cell-biology (visited on 16/06/2025).

[250] “Stiffness — an unknown world of mechanical science?” In: *Injury* 31 (May 2000), pp. 14–84.

[251] Carsen Stringer et al. “Cellpose: a generalist algorithm for cellular segmentation”. In: *Nature methods* 18.1 (2021), pp. 100–106.

[252] S. H. Strogatz. “Exploring complex networks”. In: *Nature* 410 (2001), pp. 268–276.

[253] Nikolaus Sucher. *Text For Biology at Roxbury Community College: The Cell*. URL: https://nikolaussucher.github.io/biology-text/the-cell.html (visited on 16/06/2025).

[254] Christian Szegedy et al. “Inception-v4, Inception-ResNet and the impact of residual connections on learning”. In: *Proc. Conf. AAAI Artif. Intell.* 31.1 (Feb. 2017).

[255] Wenbing Tao, Hai Jin, and Yimin Zhang. “Color Image Segmentation Based on Mean Shift and Normalized Cuts”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.5 (2007), pp. 1382–1389. DOI: 10.1109/TSMCB.2007.902249.

[256] Santosh Thapa et al. “Al-induced proteomics changes in tomato plants over-expressing a glyoxalase I gene”. In: (Apr. 2020).

[257] Helen Theissen et al. “Multi-Scale Graphical Representation of Cell Environment”. In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 2022, pp. 3522–3525. DOI: 10.1109/EMBC48229.2022.9871710.

[258] Jean-Yves Tinevez et al. “TrackMate: An open and extensible platform for single-particle tracking”. In: *Methods* 115 (2017), pp. 80–90.

[259] V. A. Traag, L. Waltman, and N. J van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Sci. Rep.* 9.1 (Mar. 2019), p. 5233.

[260] Lauren E Tracy, Raquel A Minasian, and E J Caterson. “Extracellular matrix and dermal fibroblast function in the healing wound”. In: *Adv. Wound Care (New Rochelle)* 5.3 (Mar. 2016), pp. 119–136.

[261] Rakshit Trivedi et al. *Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs*. 2017. arXiv: 1705.05742 [cs.AI]. URL: https://arxiv.org/abs/1705.05742.

[262] Hsieh-Fu Tsai et al. “Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning”. In: *SoftwareX* 9 (Jan. 2019), pp. 230–237.

[263] Wen-Hsiang Tsai and King-Sun Fu. “Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.12 (1979), pp. 757–768. DOI: 10.1109/TSMC.1979.4310127.

[264] R. Y Tsien. “The green fluorescent protein”. In: *Annu. Rev. Biochem.* 67.1 (1998), pp. 509–544.

[265] Amit Tzur et al. “Cell growth and size homeostasis in proliferating animal cells”. In: *Science* 325.5937 (July 2009), pp. 167–171.

[266] R. B. Urquhart. “Algorithms for computation of relative neighborhood graph”. In: *Electronics Letters* 16.14 (1980), pp. 556–557.

[267] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[268] Julia Vaynberg and Jun Qin. “Chapter Six - Characterization of pseudokinase ILK-mediated actin assembly”. In: *Pseudokinases*. Ed. by Natalia Jura and James M. Murphy. Vol. 667. Methods in Enzymology. Academic Press, 2022, pp. 123–146. DOI: https://doi.org/10.1016/bs.mie.2022.03.027. URL: https://www.sciencedirect.com/science/article/pii/S0076687922001185.

[269] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML].

[270] J. Vergés et al. "Face recognition: graph matching versus neural techniques". In: *Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*. May 1999, p. 259. URL: http://www.iri.upc.edu/english/fitxa.php?id=292.

[271] Matheus P Viana et al. "The simplicity of planar networks". In: *Sci. Rep.* 3.1 (Dec. 2013), p. 3495.

[272] Miguel Vicente-Manzanares et al. "Non-muscle myosin II takes centre stage in cell adhesion and migration". In: *Nat. Rev. Mol. Cell Biol.* 10.11 (Nov. 2009), pp. 778–790.

[273] Chun Wang et al. "MGAE: Marginalized Graph Autoencoder for Graph Clustering". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. Singapore Singapore: ACM, Nov. 2017.

[274] Juexin Wang et al. "scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses". In: *Nat. Commun.* 12.1 (Mar. 2021), p. 1882.

[275] Lijing Wang et al. "CausalGNN: Causal-Based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.11 (June 2022), pp. 12191–12199. DOI: 10.1609/aaai.v36i11.21479. URL: https://ojs.aaai.org/index.php/AAAI/article/view/21479.

[276] Shoujin Wang et al. *Graph Learning Approaches to Recommender Systems: A Review*. 2020. arXiv: 2004.11718 [cs.IR].

[277] Yanan Wang et al. "Cell graph neural networks enable the precise prediction of patient survival in gastric cancer". In: *NPJ Precis. Oncol.* 6.45 (June 2022).

[278] Ronald L Wasserstein and Nicole A Lazar. "The ASA statement on p-values: Context, process, and purpose". In: *Am. Stat.* 70.2 (Apr. 2016), pp. 129–133.

[279] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), pp. 440–442. DOI: 10.1038/30918.

[280] Shui-gen Wei et al. "Motion Detection Based on Optical Flow and Self-adaptive Threshold Segmentation". In: *Procedia Engineering* 15 (2011). CEIS 2011, pp. 3471–3476. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2011.08.650. URL: https://www.sciencedirect.com/science/article/pii/S1877705811021515.

[281] Rebecca A. Worthylake and Keith Burridge. "RhoA and ROCK promote migration by limiting membrane protrusions". In: *Journal of Biological Chemistry* 278.15 (2003), pp. 13578–13584.

[282] Yongji Wu et al. "Graph convolutional networks with markov random field reasoning for social spammer detection". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 01. 2020, pp. 1054–1061.

[283] Z. Wu and R. Leahy. "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.11 (1993), pp. 1101–1113. DOI: 10.1109/34.244673.

[284] Zhenqin Wu et al. “Graph deep learning for the characterization of tumour microenvironments from spatial protein profiles in tissue specimens”. In: *Nature Biomedical Engineering* 6 (Nov. 2022), pp. 1–14. DOI: 10.1038/s41551-022-00951-w.

[285] Zonghan Wu et al. *Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks*. 2020. arXiv: 2005.11650 [cs.LG].

[286] Chenjin Xu et al. “Temporal Knowledge Graph Completion Based on Time Series Gaussian Embedding”. In: *The Semantic Web – ISWC 2020*. Cham: Springer International Publishing, 2020, pp. 654–671. ISBN: 978-3-030-62419-4.

[287] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* 2019. arXiv: 1810.00826 [cs.LG].

[288] Lin Yao et al. “Link prediction based on common-neighbors for dynamic social network”. In: 83 (2016), pp. 82–89.

[289] Yao Yao et al. “Deep-learning method for data association in particle tracking”. In: *Bioinformatics* 36.19 (2020), pp. 4935–4941.

[290] Bülent Yener. “Cell-graphs: image-driven modeling of structure-function relationship”. In: *Commun. ACM* 60.1 (Dec. 2016), pp. 74–84. ISSN: 0001-0782. DOI: 10.1145/2960404. URL: https://doi.org/10.1145/2960404.

[291] Jiaxuan You, Tianyu Du, and Jure Leskovec. *ROLAND: Graph Learning Framework for Dynamic Graphs*. 2022. arXiv: 2208.07239 [cs.LG].

[292] Bing Yu, Haoteng Yin, and Zhanxing Zhu. “Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, July 2018. DOI: 10.24963/ijcai.2018/505. URL: https://doi.org/10.24963%2Fijcai.2018%2F505.

[293] Wenchao Yu et al. “NetWalk: A Flexible Deep Embedding Approach for Anomaly Detection in Dynamic Networks”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 2672–2681. ISBN: 9781450355520. DOI: 10.1145/3219819.3220024. URL: https://doi.org/10.1145/3219819.3220024.

[294] Xiang Yue et al. “Graph embedding on biomedical networks: methods, applications and evaluations”. In: *Bioinformatics* 36.4 (Feb. 2020), pp. 1241–1251.

[295] F Zernike. “How I discovered phase contrast”. In: *Science* 121.3141 (Mar. 1955), pp. 345–349.

[296] Chong Zhang, Julian Yarkony, and Fred Hamprecht. “Cell Detection and Segmentation Using Correlation Clustering”. In: vol. 17. Sept. 2014, pp. 9–16. ISBN: 978-3-319-10403-4. DOI: 10.1007/978-3-319-10404-1_2.

[297] T. Y. Zhang and C. Y. Suen. “A fast parallel algorithm for thinning digital patterns”. In: *Communications of the ACM* 27.3 (1984), pp. 236–239.

[298] Xiyue Zhang et al. *Traffic Flow Forecasting with Spatial-Temporal Graph Diffusion Network*. 2021. arXiv: 2110.04038 [cs.LG].

[299] Yu Zhang et al. *Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition*. 2022. arXiv: 2010.10504 [eess.AS]. URL: https://arxiv.org/abs/2010.10504.

[300] Yulun Zhang et al. "Image super-resolution using very deep residual channel attention networks". In: *Computer Vision – ECCV 2018*. Lecture notes in computer science. Cham: Springer International Publishing, 2018, pp. 294–310.

[301] Keliang Zhao et al. "A Novel Deep Learning Approach Featuring Graph-Based Algorithm for Cell Segmentation and Tracking". In: *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Istanbul, Turkiye: IEEE, Dec. 2023, pp. 1752–1757.

[302] Xiaojin Zhu and Zoubin Ghahramani. "Learning from Labeled and Unlabeled Data with Label Propagation". In: *Technical Report CMU-CALD-02-107* (2002).

[303] K. Zuiderveld. "Contrast Limited Adaptive Histogram Equalization". In: *P. Heckbert: Graphics Gems IV*. Academic Press, June 1994.