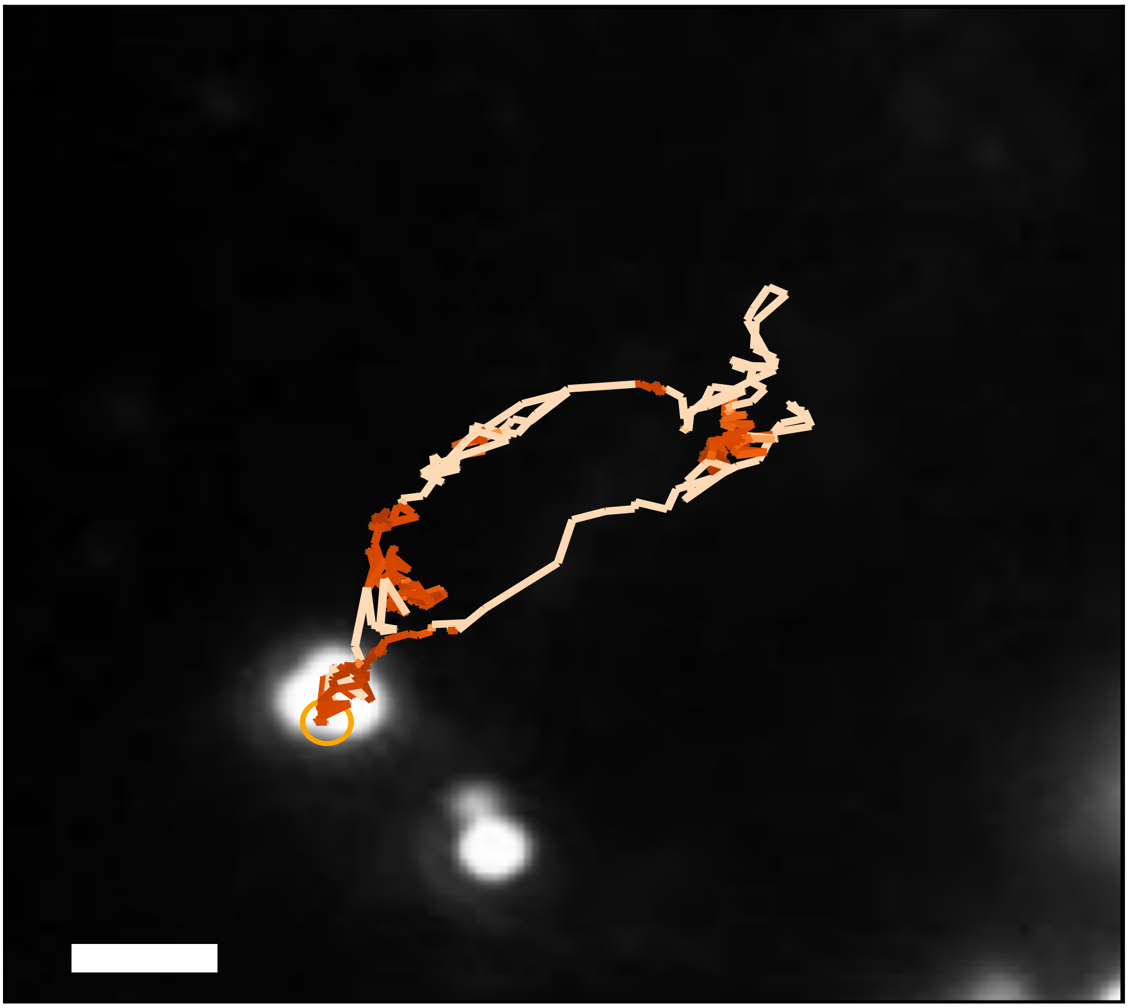
CHALMERS
UNIVERSITY OF TECHNOLOGY



# AI-driven single-particle tracking for cancer cell characterization

## Tracking vesicle-encapsulated nanoparticles as dynamic biomarkers

Master's thesis in Complex adaptive systems

Yuchao He

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

MASTER'S THESIS 2025

# AI-driven single-particle tracking for cancer cell characterization

## Tracking vesicle-encapsulated nanoparticles as dynamic biomarkers

Yuchao He

Department of physics
Soft Matter Lab
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025

AI-driven single-particle tracking for cancer cell characterization
Tracking vesicle-encapsulated nanoparticles as dynamic biomarkers
Yuchao He

Supervisor: Mirja Granfors, Department of Physics; Jesús Pineda, Department of Physics
Examiner: Giovanni Volpe, Department of Physics

Master's Thesis 2025
Department of Physics
Soft Matter Lab
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 73 495 6888

Cover: Visualization of a gold nanoparticle trajectory with colorful segments representing the temporal changes of diffusion exponent values.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

AI-driven single-particle tracking for cancer cell characterization
Tracking vesicle-encapsulated nanoparticles as dynamic biomarkers
Yuchao He
Department of Physics
Chalmers University of Technology

# Abstract

Prostate cancer exhibits altered intracellular phenotypes that require novel biomarker approaches for accurate diagnosis and treatment stratification. This thesis presents an integrated computational framework to reconstruct and analyze the intracellular dynamics of vesicle-encapsulated gold nanoparticles (AuNPs) from large-scale scattering microscopy videos, with the goal of identifying dynamic indices that can serve as potential biomarkers for distinguishing between prostate cancer cell lines.

The proposed framework combines two major components: trajectory reconstruction using deep learning models, specifically the MAGIK graph neural network, and inference of diffusion properties from the reconstructed trajectories. To address the challenges caused by the large scale of the experimental videos, the study introduces a segmentation-based pipeline that processes smaller video sequences, integrates predicted graphs, and builds accurate nanoparticle trajectories. Importantly, the MAGIK model is trained on simulated trajectories that simulate biologically relevant motion types, reducing the need for labor-intensive manual annotations.

Subsequently, the study applies a modified version of the MAGIK model to predict point-wise diffusion exponent values for each trajectory, allowing classification of distinct motion types such as directed motion and subdiffusive motion. Among the extracted dynamic indices, nanoparticle velocities during directed motion emerge as promising biomarkers, showing different distributions between the LNCaP and PC3 prostate cancer cell lines.

Overall, this work demonstrates the potential of applying deep learning methods to uncover novel dynamic biomarkers in cancer research. Future directions include extending the analysis to additional diffusion properties and expanding the repertoire of dynamic indices with biological significance to further enhance biomarker discovery and improve our understanding of intracellular transport in cancer cells.

Keywords: single-particle tracking (SPT), gold nanoparticles (AuNPs), graph neural networks (GNNs), trajectory reconstruction, diffusion exponent inference, intracellular transport.

# Acknowledgements

I would like to express my gratitude to my supervisors Jesús Pineda and Mirja Granfors first. Their insights and explanations greatly deepened my understanding of the models applied in this work. I am also deeply thankful to my collaborators, Hana Jungová and Emelie Tornéus, for giving me the opportunity to tackle this challenge. Their great efforts in preparing the microscopy images made this analysis possible. Finally, I would like to thank Giovanni Volpe for welcoming me into his research group. Working in his group has truly inspired my enthusiasm to pursue scientific research as a lifelong journey.

Yuchao He, Gothenburg, Jun 2025

# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

| | |
|---|---|
| AuNP | Gold Nanoparticle |
| EG6 | Ethylene Glycol Chain |
| FPS | Frames Per Second |
| GNN | Graph Neural Networks |
| LNCaP | Lymph Node Carcinoma of the Prostate Cancer Cell line |
| MAE | Mean Absolute Error |
| MPNN | Message-Passing Neural Network |
| MSD | Mean Squared Displacemen |
| PC3 | Human Prostatic Carcinoma Cell Line |
| PEG | Polyethylene Glycol |
| SPT | Single-particle Tracking |
| TRA | Tracking Accuracy |

# Contents

**List of Acronyms** ix

**List of Figures** xiii

**List of Tables** xvii

**1 Introduction** 1

**2 Background** 3
- 2.1 Endocytic pathways driving nanoparticle intracelluar transport . . . . 3
- 2.2 Framework for single particle tracking . . . . . . . . . . . . . . . . . . 4
- 2.3 Deep learning-based trajectory reconstruction . . . . . . . . . . . . . . 5
- 2.4 Single-trajectory characterization . . . . . . . . . . . . . . . . . . . . . 6
  - 2.4.1 Common dynamics of intracellular nanoparticle transport . . . 6
  - 2.4.2 Inference of diffusion parameters . . . . . . . . . . . . . . . . 7
- 2.5 Graph neural network MAGIK model architectures . . . . . . . . . . 8
  - 2.5.1 Graph representation in MAGIK . . . . . . . . . . . . . . . . . 8
  - 2.5.2 Message-passing neural networks (MPNNs) . . . . . . . . . . . 8

**3 Reconstructing trajectories of intracellular nanoparticle transport** **11**
- 3.1 Model training within the pipeline . . . . . . . . . . . . . . . . . . . . 11
  - 3.1.1 Training the LodeSTAR model . . . . . . . . . . . . . . . . . . 12
  - 3.1.2 Training the MAGIK model . . . . . . . . . . . . . . . . . . . . 12
  - 3.1.3 Generating synthetic data for the training of MAGIK . . . . . 14
- 3.2 Processing experimental microscopy videos through the pipeline . . . 16
  - 3.2.1 Temporal segmentation of experimental videos . . . . . . . . . 17
  - 3.2.2 Graph integration . . . . . . . . . . . . . . . . . . . . . . . . . 17
  - 3.2.3 Trajectory extraction . . . . . . . . . . . . . . . . . . . . . . . 18
- 3.3 Results on synthetic data . . . . . . . . . . . . . . . . . . . . . . . . . 18
  - 3.3.1 Performance evaluation . . . . . . . . . . . . . . . . . . . . . . 18
- 3.4 Results on experimental data . . . . . . . . . . . . . . . . . . . . . . . 20
- 3.5 Discussion and further directions . . . . . . . . . . . . . . . . . . . . 20

**4 Characterizing intracellular nanoparticle transport dynamics** **23**
- 4.1 Inference of nanoparticle diffusion exponent . . . . . . . . . . . . . . 23
  - 4.1.1 Foundations of modified node-specific MAGIK . . . . . . . . . 25
  - 4.1.2 Generating synthetic data for the training . . . . . . . . . . . 26

4.1.3 Results on simulated data . . . . . . . . . . . . . . . . . . . . . 26
4.1.4 Results on experimental data . . . . . . . . . . . . . . . . . . . . 29
4.1.5 Validation of inferred diffusion exponents on experimental datasets 31
4.2 Extracting differentially expressed dynamic indices from nanoparticle trajectories . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
4.3 Discussion and further directions . . . . . . . . . . . . . . . . . . . . . 33

**5 Conclusion** **35**

**Bibliography** **37**

**A Supplementary Information** **I**
A.1 Computation of node attributes for input graph construction . . . . . I
A.2 Pearson correlation coefficients computed from Figure 4.5 . . . . . . . I

# List of Figures

1.1 **Structures of polymer-coated gold nanoparticles (AuNPs)**. **a-b**, Visual examples of two common AuNP architectures: **a** for spherical gold nanoparticles with coated polymers and **b** for rod-shaped gold nanoparticles also coated with polymers. . . . . . . . . . 1

2.1 **Schematic illustration for the endocytic trafficking pathway**. The endocytic trafficking pathway consists of two major stages: endocytosis and ndosomal network processing. Initially, nanoparticles are internalized through endocytic mechanisms and subsequently engage with the cellular endosomal networks. Once inside, they enter into early endosomes, which gradually mature into late endosomes. From here, a portion of internalized particles are transported to lysosomes for degradation, while others are recycled back to the plasma membrane via recycling endosomes. This schematic demonstrates the complete intracellular journey of endocytosed nanoparticles, visualizing key steps in their trafficking. This figure is adapted from Guo et al. [1]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.2 **Framework for single-particle tracking (SPT)**. The SPT framework consists of two major steps: **a** tracking detected particles and **b** analyzing reconstructed trajectories. **a**, After capturing scattering microscopy images, particles are first detected in each frame, followed by a linking process to reconstruct continuous trajectories across frames. Scale bar, 2$\mu$m. **b**, Once trajectories are reconstructed, various analysis tools are applied to extract dynamic properties such as diffusion coefficients, transition rates, and anomalous diffusion exponents, providing quantitative insights into the underlying particle dynamics. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.3 **Representative motion types in intracellular nanoparticle transport**. **a**, Examples of diffusive and motor-driven motion: left panel is the diffusion of membrane protein Sec61 within the peripheral tubular network of the endoplasmic reticulum (ER) in a COS7 cell, and the right panel is a motor-driven lysosome trajectory (red) aligned along microtubules (green) in a monkey kidney cell. This figure is adapted from Saurabh S. Mogre et al. [2]. **b**, Motion with $\alpha = 0.6$, Brownian motion with $\alpha = 1$, and superdiffusive motion with $\alpha = 1.7$. **c**, Corresponding mean squared displacement (MSD) curves for the motion types on a log-log scale shown in **b**. . . . . . . . 7

2.4 **Internal structure of a message-passing neural network (MPNN)**. **a-c**, Visualization of three core update functions: **a** for updating edge attributes, **b** for updating node attributes, and **c** for updating global attributes. In each figure, blue elements represent the components being updated, while orange elements represent the components providing information for the update. **d**, Overview of the MPNN computational process, where node attributes $V$ and edge attributes $E$ are updated to $V'$ and $E'$ through a structured three-step message-passing procedure. **e**, The message-passing steps propagate information to a target node from a limited number of steps. . . . . . . . . . 9

3.1 **Training and detection using the LodeSTAR model. a-b**, Visualization of selected image patches from the microscopy image. **a**, Orange rectangles highlight selected patches from scattering microscopy images, where the bright spots correspond to vesicle-encapsulated AuNPs. Scale bar: $15\mu$m. **b**, Examples displaying the diversity of nanoparticle shapes and intensities. **c**, Visualization of detection results produced by the LodeSTAR model on frame 100; orange rectangles highlight the predicted nanoparticle positions. Scale bar: $14\mu$m. 12

3.2 **Overview of the edge-specific MAGIK model training process**. **a**, Internal architecture of the edge-specific MAGIK model, where orange elements correspond to node-related components and blue elements represent edge-related components. **b**, Example of a synthetic training sample containing multiple simulated trajectories. **c**, Breakdown of node attributes in the input graphs used for the linking task, where each node attributes include the centroid position $(x, y)$, the raw particle intensity $I_r$, and the local background intensity $I_b$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.3 **Structure of the segment-wise trajectory reconstruction pipeline**. **a**, Temporal segmentation: the entire microscopy video is divided into smaller overlapping sequences $S_i$. **b**, Segment-wise edge classification: for each segmented sequence $S_i$, an input graph is constructed and processed by the edge-specific MAGIK model to predict edge connection probabilities. **c**, Graph integration: all predicted graphs $G_i$ are merged into one graph $G_{int}$, combining connections across the entire video. Nodes and edges in green dashed circles represent shared elements between consecutive graphs. **d**, Trajectory extraction: final particle trajectories are reconstructed by tracing the connection probabilities within the integrated graph $G_{int}$. . . . . . . . . . . . . . 16

3.4 **Reconstructed trajectories on synthetic data**. **a-b**, Comparison between reconstructed trajectories (right panels) generated from MAGIK model predictions and the corresponding ground truth, simulated trajectories (left panels). **a**, Example at frame 50; **b**, Example at frame 100. **c**, Example confusion matrix computed from 150 validation samples, used to calculate the Tracking Accuracy (TRA). **d**, Distribution of TRA scores across the validation dataset (150 samples). 19

3.5 **Reconstructed trajectories on experimental videos**. Time evolution of predicted trajectories, shown at three time steps (t = 0, 90, 190 s) under different experimental conditions. Videos were recorded at a frame rate of 1 FPS; scale bar: 2 $\mu$m. **a**, EG6-coated AuNPs within LNCaP cells. **b**, EG6-coated AuNPs within PC3 cells. **c**, PEG-coated AuNPs within LNCaP cells. **d**, PEG-coated AuNPs within PC3 cells. . . . . . . . . . . . . . . . . . . . . . . . . . . 21

4.1 **Training process of the modified node-specific MAGIK model**. **a**, The internal architecture of the modified node-specific MAGIK model, highlighting the flow of node attribute updates. **b-c**, Examples of simulated trajectories used in the training datasets: **b**, mixed motion combining diffusive and directed behaviors; **c**, purely diffusive motion. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

4.2 **Predicted diffusion exponents on synthetic trajectories. a–d**, Examples of individual 2D simulated trajectories undergoing changes of diffusion exponents. The color of the segments represents the value of the diffusion exponent. Specifically, **a–b** display purely diffusive motion, while **c–d** show mixed motion. **e–h**, Time traces of diffusion exponent values along the same trajectories. The ground-truth value used in the simulations is shown by the orange curve. The predictions obtained by MAGIK at the single-node level are shown in grey. **e–f** correspond to purely diffusive motion; **g–h** correspond to mixed motion. 27

4.3 **Performance comparison between the modified and original node-specific MAGIK models on synthetic data. a-b**, Probability distribution of predicted diffusion exponents using the modified node-specific MAGIK model versus ground-truth for simulated trajectories with mixed motion (**a**) and with purely diffusive motion (**b**). **c-d**, Probability distribution of predicted diffusion exponents using the original node-specific MAGIK model versus ground-truth for simulated trajectories with mixed motion (**c**) and with purely diffusive motion (**d**). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

4.4 **Predicted diffusion exponents on reconstructed trajectories from experimental videos.** Left panels show 2D trajectories in experimental videos and the color of their segments represents the diffusion exponent values; middle panels present time traces of diffusion exponent values along the same trajectories (orange trace); right panels present time traces of nanoparticle velocities along the same trajectories (blue trace). **a-c**, Three example trajectories of EG6-coated AuNPs within PC3 cells; **d-f**, Three example trajectories of EG6-coated AuNPs moving within LNCaP cells. . . . . . . . . . . . . 30

4.5 **Validation of inferred diffusion exponents on experimental data. a-b**, Results for experimental videos recording EG6-coated AuNPs within PC3 cells. **a**, Distribution of point-wise diffusion exponent values across all reconstructed trajectories under this condition. **b**, Joint probability distribution of point-wise diffusion exponents versus corresponding particle velocities. **c-d**, Results for experimental videos capturing EG6-coated AuNPs within LNCaP cells. **c**, Distribution of point-wise diffusion exponent values across all reconstructed trajectories under this condition. **d**, Joint probability distribution of point-wise diffusion exponents versus corresponding particle velocities. 32

4.6 **Identified dynamic indices from reconstructed trajectories that show differential expression across cancer cell lines. a**, Distribution comparison of nanoparticle velocities during directed motion under two conditions: EG6-coated AuNPs in PC3 cells (orange) versus EG6-coated AuNPs in LNCaP cells (blue). **b**, Distribution comparison of nanoparticle velocities during directed motion under two conditions: PEG-coated AuNPs in PC3 cells (orange) versus PEG-coated AuNPs in LNCaP cells (blue). . . . . . . . . . . . . . 33

# List of Tables

A.1 Pearson correlation coefficients between point-wise diffusion exponent values and velocities for EG6-AuNPs in different prostate cancer cell lines. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . II

# 1

# Introduction

Prostate cancer ranks as the second most commonly diagnosed cancer in men worldwide, with an estimated 1.4 million new cases and over 375,000 patient deaths recorded each year [3, 4]. While early detection can dramatically improve patient survival, metastatic prostate cancer remains challenging, resulting in a five-year survival rate of only 31% [3]. The development and validation of clinical biomarkers represents a basic approach for improving early detection. However, traditional biomarker assays, such as polymerase chain reaction (PCR)-based tests, DNA microarrays, and fluorescence in situ hybridization, are often invasive, costly, and limited in sensitivity for capturing dynamic molecular events [5].

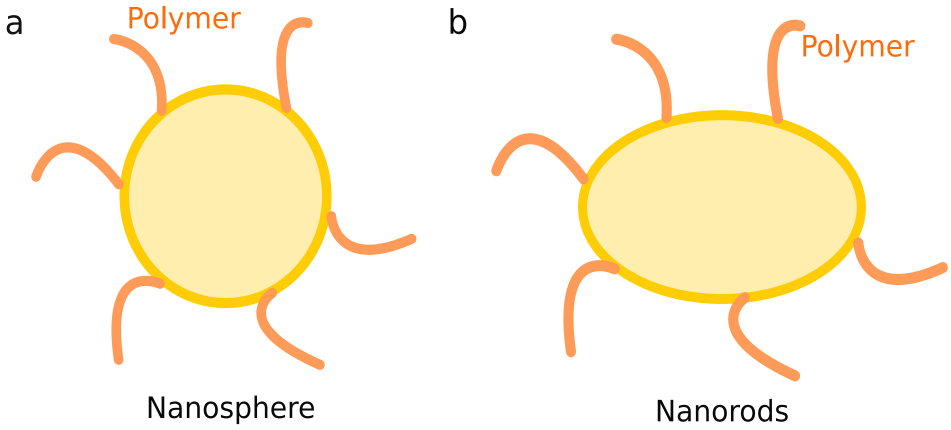


**Figure 1.1: Structures of polymer-coated gold nanoparticles (AuNPs). a-b**, Visual examples of two common AuNP architectures: **a** for spherical gold nanoparticles with coated polymers and **b** for rod-shaped gold nanoparticles also coated with polymers.

Therefore, there is an urgent need to advance cancer diagnostics through the development of novel, non-invasive approaches that target specific biomarkers with enhanced specificity and sensitivity. In recent years, metallic nanomaterials have emerged as powerful diagnostic tools capable of targeting potential cancer biomarkers with remarkable precision [6]. Among these innovative nanomaterials, polymer-coated gold nanoparticles (AuNPs) have gained particular attention due to several distinctive advantages:

- Their intrinsic surface plasmon resonance properties enable highly sensitive optical detection via scattering microscopy techniques;
- Well-designed polymer coatings substantially enhance circulation time within

the human body, while their surface functionalization facilitates specific binding to targeted biomarkers;
- Their exhibited lower toxicity makes them good candidates for potential clinical applications in diagnostics [7].

Given these advanced characteristics, this thesis focuses on polymer-coated gold nanoparticles (AuNPs) as carrier platforms for targeting potential cancer biomarkers. A representative structure of these gold nanoparticles is presented in Figure 1.1 **a-b**.

Having established polymer-coated gold nanoparticles as our carrier platforms, this thesis addresses a fundamental question: Which pathogenic or biogenetic pathways differ among prostate cell lines, and how can these differences be exploited using polymer-coated AuNPs for diagnostic targeting? Recent studies have revealed significant dysregulation of endosomal–lysosomal biogenesis in prostate cancer, characterized by altered expression patterns of genes governing lysosomal trafficking [8]. Notably, several of these differentially expressed genes encode motor proteins responsible for directed vesicle transport along the cytoskeleton. These findings suggest that dysregulated vesicular transport machinery could generate distinctive movement-specific signatures, such as mean squared displacement (MSD), that are differentially expressed between normal and cancerous prostate cell lines.

Considering that polymer-coated gold nanoparticles (AuNPs) can enter intracellular vesicles through endocytic pathways [1], the dynamic behavior of these vesicles can be analyzed by tracking the motion of vesicle-encapsulated AuNPs. The optical properties of AuNPs make this approach feasible: their strong light-scattering signature under dark-field microscopy ensures they are clearly visible in scattering microscopy images, enabling precise particle detection and tracking within the complex cellular environment.

By correlating these particle trajectories with the underlying vesicle trafficking machinery, the central aim of this thesis is to extract motion-specific signatures of vesicle behavior that can serve as novel biomarkers for prostate cancer characterization. These dynamic signatures are computed by analyzing microscopy images that capture the intracellular transport of vesicle-encapsulated AuNPs. To achieve this, the primary objective of this work is to apply deep learning-based single-particle tracking (SPT) methods to extract and analyze vesicle trajectories from large-scale scattering microscopy datasets.

Since SPT broadly consists of two key steps, the thesis consists of two main tasks: developing a robust pipeline to reconstruct vesicle-encapsulated nanoparticle trajectories from large-scale microscopy images, and accurately estimating the diffusion properties at each point along the trajectory, such as diffusion exponents ($\alpha$) and diffusion coefficients ($D$). The structure of this thesis reflects these goals. Chapter 3 presents the trajectory reconstruction pipeline, which integrates pre-trained models to process large-scale microscopy videos. Chapter 4 introduces the modified MAGIK model [9] for the inference of diffusion properties of each point in reconstructed trajectories in Chapter 3. Finally, Chapter 5 discusses the performance evaluation of the proposed methods, key findings, and future directions for improving the models to achieve even more robust and accurate results.

# 2

# Background

In the last chapter, I introduced the ultimate aim of this thesis: identifying novel biomarkers for cancer cell characterization. Given the known alterations in lysosomal trafficking machinery across different prostate cancer cell lines [8], this work focuses specifically on vesicular dynamics related indices as promising biomarker candidates. To achieve this, I use polymer-coated gold nanoparticles (AuNPs) as tracers of intracellular vesicle movement, capturing their dynamics through scattering microscopy. Once these microscopy videos are acquired, a key challenge is to extract meaningful dynamic indices. For this, single-particle tracking (SPT) offers a powerful computational framework to reconstruct trajectories and characterize their dynamic properties [10]. However, traditional computational methods often struggle to interpret the complex movements in biological systems. They tend to be sensitive to outliers and anomalies and typically have a limited scope when it comes to inferring motion parameters [11]. To address these challenges, deep learning has emerged as a promising approach for both particle tracking and the inference of diffusion parameters.

In this chapter, I first introduce the endocytic trafficking pathway, which enables the use of polymer-coated AuNPs to track intracellular vesicle movements. Next, I present the general single-particle tracking (SPT) framework and related deep learning models applied to trajectory reconstruction and characterization. Finally, I introduce the core graph neural network component used in this thesis: the message-passing neural network (MPNN).

## 2.1 Endocytic pathways driving nanoparticle intracelluar transport

Feasibility of tracking intracellular vesicle movements using polymer-coated gold nanoparticles (AuNPs) is enabled by the cell's inherent endocytic trafficking pathways. When introduced into cancer cells, these nanoparticles engage with the plasma membrane through various endocytic mechanisms, including clathrin-mediated endocytosis and macropinocytosis  [1]. After internalization, the nanoparticles enter the endosomal network, where the fate of nanoparticles is determined. Initially, internalized AuNPs enter into early endosomes near the cellular membrane. Then early endosomes mature and move, becoming late endosomes surrounding the nuclei [1]. Eventually, most of them will be transported to lysosomes, mainly around the peri-nuclear region [12]. Together, endocytosis and the endosomal network form the complete endocytic trafficking pathway.

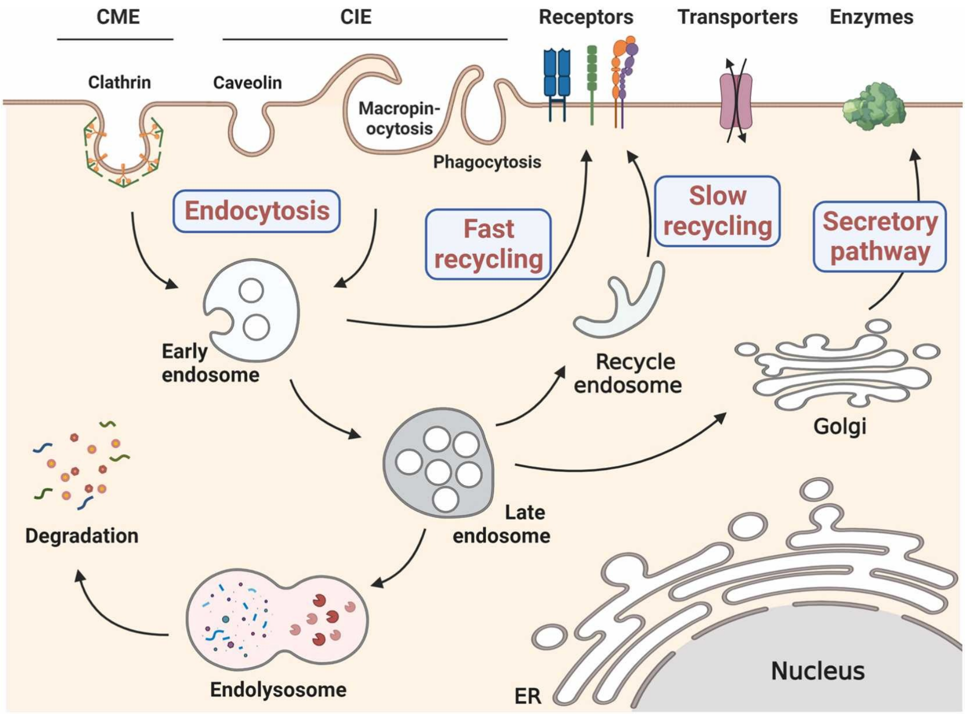


**Figure 2.1: Schematic illustration for the endocytic trafficking pathway**. The endocytic trafficking pathway consists of two major stages: endocytosis and ndosomal network processing. Initially, nanoparticles are internalized through endocytic mechanisms and subsequently engage with the cellular endosomal networks. Once inside, they enter into early endosomes, which gradually mature into late endosomes. From here, a portion of internalized particles are transported to lysosomes for degradation, while others are recycled back to the plasma membrane via recycling endosomes. This schematic demonstrates the complete intracellular journey of endocytosed nanoparticles, visualizing key steps in their trafficking. This figure is adapted from Guo et al. [1].

This pathway provides a valuable opportunity to analyze vesicle dynamics by tracking the movements of vesicle-encapsulated polymer-coated AuNPs. A schematic overview of the full endocytic trafficking process is shown in Figure 2.1.

## 2.2 Framework for single particle tracking

Single-particle tracking (SPT) is an approach to follow and analyze the motion of individual particles within living cells. To obtain precise results, microscopy images must be captured under conditions that allow accurate tracking of nanoparticle movements. In this work, they shared the same frame rate (1 FPS) and scale (1 pixel = 0.1 $\mu$m). After acquiring sufficient microscopy images, the first major step in the SPT pipeline is trajectory reconstruction (Figure 2.2 **a**). This step consists of two major tasks: detecting the position of each nanoparticle in each frame and linking of positions into trajectories. Once the trajectories are reconstructed, the next step is to characterize their diffusion properties (Figure 2.2 **b**). This analysis

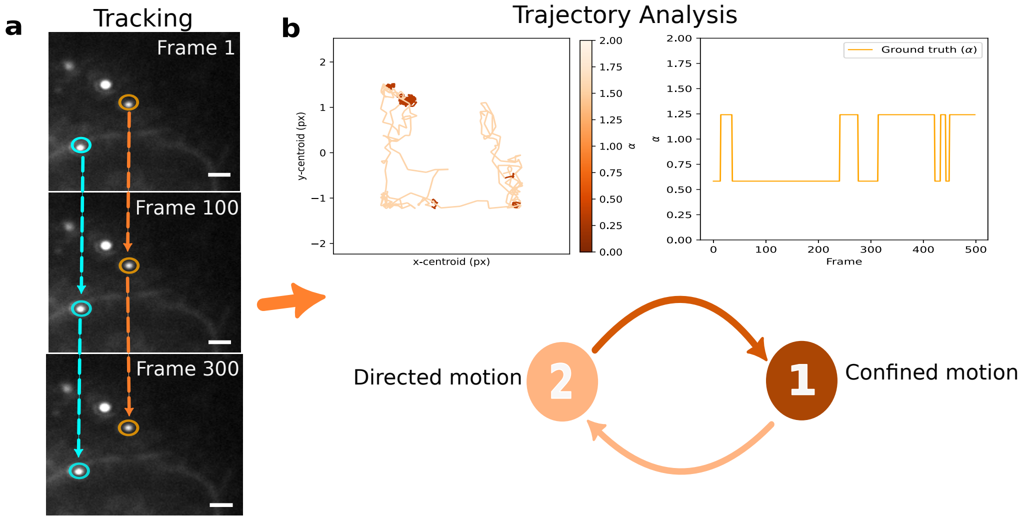


**Figure 2.2: Framework for single-particle tracking (SPT)**. The SPT framework consists of two major steps: **a** tracking detected particles and **b** analyzing reconstructed trajectories. **a**, After capturing scattering microscopy images, particles are first detected in each frame, followed by a linking process to reconstruct continuous trajectories across frames. Scale bar, $2\mu$m. **b**, Once trajectories are reconstructed, various analysis tools are applied to extract dynamic properties such as diffusion coefficients, transition rates, and anomalous diffusion exponents, providing quantitative insights into the underlying particle dynamics.

mainly focuses on two tasks: inferring diffusion parameters and extracting dynamic indices of trajectories based on these inferred parameters, which is closely related to change-point detection in time series analysis [13].

## 2.3 Deep learning-based trajectory reconstruction

The standard pipeline for trajectory reconstruction consists of two major steps: (1) precisely detect the positions of particles in each frame; (2) detected positions at each frame are connected into trajectories (linking) [14]. Numerous algorithmic solutions have been proposed to tackle these two tasks. Specifically, current deep-learning revolution has fostered the development of tracking reconstruction methods. Therefore, this section focuses specifically on deep-learning-based tracking approaches.

For the particle detection task, traditional deep learning models based on supervised learning often require large, precisely labeled datasets, which can be time-consuming to generate in biological fields. However, a recently developed model, LodeSTAR, overcomes this challenge by combining self-supervised learning and geometric learning to eliminate the need for manual annotations [15]. Notably, LodeSTAR can be trained on images containing a single object and then detect multiple objects with similar shapes and optical properties to objects within trained images, making it highly suitable for detecting the particles in my datasets, which include nanoparticles with similar shapes in scattering microscopy images.

For the trajectory linking task, geometric deep learning provides advanced approaches to tackle multi-object tracking from a different perspective compared to traditional methods [14]. Instead of dealing with information from detected objects directly, geometric deep learning will extend neural networks to problems where inputs are best represented as graphs. For the trajectory linking task, graphs provide a natural and intuitive way to represent both spatial and temporal information from detected objects [16,17]. Therefore, deep-learning methods based on graphs referred to as graph neural networks (GNNs) have been successfully applied to the trajectory linking task, such as MAGIK [9].

## 2.4 Single-trajectory characterization

After discussing the deep learning models for trajectory reconstruction, the first step in the SPT framework, this section introduces the second step in the SPT framework, single-trajectory characterization. To identify dynamic related indices as potential biomarkers, the quantitative characterization of single-trajectory diffusion properties is essential and typically related to the inference of diffusion properties [13, 18]. This process relies on understanding the types of nanoparticle motion present in biological environments. Therefore, before diving into the analysis, it is important to first discuss the main motion types observed for vesicle-encapsulated AuNPs within cancer cells.

### 2.4.1 Common dynamics of intracellular nanoparticle transport

The motion of vesicle-encapsulated AuNPs reflects the dynamics of the surrounding vesicles. According to the work by Saurabh S. Mogre et al. [2], two primary types of vesicular motion are typically observed within cells. When nanoparticles are wrapped in late endosomes or lysosomes in the perinuclear region, they often exhibit subdiffusive motion type [19]. In contrast, when nanoparticles are wrapped in vesicles actively transported by motor proteins along cytoskeletal filaments, they are driven by motor proteins and display directed motion type [20, 21]. These two representative motion types in the physical world are displayed in Figure 2.3 **a**.

A key question, then, is: how can these motion types be described mathematically? A typical method is to use point-wise diffusion exponent values to determine motion types [22]. The computation of diffusion exponents ($\alpha$) is derived from the scaling of the mean squared displacement (MSD) with lag time $\tau$, given as:

$$\mathrm{MSD}(\tau) = \langle [\mathbf{x}(t+\tau) - \mathbf{x}(t)]^2 \rangle = 2d\, D_\alpha\, \tau^\alpha \tag{2.1}$$

where $d$ is the dimensionality, $D_\alpha$ is the generalized diffusion coefficient, $\mathbf{x}(t)$ is the position of nanoparticles at time $t$, and $\alpha$ is the diffusion exponent.

By calculating $\alpha$, the motion types can be classified using threshold values [22]:

- If $\alpha < 1$, the MSD grows sublinearly, corresponding to subdiffusive motion type.
- If $1 \leq \alpha < 1.5$, the MSD grows linearly, corresponding to weakly directed or Brownian-like motion.

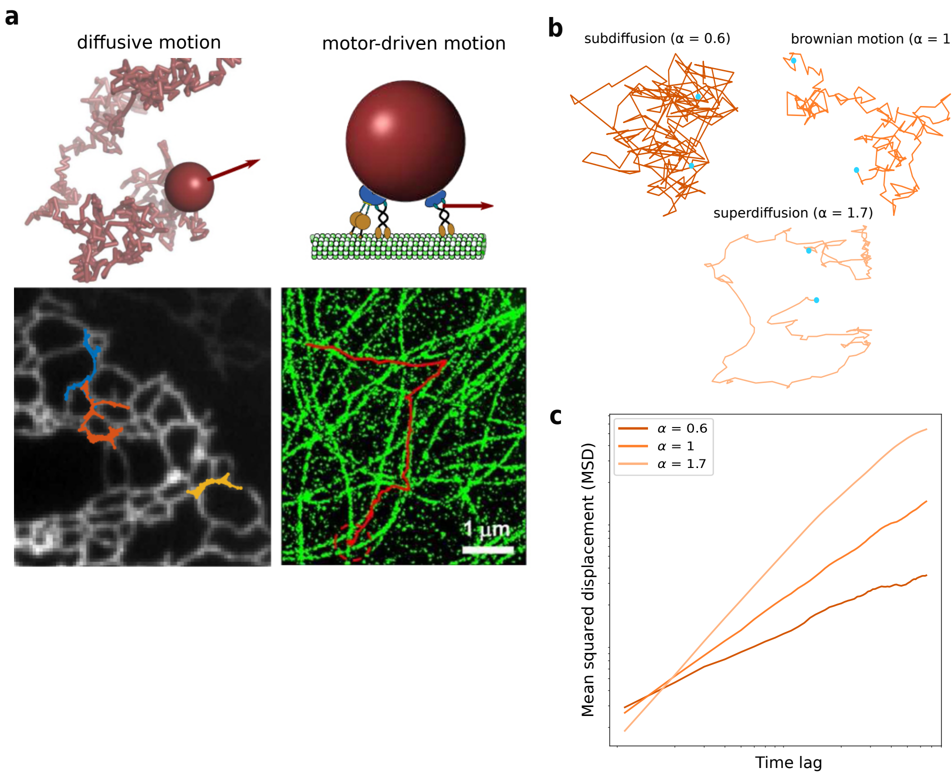


**Figure 2.3: Representative motion types in intracellular nanoparticle transport**. **a**, Examples of diffusive and motor-driven motion: left panel is the diffusion of membrane protein Sec61 within the peripheral tubular network of the endoplasmic reticulum (ER) in a COS7 cell, and the right panel is a motor-driven lysosome trajectory (red) aligned along microtubules (green) in a monkey kidney cell. This figure is adapted from Saurabh S. Mogre et al. [2]. **b**, Motion with $\alpha = 0.6$, Brownian motion with $\alpha = 1$, and superdiffusive motion with $\alpha = 1.7$. **c**, Corresponding mean squared displacement (MSD) curves for the motion types on a log-log scale shown in **b**.

- If $\alpha >= 1.5$, the MSD grows superlinearly corresponding to strongly directed motion.

These motion types and their MSD curves are visualized in Figure 2.3 **b-c**, where Figure 2.3 **c** illustrates the linear relationship between the MSD and lag time on a log-log scale, with the slope corresponding to the diffusion exponent $\alpha$.

### 2.4.2 Inference of diffusion parameters

Once the possible motion types are established, the next step is to estimate the diffusion parameters, such as the diffusion coefficients ($D$) and diffusion exponents ($\alpha$). Before the utilization of machine learning, statistics-based methods have been developed to infer these diffusion parameters, such as time-averaged mean squared displacement (TA-MSD), hidden Markov model and variational Bayesian methods [20]. However, these statistics-based methods are sensitive to outliers, and typically have a narrow scope when performing parameter inference.

To address this challenge, deep learning methods have been introduced to the inference task. Convolutional neural networks (CNNs) [23], recurrent neural networks (RNNs) [24], and graph neural networks (GNNs) [9, 25] are among the most widely used architectures. Compared to CNNs and RNNs, GNNs offer a distinct advantage in modeling complex spatial–temporal relationships in intracellular transport data [9]. In this work, I adopt and extend a GNN-based framework, specifically building on the MAGIK architecture [9], to tackle the inference task more precisely.

## 2.5 Graph neural network MAGIK model architectures

As discussed in the previous sections, graph neural networks (GNNs) have emerged as powerful tools for both the trajectory linking task and trajectory characterization task. In this work, I selected the MAGIK model [9], a GNN-based architecture, as the foundational model for linking nanoparticle detections across frames and inferring diffusion-related parameters. This section presents the fundamental principles of graph neural networks underlying the MAGIK model design.

### 2.5.1 Graph representation in MAGIK

The effectiveness of graph neural networks highly depends on how the underlying task is represented as a graph structure. In the MAGIK model, each input graph is formally defined as a 3-tuple $\mathcal{G} = (\mathbf{u}, \mathbf{V}, \mathbf{E})$, where:

- $\mathbf{u}$ represents global attributes that characterize the entire input graph.
- $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{N^v}$ is the set of nodes, where each $\mathbf{v}_i$ contains the attribute vector for node $i$ and $N^v$ is the total number of nodes
- $\mathbf{E} = \{(e_k, r_k, s_k)\}_{k=1}^{N^e}$ represents the set of edges, where $r_k$ indicates the receiver node index belonging to the edge $k$, $s_k$ indicates the sender node index belonging to the edge $k$, and $e_k$ contains the attribute vector for edge $k$.

In the nanoparticle trajectory analysis, nodes ($\mathbf{V}$) typically represent detected nanoparticles with their associated features (position, intensity, frame number), while edges ($\mathbf{E}$) encode relationships between particles, such as Euclidean distances and temporal relations. This graph representation captures both spatial and temporal relationships within detected nanoparticles across the video for trajectory reconstruction and analysis.

### 2.5.2 Message-passing neural networks (MPNNs)

The core computational blocks of the MAGIK model are based on message-passing neural networks (MPNNs), which provide a flexible and powerful approach for processing graph-structured datasets. Specifically adapted for the trajectory analysis tasks in this work, the MAGIK model implemented focuses primarily on learning and updating node and edge attributes while maintaining global attributes as static throughout the computation process.

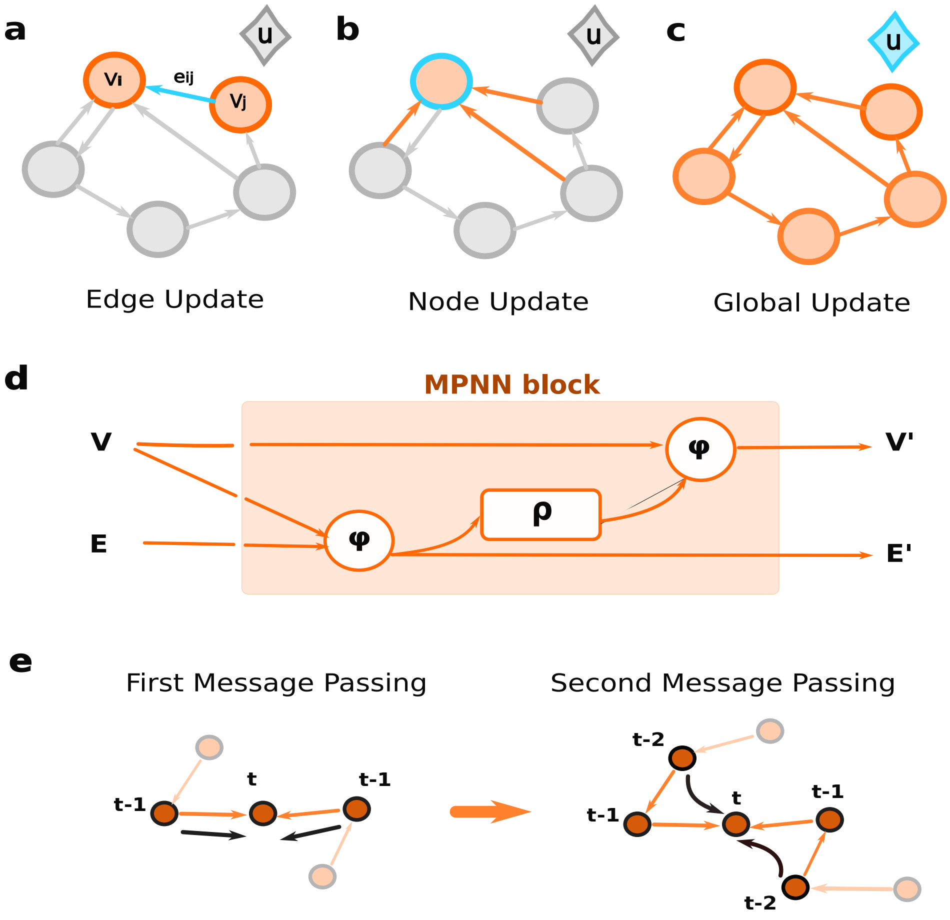


**Figure 2.4: Internal structure of a message-passing neural network (MPNN)**. **a-c**, Visualization of three core update functions: **a** for updating edge attributes, **b** for updating node attributes, and **c** for updating global attributes. In each figure, blue elements represent the components being updated, while orange elements represent the components providing information for the update. **d**, Overview of the MPNN computational process, where node attributes $V$ and edge attributes $E$ are updated to $V'$ and $E'$ through a structured three-step message-passing procedure. **e**, The message-passing steps propagate information to a target node from a limited number of steps.

The internal structure of MPNNs contains two fundamental functions: update functions ($\phi$) and aggregation functions ($\rho$). The update functions ($\phi$) are responsible for generating new attribute values for nodes, edges, or global features by incorporating information from neighboring elements. As displayed in Figure 2.4 **a-c**, there are three types of update functions, each for updating node attributes, edge attributes, or global attributes, respectively.

The computation within each MPNN block follows a structured three-step process (shown in Figure 2.4 **d**):

1. **Edge Update**: Edge attributes are updated based on the features of their connected nodes, allowing the model to learn relationships between particles

based on their individual characteristics.

2. **Edge Aggregation**: For each node, connected edge attributes are aggregated to create a comprehensive representation of the local neighborhood structure.
3. **Node Update** Node attributes are updated using both the original node features and the aggregated edge information, enabling each particle to incorporate information from its local environment.

This iterative message-passing procedure enables information propagation across the graph structure. After certain steps of propagation, each node has access to information from all nodes and edges [26]. This multi-hop information aggregation is particularly valuable for trajectory analysis, as it allows the model to capture both local motion patterns and longer-range temporal relations. This aggregation process is displayed in Figure 2.4 **e**.

Additionally, by using message passing to propagate information across temporal connections, the MPNN can effectively capture the sequential relationships for understanding particle transport dynamics. This capability makes the architecture particularly suitable for inferring diffusion parameters from particle trajectory data.

# 3

# Reconstructing trajectories of intracellular nanoparticle transport

In the previous chapter, I explored methods for detecting nanoparticles and linking detections across video frames to form trajectories. While these approaches provide practical solutions, they present significant limitations with respect to the precision-memory tradeoff. Furthermore, current approaches lack a unified framework for processing the high-throughput video data generated by modern microscopy techniques.

To address these limitations, this chapter introduces a robust computational pipeline designed to reconstruct nanoparticle trajectories directly from raw microscopy videos with enhanced efficiency and accuracy. My approach integrates deep learning methodologies within a two-stage framework: (1) training neural network models on designed synthetic datasets that capture the features of experimental data; (2) applying this pipeline integrated with these pre-trained models to reconstruct trajectories from experimental microscopy videos.

The following sections provide detailed technical descriptions of each pipeline component, including the synthetic data generation for the model, training and integration strategies. I demonstrate the pipeline's performance through the validation on both synthetic datasets and experimental datasets acquired under diverse imaging conditions, establishing its robustness and practical utility for intracellular transport studies.

## 3.1 Model training within the pipeline

The trajectory reconstruction task contains two major steps: detecting nanoparticles in each microscopy image and linking detected nanoparticles to construct corresponding trajectories. Therefore, two deep learning models are trained within this pipeline: lodeSTAR model [15] for the nanoparticle detection and MAGIK model [9] for linking detected nanoparticles. To train these models, it is essential to prepare training datasets representative of the experimental data. However, for microscopy-based particle tracking, especially within densely populated intracellular environments, manually annotating nanoparticle trajectories faces significant challenges. The high density and similarity of nanoparticles in microscopy images make it difficult for human annotators to accurately distinguish and label individual particles. To overcome these challenges, synthetic training data have been employed to simulate complex nanoparticle motions within cells [27].

### 3.1.1 Training the LodeSTAR model

To train the lodeSTAR model, an unlabeled dataset of image patches is constructed, each containing exactly one nanoparticle, cropped from experimental microscopy videos [15]. Since LodeSTAR is based on self-supervised learning, no manual annotations are required. All patches are standardized to a fixed size, for example, $30 \times 30\,\mathrm{px}^2$ in this work to ensure uniformity of the input. To capture the natural heterogeneity of our nanoparticles, I sample patches from multiple frames and include nanoparticles of varying shapes and contrast levels (Figure 3.1 **b**).

After the training process, the performance of the trained lodeSTAR model is validated on scattering microscopy images. In Figure 3.1**a**–**b**, each bright spot represents a vesicle containing nanoparticles, made visible due to light scattering and reflection. The results demonstrate that lodeSTAR achieves high accuracy in detecting nanoparticles in these microscopy images, indicating its effectiveness for this work (Figure 3.1 **c**).

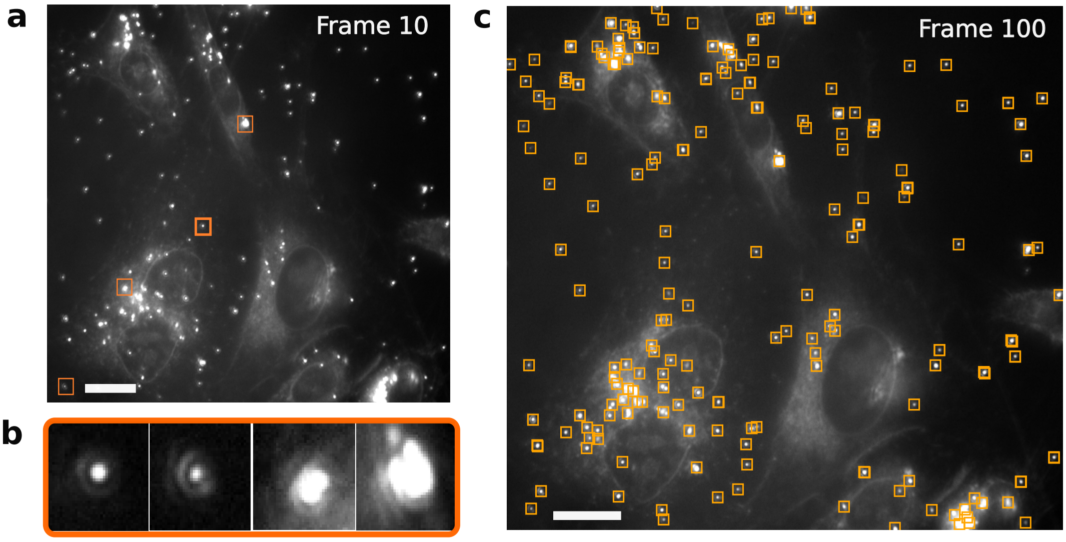


**Figure 3.1: Training and detection using the LodeSTAR model. a-b**, Visualization of selected image patches from the microscopy image. **a**, Orange rectangles highlight selected patches from scattering microscopy images, where the bright spots correspond to vesicle-encapsulated AuNPs. Scale bar: 15$\mu$m. **b**, Examples displaying the diversity of nanoparticle shapes and intensities. **c**, Visualization of detection results produced by the LodeSTAR model on frame 100; orange rectangles highlight the predicted nanoparticle positions. Scale bar: 14$\mu$m.

### 3.1.2 Training the MAGIK model

To link detected nanoparticles and reconstruct their trajectories, this work applies the edge-specific MAGIK model. The model formulates the trajectory reconstruc-

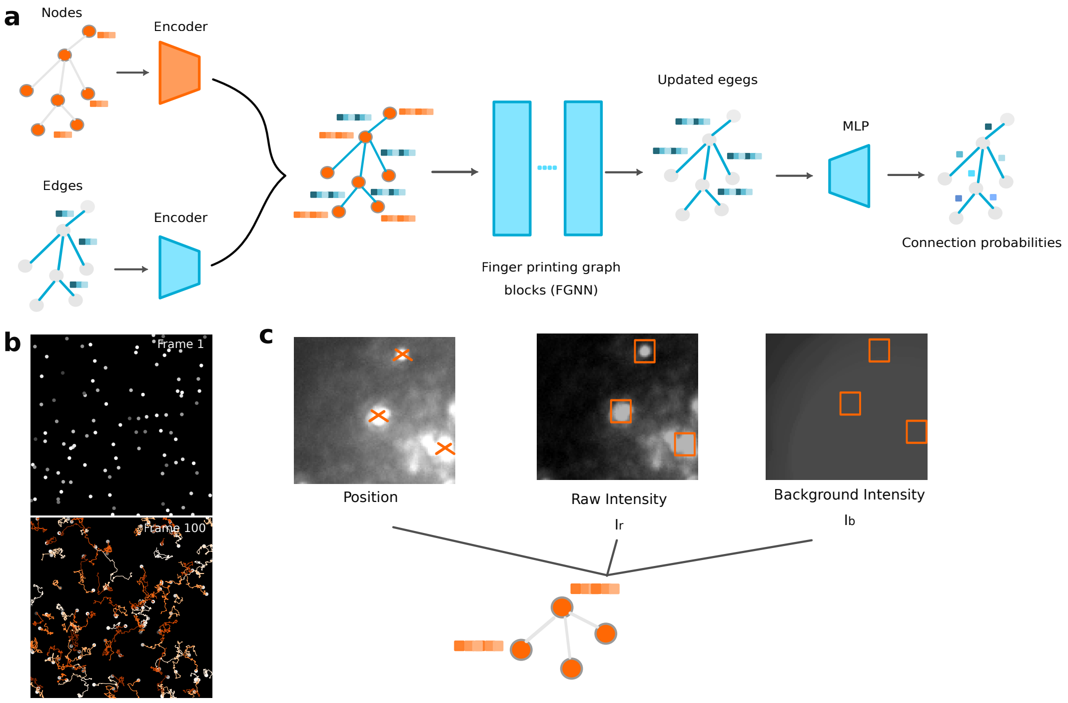


**Figure 3.2: Overview of the edge-specific MAGIK model training process**. **a**, Internal architecture of the edge-specific MAGIK model, where orange elements correspond to node-related components and blue elements represent edge-related components. **b**, Example of a synthetic training sample containing multiple simulated trajectories. **c**, Breakdown of node attributes in the input graphs used for the linking task, where each node attributes include the centroid position $(x, y)$, the raw particle intensity $I_r$, and the local background intensity $I_b$.

tion task as an edge-classification problem, enabling it to predict the connection probability between two detected nanoparticles across consecutive frames. A critical pre-processing step before applying the model is the graph construction process. Here, I generate input graphs where each node corresponds to a detected nanoparticle, and each edge represents a potential relationship between detected particles in adjacent frames.

Specifically, these input graphs are constructed using synthetic trajectories that serve as ground truth data. Each candidate edge is assigned a binary ground truth label $p \in \{0, 1\}$, where 1 signifies a verified connection between detected particles (i.e., detections of the same nanoparticle across consecutive frames) and 0 indicates the absence of a connection (i.e., detections of different nanoparticles across consecutive frames). This binary labeling transforms the trajectory reconstruction challenge into a supervised learning task.

During training, the MAGIK model keeps node attributes fixed while updating edge attributes that encode connection probabilities. The model is optimized by minimizing the binary cross-entropy loss between the predicted connection probabilities $\hat{P}$ and the corresponding ground truth labels $P$ [9]. This optimization strategy enables the model to effectively learn the complex spatiotemporal patterns that distinguish genuine particle connections from spurious associations, ultimately yielding accurate trajectory reconstructions even in challenging imaging conditions.

The edge-specific MAGIK model architecture is illustrated in Figure 3.2 **a**. Since our aim is only to predict edge connection probabilities, we modify the original MAGIK framework to reduce computation. The simplified model consists of three stages:

1. **Feature encoding:** Node attributes and edge attributes are passed through encoder networks (MLPs) separately.
2. **Graph fingerprinting:** The encoded graph is processed by a stack of finger printing graph blocks. To pass the information of potential useful nodes, the design of graph blocks is based on message passing neural networks.
3. **Edge decoding:** The updated edge embeddings are passed through a small MLP to predict the connection probability for each edge.

Compared to the original MAGIK [9], the edge-specific MAGIK model replaces the node-update self-attention mechanism with a simple dense layer (discussed in Pineda, J. et al. [9]). This modification is motivated by the observation that, for this task, all neighboring nodes are equally important for the linking prediction. Therefore, the self-attention mechanism, which dynamically learns attention weights, adds unnecessary computation.

For trajectory linking task in this work, the edge-specific MAGIK model is trained for 50 epochs using the 1024 training samples processed in batches of 8 samples per iteration. In addition, each sample considered 100-200 particle trajectories within $1000 \times 1000\,\mathrm{px}^2$ region.

### 3.1.3 Generating synthetic data for the training of MAGIK

As discussed in the last section, the training datasets for the MAGIK model's linking task are built on synthetic data, instead of using manually labeled experimental data. In this work, synthetic training datasets consist of labeled samples, which are generated by using the `andi-datasets` Python package [13]. Each sample in synthetic datasets contains multiple simulated trajectories of varying lengths, representing nanoparticle movements independently. In each sample, trajectories are sampled from two major motion types observed in vesicle dynamics: (1) **subdiffusive motion:** simulating vesicles confined near the nucleus (anomalous exponent $\alpha < 1$); (2) **mixed motion:** simulating vesicles distant from the nucleus that exhibit both subdiffusive and directed (strong superdiffusive) behaviors, the latter occurring when vesicles attach to cytoskeletal filaments, resulting in directed transport [28]. These scenarios will serve as ground truth for edge labels (Figure 3.2 **b**).

For the construction of input graphs, each node in graphs represents a detected nanoparticle with attributes extracted from the fixed-size image patches (e.g. $30 \times 30\,\mathrm{px}^2$) centered on each detection. In scattering microscopy images, node attributes include (displayed in Fig. 3.2 **c**):

- **Centroid coordinates** $(x, y)$**:** positions of detected nanoparticles within the scattering microscopy image;
- **Raw particle intensity** $(I_r)$: the measured scattering intensity associated with nanoparticles;
- **Local background intensity** $(I_b)$**:** the measured background intensity within

the extracted image patches.

The computation of these node attributes is illustrated in Appendix A.1.

While centroid positions can be directly sampled from the simulated trajectories, accurate modeling of intensity variations requires additional consideration to ensure synthetic data close to experimental videos. To address this problem, I implement a structured approach for simulating intensity values along each trajectory. The procedure contains three major steps:

1. **Distribution characterization:** The first step is to quantify the statistical distributions of raw intensity ($I_r$) and background intensity ($I_b$) from experimental data, calculating their means ($\mu_{I_r}^{\exp}$, $\mu_{I_b}^{\exp}$) and standard deviations ($\sigma_{I_r}^{\exp}$, $\sigma_{I_b}^{\exp}$).
2. **Initialization:** For each simulated trajectory, intensity values of nanoparticles in the first frame are initialized according to:
   (a) Initialization of raw intensity:

$$I_{r,i}^{(1)} \sim \mathcal{N}\left(\mu_{I_r}^{\exp}, \sigma_{I_r}^{\exp}\right) \quad \text{subject to} \quad 1 \leq I_{r,i}^{(1)} \leq 2^k \tag{3.1}$$

   (b) Initialization of background intensity:

$$I_{b,i}^{(1)} \sim \mathcal{N}\left(\mu_{I_b}^{\exp}, \sigma_{I_b}^{\exp}\right) \quad \text{subject to} \quad 1 \leq I_{b,i}^{(1)} \leq 2^k - I_{r,i}^{(1)} \tag{3.2}$$

   where $I_{r,i}^{(1)}$ represents the raw intensity of the $i^{th}$ trajectory at the first frame; $I_{b,i}^{(1)}$ represents the background intensity of the $i^{th}$ trajectory at the first frame; $\mathcal{N}$ represents the normal distribution; $k$ represents the bit depth of the image, and the constraints ensure physically realizable intensity values.
3. **Temporal evolution:** For subsequent frames in each trajectory, intensity values are generated following rules that reflect experimental variation patterns:

$$I_{r,i}^{(t)} \sim \mathcal{N}(I_{r,i}^{(1)}, \sigma_r) \quad \text{subject to} \quad 1 \leq I_{r,i}^{(t)} \leq 2^k \tag{3.3}$$

$$I_{b,i}^{(t)} \sim \mathcal{N}(I_{b,i}^{(1)}, \sigma_b) \quad \text{subject to} \quad 1 \leq I_{b,i}^{(t)} \leq 2^k - I_{r,i}^{(t)} \tag{3.4}$$

   where $I_{r,i}^{(1)}$ represents the raw intensity of the $i^{th}$ trajectory at the frame $t$ ($t > 1$); $I_{b,i}^{(1)}$ represents the background intensity of the $i^{th}$ trajectory at the frame $t$ ($t > 1$); and $\sigma_r \ll \sigma_b$, reflecting the observation that the raw intensity exhibits greater stability than the background intensity, as nanoparticle merging or dividing events occur infrequently to influence raw intensities.

This simulation framework captures the temporal intensity changes observed in experimental videos, thereby enhancing the biological relevance of our validation datasets.

Additionally, Edge attributes are defined as Euclidean distances between connected nodes, providing spatial relations for the model's predictions. By training on these varied, fully annotated synthetic graphs, the edge-specific MAGIK model enables analyzing the varied experimental data.

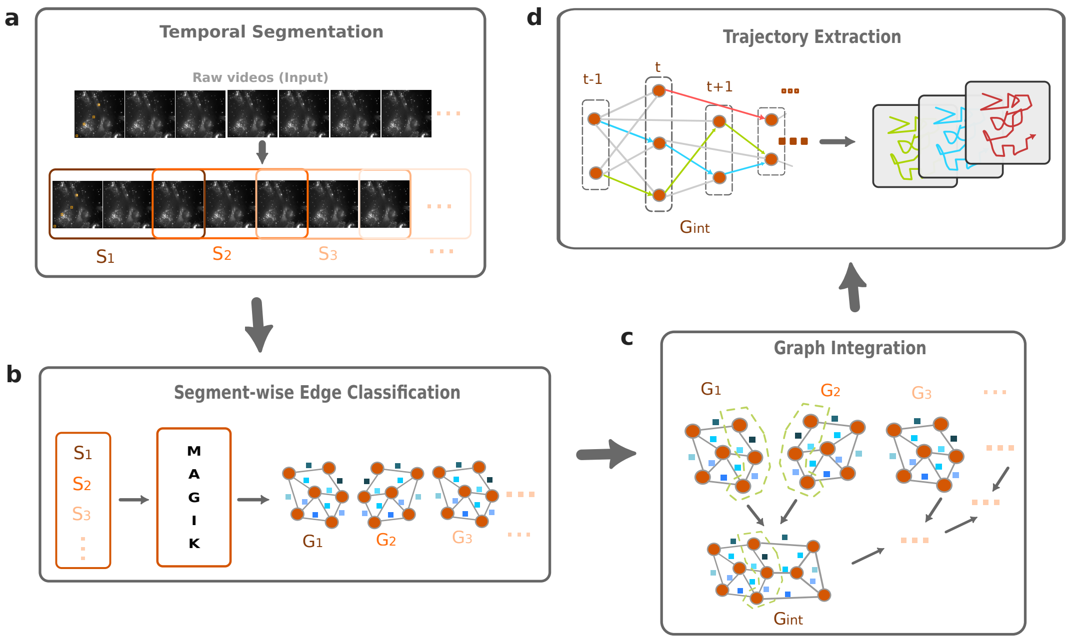


**Figure 3.3: Structure of the segment-wise trajectory reconstruction pipeline**. **a**, Temporal segmentation: the entire microscopy video is divided into smaller overlapping sequences $S_i$. **b**, Segment-wise edge classification: for each segmented sequence $S_i$, an input graph is constructed and processed by the edge-specific MAGIK model to predict edge connection probabilities. **c**, Graph integration: all predicted graphs $G_i$ are merged into one graph $G_{int}$, combining connections across the entire video. Nodes and edges in green dashed circles represent shared elements between consecutive graphs. **d**, Trajectory extraction: final particle trajectories are reconstructed by tracing the connection probabilities within the integrated graph $G_{int}$.

## 3.2 Processing experimental microscopy videos through the pipeline

As deep learning models discussed before are pre-trained, straightforward approach would be to construct input graphs from entire experimental videos and predict connection probabilities through the MAGIK model. However, experimental microscopy videos typically contain hundreds or thousands of frames (often more than 500 frames), presenting significant computational challenges for graph construction and prediction. To address the memory limitations of laboratory laptops, I developed a segmented processing pipeline that maintains trajectory continuity. This pipeline contains four major stages:

1. **Temporal segmentation:** The input video is segmented into shorter, overlapping sequences of manageable length.
2. **Segment-wise graph prediction:** For each sequence, corresponding input graphs are constructed and applied to the pre-trained MAGIK model, predicting edge connection probabilities.
3. **Graph integration:** Predicted graphs from consecutive segments are merged

through their overlapping frames to form a unified trajectory graph for the entire video.

4. **Trajectory extraction:** Connected components analysis is performed on the integrated graph to identify continuous nanoparticle trajectories.

The framework of this pipeline is described in Figure 3.3.

### 3.2.1 Temporal segmentation of experimental videos

To reduce the memory usage, instead of inputting the entire video at once, I divide the video into smaller, overlapping sequences $S_i$. This segmentation strategy allows the pipeline to process smaller sizes of data while keeping the temporal consistency of the data. In addition, the method relies on maintaining sufficient frame overlap between consecutive segments to ensure that trajectory fragments remain continuous across the full video.

As illustrated in Figure 3.3 **a**, each segment shares multiple frames with its neighbors, creating anchor points where trajectory fragments can be connected. Without this overlap, predictions from individual segments would remain isolated, making it impossible to reconstruct continuous trajectories across the entire video.

### 3.2.2 Graph integration

For a video segmented into $N$ overlapping sequences, each sequence $S_i$ (where $i \in \{1, 2, ..., N\}$) will generate a predicted graph $G_i$ through MAGIK. This set of predicted graphs from the set of sequences is then defined as:

$$\{G_i = (V_i, E_i, \hat{P}_i) \mid i = 1, 2, \ldots, N\} \tag{3.5}$$

where:

- $V_i$ is the set of detected nanoparticles (nodes),
- $E_i$ is the set of predicted edges (connections),
- $\hat{P}_i$: is the set of predicted edge connection probabilities from MAGIK.

An integrated graph $G_{\text{int}}$ is then generated from this set of isolated predicted graphs, which is defined as:

$$G_{\text{int}} = (V\text{int}, E\text{int}, \hat{P}\text{int}) \tag{3.6}$$

with:

$$V\text{int} = \bigcup_{i=1}^{N} V_i, \qquad E\text{int} = \bigcup_{i=1}^{N} E_i, \qquad \hat{P}\text{int} = \bigcup_{i=1}^{N} \hat{P}_i' \tag{3.7}$$

Specifically, for the integration of predicted graphs, the modified connection probabilities $\hat{P}_i'$ based on $\hat{P}_i$ for each edge $e$ within $E_i$ is defined as:

$$\hat{P}_i'(e) = \begin{cases} \max\left(\hat{P}_i(e), \hat{P}_j(e)\right) & \text{if } e \in E_i \cap E_j, \forall j \neq i \\ \hat{P}_i(e) & \text{if } e \in E_i \text{ and } e \notin E_j, \forall j \neq i \end{cases} \tag{3.8}$$

This part is visualized in Figure 3.3 **b-c**.

### 3.2.3 Trajectory extraction

Nanoparticle trajectories are built on the integrated graph $G_{\text{int}}$, where each edge is associated with a connection probability. To extract trajectories from $G_{\text{int}}$, this graph is first binarized by thresholding the connection probabilities as follows:

$$\hat{P}_{\text{int}}(e) = \begin{cases} 1 & \text{if } \hat{P}_{\text{int}}(e) \geqslant 0.5, \\ 0 & \text{if } \hat{P}_{\text{int}}(e) < 0.5. \end{cases} \tag{3.9}$$

where a binary value of 1 indicates a strong connection between two nodes (particles), and 0 indicates no connection.

Trajectory reconstruction then proceeds iteratively. Starting from randomly selected nodes in the first frame, we follow edges labeled with 1 to trace forward connections to subsequent frames. When a node has no outgoing edges labeled with 1, the trajectory is terminated. Once a complete trajectory is extracted, all nodes within this trajectory are removed from the graph. This process is repeated until all nodes in $G_{\text{int}}$ have been assigned to a trajectory or removed. The output is a set of independent trajectories, which is also the final output of the pipeline. This section is also illustrated in Figure 3.3 **d**.

## 3.3 Results on synthetic data

Before testing the pipeline on experimental microscopy videos, I first validate its performance on synthetic datasets. The synthetic data used for validation are generated following the same procedure as the training datasets, but with different parameters. Specifically, each scenario contains between 100 and 200 trajectories, each around 500 frames long. These complex scenarios are designed such that accurate prediction requires multiple inference steps, while maintaining motion patterns that closely resemble those observed in experimental videos.

The key results are shown in Figure 3.4 **.** Figure 3.4 **a–b** presents the reconstructed trajectories from our pipeline, compared to the ground truth, for frames 50 and 100. Even in scenarios with dense particle populations and varying motion types, the pipeline successfully reconstructs trajectories with high accuracy.

### 3.3.1 Performance evaluation

To quantify the tracking performance, I use the Tracking Accuracy (TRA) metric based on the Acyclic Oriented Graph Matching (AOGM) framework introduced in [29]. In this framework, both the predicted and ground truth nanoparticle connections are represented as directed graphs, where nodes represent detected particles at each frame and edges represent connections between particles (whether belonging to the same objects).

The AOGM measures the weighted sum of the executed operations to transform the predicted graph into the ground-truth one [29]. Specifically, for this linking task, only edge-based operations are considered, assuming no vesicle division or merging. Thus, the following operation weights are used for this linking task: $w_{NS} = w_{EC} =$

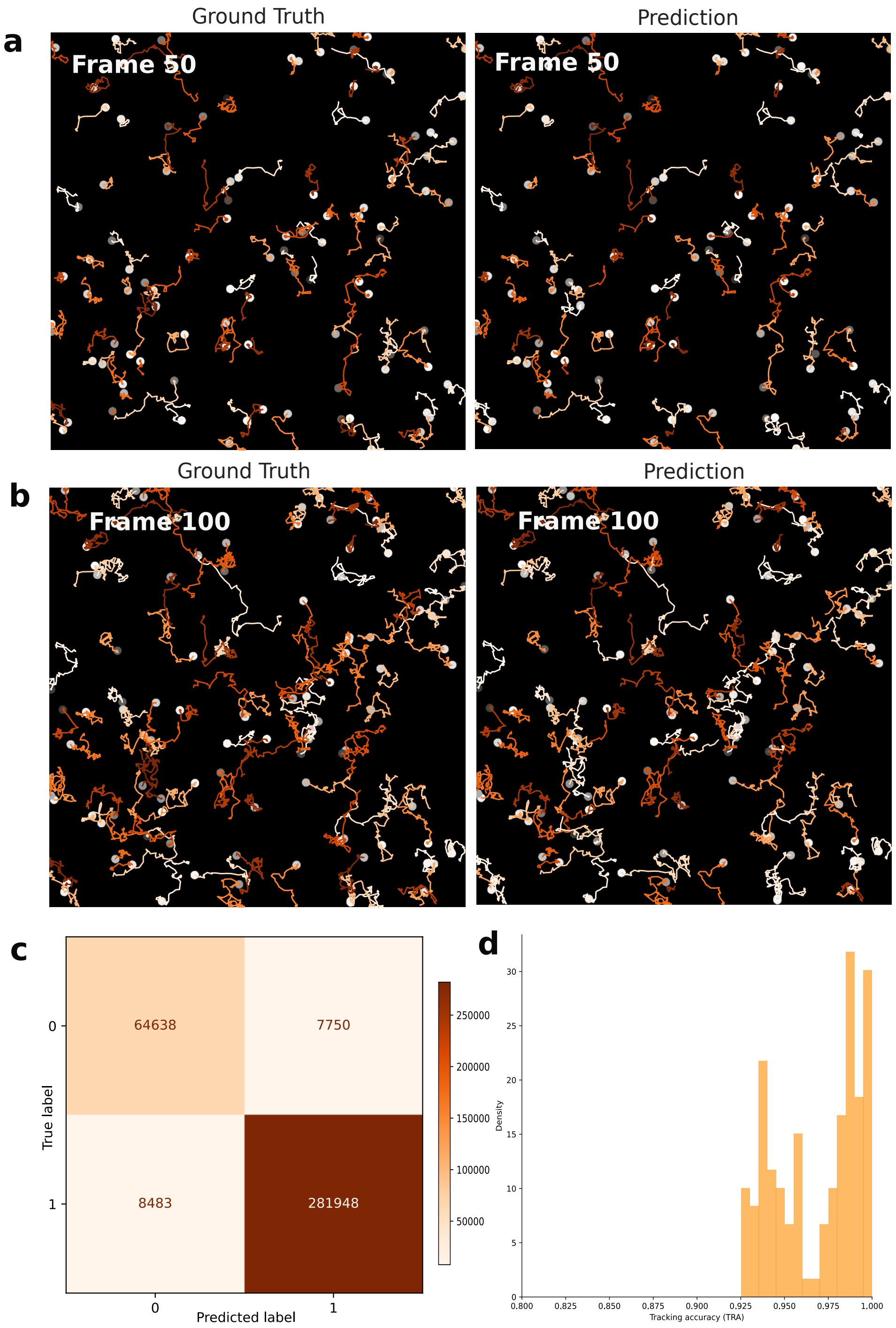


**Figure 3.4: Reconstructed trajectories on synthetic data**. **a-b**, Comparison between reconstructed trajectories (right panels) generated from MAGIK model predictions and the corresponding ground truth, simulated trajectories (left panels). **a**, Example at frame 50; **b**, Example at frame 100. **c**, Example confusion matrix computed from 150 validation samples, used to calculate the Tracking Accuracy (TRA). **d**, Distribution of TRA scores across the validation dataset (150 samples).

$w_{FN} = w_{FP} = 0; w_{ED} = 1, w_{EA} = 1.5$. The total AOGM score for the predicted graph, denoted AOGM-A, is computed as:

$$\text{AOGM-A} = w_{ED} \cdot N_{ED} + w_{EA} \cdot N_{EA} \tag{3.10}$$

where $N_{ED}$ is the number of true edges missing in the prediction, and $N_{EA}$ is the number of false edges predicted. The cost of building the entire ground truth graph from scratch, denoted AOGM-A$_0$, is computed as:

$$\text{AOGM-A}_0 = w_{EA} \cdot N_{GT} \tag{3.11}$$

where $N_{GT}$ is the number of edges in the ground truth graph. The tracking accuracy (TRA) is then defined as:

$$\text{TRA} = 1 - \frac{\min(\text{AOGM-A, AOGM-A}_0)}{\text{AOGM-A}_0} \tag{3.12}$$

This value ranges from 0 to 1, with 1 indicating perfect reconstruction of the ground truth graph.

For the trajectory linking task, the values of $N_{ED}$ and $N_{EA}$ can be considered as false negatives and false positives, derived from the confusion matrices. Figure 3.4 **c** shows an example confusion matrix generated from the validation datasets, providing the key components necessary for computing the TRA metric. The distribution of TRA values is calculated across 150 synthetic validation examples, which are all different from the training data. This distribution is illustrated in Figure 3.4 **d**. Notably, all TRA scores surpass 0.9, underscoring the robustness and high accuracy of our pipeline in reconstructing particle trajectories.

## 3.4 Results on experimental data

In this study, trajectories of vesicle-encapsulated nanoparticles are successfully reconstructed from scattering microscopy images. To evaluate the pipeline's adaptability to diverse biological environments, I investigated four different conditions: ethylene glycol chain-functionalized gold nanoparticles (EG6-AuNPs) and polyethylene glycol-functionalized gold nanoparticles (PEG-AuNPs), each tracked within two different prostate cancer cell lines—LNCaP and PC3. This experimental design allowed me to assess the influence of both nanoparticle surface chemistry and cellular microenvironment on tracking performance in future research.

Figure 3.5 presents representative reconstructed trajectories from each experimental condition. Although no manually labeled ground truth is available for direct comparison, it is evident that most of the reconstructed trajectories are consistent and successfully track the same nanoparticle over time. The robustness of these tracking results across diverse experimental conditions highlights the pipeline's strong potential for broad applicability across various tracking scenarios.

## 3.5 Discussion and further directions

In this chapter, I have presented a robust pipeline for reconstructing trajectories of vesicle-encapsulated nanoparticles from large-scale microscopy videos. This pipeline

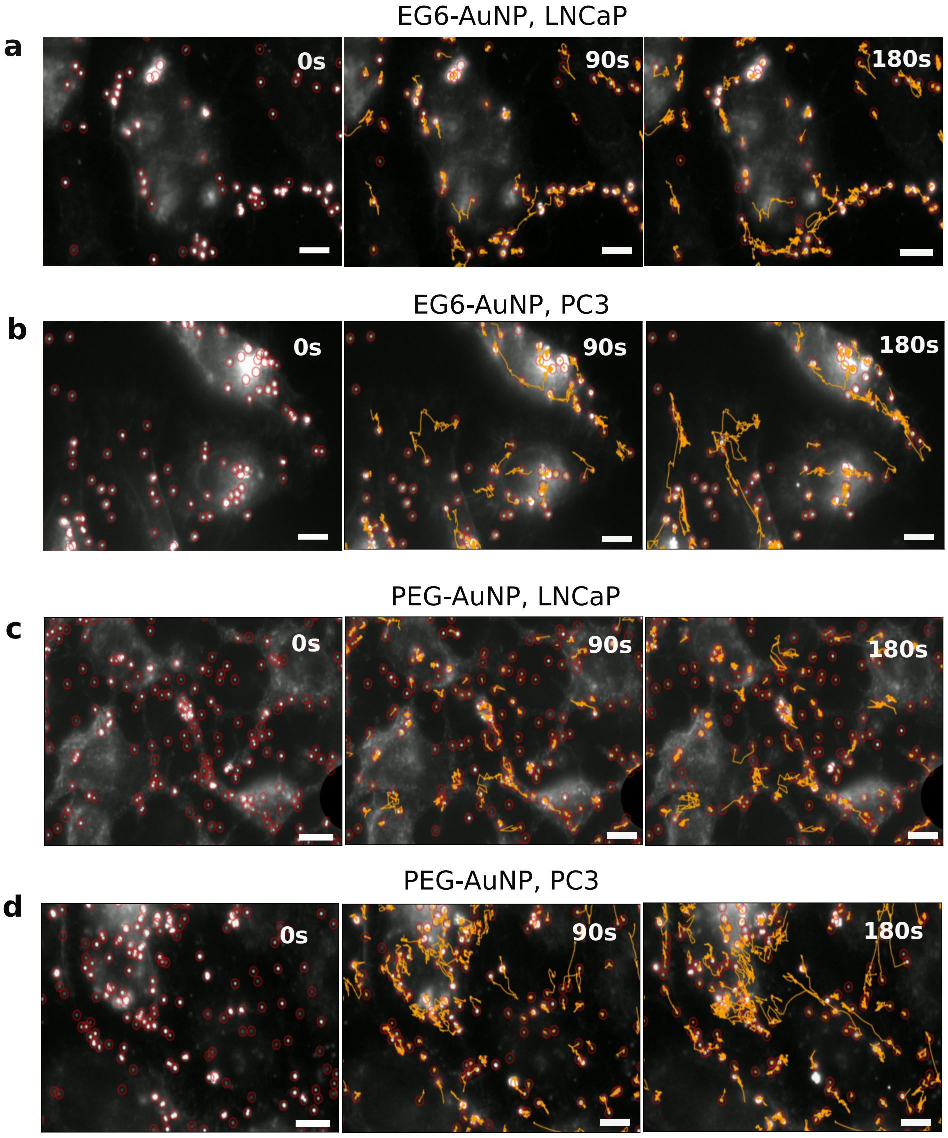


**Figure 3.5: Reconstructed trajectories on experimental videos**. Time evolution of predicted trajectories, shown at three time steps (t = 0, 90, 190 s) under different experimental conditions. Videos were recorded at a frame rate of 1 FPS; scale bar: 2 $\mu$m. **a**, EG6-coated AuNPs within LNCaP cells. **b**, EG6-coated AuNPs within PC3 cells. **c**, PEG-coated AuNPs within LNCaP cells. **d**, PEG-coated AuNPs within PC3 cells.

reduces the memory usage when processing large-scale videos. In addition, another strength of this approach is the use of synthetic data, which provides high flexibility in data generation. By designing suitable synthetic datasets, it becomes possible to closely mimic diverse experimental conditions while avoiding the substantial time and cost associated with manual annotation.

Despite these strengths, several limitations remain. First, complex biological behaviors, such as vesicle fusion or division, may be difficult to simulate, potentially leading to gaps between synthetic and real data. Second, even with temporal segmentation, the size of each graph built on segmented sequences is constrained by the number of detected objects per frame. Future work could explore a more efficient design of fingerprinting graph blocks that balances model precision with scalability.

By addressing these challenges, I anticipate that this pipeline will become a versatile tool for the quantitative analysis of vesicle movements in a wider range of cellular contexts.

# 4

# Characterizing intracellular nanoparticle transport dynamics

In the last chapter, I described a robust pipeline combining the edge-specific MAGIK model and the LodeSTAR model, to reconstruct vesicle-encapsulated nanoparticle trajectories from large-scale scattering microscopy videos. I demonstrated its performance on synthetic datasets which simulated vesicle-encapsulated nanoparticle movements within cells. Finally, I tested it on experimental datasets collected under four distinct conditions.

As accurate trajectories are obtained, it is crucial to understand the underlying transport mechanisms represented by these trajectories for biomarker extraction. For example, one potential task is to identify segments of "active transport" within each trajectory, representing a directed motion state facilitated by molecular motors along cytoskeletal structures. These active transport phases represent critical components of intracellular drug delivery and hold potential as biomarkers for cancer cell characterization and classification.

This chapter will focus on the deep learning methods for inferring diffusion properties of reconstructed trajectories. In this work, I mainly focus on one task: inference of the anomalous diffusion exponent ($\alpha$), which can be used for deciding nanoparticle motion types. Unlike traditional approaches that rely on manually designed features, such as the random forest-based methods developed by G.Muñoz-Gil et al. [30], I apply graph neural networks to the inference of diffusion properties.

## 4.1 Inference of nanoparticle diffusion exponent

As described in previous chapters, the definition of different nanoparticle motion types is based on the diffusion exponent values ($\alpha$). To achieve this, I will first estimate diffusion exponents for each reconstructed trajectory point. While convolutional neural networks (CNNs) and long short-term memory (LSTM) struggle with complex spatiotemporal patterns, I develop a modified graph neural network based on the MAGIK model [9] to analyze spatial and temporal information at the same time.

To estimate diffusion exponents from trajectories, I design a modified node-specific MAGIK model that formulates the diffusion exponents inference task as a node regression problem. This approach enables the model to predict the diffusion exponent value for each detected nanoparticle in consecutive frames. A critical preprocessing step before applying this model is the graph construction process. Here,

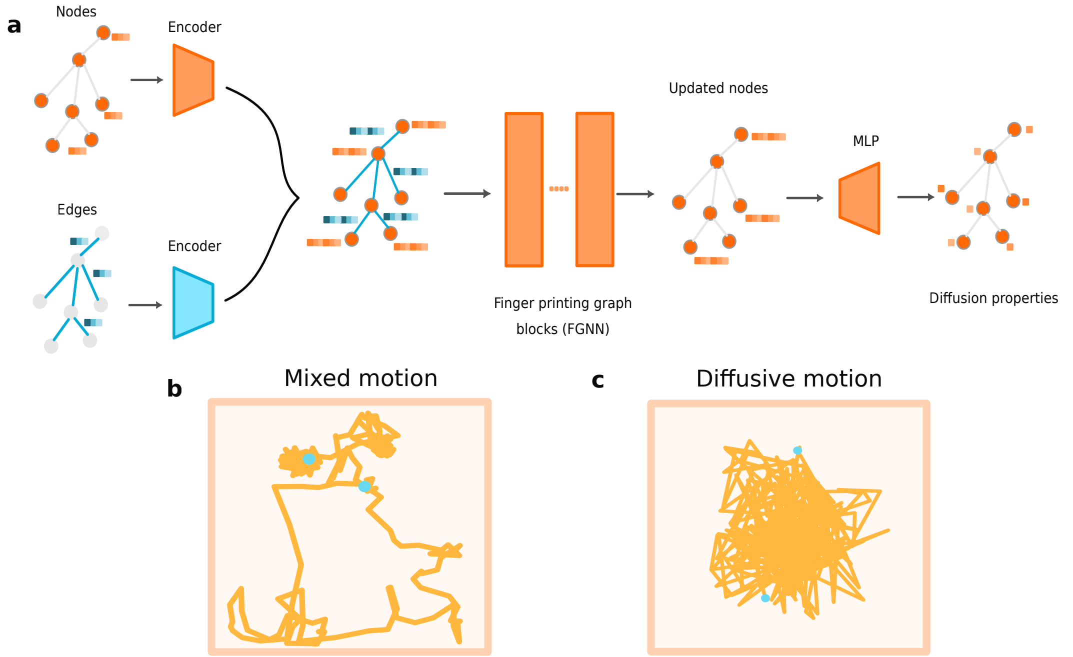


**Figure 4.1: Training process of the modified node-specific MAGIK model.** **a**, The internal architecture of the modified node-specific MAGIK model, highlighting the flow of node attribute updates. **b-c**, Examples of simulated trajectories used in the training datasets: **b**, mixed motion combining diffusive and directed behaviors; **c**, purely diffusive motion.

I generate input graphs where each node represents a nanoparticle at a specific time point, while edges capture the temporal relationships between consecutive detections of the same nanoparticle.

Specifically, these input graphs are constructed using synthetic trajectories that serve as ground truth data. Each node is assigned a diffusion exponent $\alpha \in [0, 2]$ from ground truth trajectories, where $\alpha < 1$ represents the subdiffusion motion type, and $\alpha \geq 1.5$ represents the directed (strong superdiffusive) motion type. This node labeling transforms the trajectory reconstruction challenge into a supervised learning task. Representative example trajectories are shown in Figure 4.1**b-c**.

During training, the modified MAGIK model keeps edge attributes fixed while updating node attributes that encode corresponding diffusion exponents. The model optimizes its parameters by minimizing the mean absolute errors (MAE) between the predicted diffusion exponents $\hat{\alpha}$ and the corresponding ground truth labels $\alpha$ [9]. This optimization strategy enables the model to accurately predict diffusion exponent values, combining sufficient information.

For all trajectory diffusion properties inference tasks, we trained the network for 65 epochs using 1024 training samples representing diverse motion types, processed in batches of 10 samples per iteration. Each sample consists of a single trajectory with 400 to 800 frames.

### 4.1.1 Foundations of modified node-specific MAGIK

This modified node-specific MAGIK model remains the core architectural framework of the original MAGIK model, as displayed in Figure 3.2 **a**. Similar to the edge-specific MAGIK model shown in Figure 3.2 **a**, this architecture contains three major steps [9]:

1. **Feature encoding:** Node attributes $\mathbf{v}_i$ and edge attributes $\mathbf{e}_{ij}$ are passed through separate multilayer perceptrons (MLP) to generate initial embeddings.
2. **Graph fingerprinting:** The encoded graph is passed through a stack of fingerprinting graph blocks. These blocks implement message-passing neural networks to propagate relevant information across the graph structure.
3. **Node decoding:** The updated node embeddings are processed through a decoder MLP to predict the diffusion exponent $\alpha_i$ for each node.

Specifically, the original fingerprinting graph blocks consist of the following key computational steps:

1. Updates of edge attributes are computed as:

$$\mathbf{e}'_{ij} = \mathrm{MLP}\left(\left[\mathbf{v}'_i, \mathbf{v}'_j, \mathbf{e}_{ij}\right]\right) \tag{4.1}$$

   where $[\cdot, \cdot, \cdot]$ represents concatenation of feature vectors.

2. The local representation for node $i$ is the combination of node attributes and neighbors edge attributes, which is computed as:

$$\mathbf{h}_i = \mathbf{W_H}\left[\mathbf{v}'_i, \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{e}'_{ij}\right] \tag{4.2}$$

   where $\mathbf{W_H}$ is a linear projection matrix, $\mathcal{N}_i$ represents the neighborhood of node $i$ and $w_{ij}$ represents weights (importance) of each neighbor node. The definition of $w_{ij}$ is:

$$w_{ij} = \exp\left(-\left(\frac{d_{ij}^2}{2\sigma^2}\right)^{\beta}\right) \tag{4.3}$$

   where $d_{ij}$ is the Euclidean distance between centroids of nodes $i$ and $j$; $\sigma$ and $\beta$ are learnable parameters controlling the spatial weights.

3. Combined local representations and node embeddings (global representations) are passed through gated self-attention to update node attributes:

$$\mathbf{V}''^{(z)} = \mathrm{attn}^{(z)}(\mathbf{H}) \tag{4.4}$$

   where $z = 1, \ldots, Z$, with $Z$ representing the number of attention heads; $\mathbf{H} = [\mathbf{U}; \tilde{\mathbf{H}}]$ combines learnable node embeddings $\mathbf{U}$ with the local representation matrix $\tilde{\mathbf{H}} = [h_i]$ .

However, the original fingerprinting graph blocks exhibit significant computational inefficiency when applied to the diffusion exponent inference task. This inefficiency comes from its emphasis on spatial relationships through Euclidean distance-based weighting (Equation 4.3), which is not optimal for characterizing diffusion exponents. The theoretical definition of anomalous diffusion reveals that diffusion exponent ($\alpha$) inference fundamentally depends on the power-law relationship between mean squared displacement (MSD) and time lag $\tau$, as described in Equation 2.1. This relationship reveals the critical importance of temporal correlations in diffusion exponents analysis, rather than purely spatial relationships. To address this limitation and optimize model performance for diffusion characterization, I replace the original weighting mechanism (Equation 4.3) with temporal relationships between nodes:

$$w'_{ij} = \exp\left(-\left(\frac{\Delta t_{ij}^2}{2\sigma_t^2}\right)^{\beta_t}\right) \tag{4.5}$$

where $\Delta t_{ij}$ is the frame lag between nodes $\mathbf{v}'_i$ and $\mathbf{v}'_j$; $\sigma_t$ and $\beta_t$ are learnable parameters governing temporal attention, which are set to the same values as $\sigma$ and $\beta$ in the original model.

### 4.1.2 Generating synthetic data for the training

Similar to our approach in the previous chapter, we generated fully labeled synthetic datasets for model training using the `andi-datasets` Python package [13], rather than relying on manually labeled experimental data. However, the dataset design for diffusion property inference differs significantly from that used for trajectory linking tasks. For diffusion exponent inference, each simulation scenario contains only one trajectory, in contrast to the multiple-trajectory scenarios required for the linking task. This single trajectory approach fits the nature of diffusion analysis, where the anomalous exponent $\alpha$ characterizes the scaling relationship between mean-squared displacements (MSD) and time lags within a specific trajectory. Since this relationship is determined by the individual particle's motion, it remains independent of other trajectories in the system. Therefore, training the model on individual trajectories enables more efficient learning of the relations between trajectory features and corresponding diffusion properties.

To simulate the vesicle-encapsulated nanoparticle movements within the intracellular environment accurately, each single trajectory in the scenario is sampled from two motion regimes: (1) **Diffusive motion:** trajectories with constant diffusion exponents, simulating vesicles that are either spatially confined (e.g., near the nucleus) or diffusing freely without active transport by motor proteins; (2) **Mixed motion:** trajectories with temporally changing diffusion exponents, simulating vesicles distant from the nucleus that exhibit both diffusive and directed behaviors, the latter occurring when vesicles attach to cytoskeletal filaments, resulting in directed transport [22, 28]. These scenarios will serve as ground truth for edge labels (Figure 4.1 **b-c**).

### 4.1.3 Results on simulated data

Before applying the trained model to experimental microscopy videos, I conduct a comprehensive validation of its performance using synthetic datasets. These validation datasets are generated following the same procedure as the training datasets but with different parameter configurations. Specifically, each validation scenario contains a single trajectory that is 500 to 800 frames long.

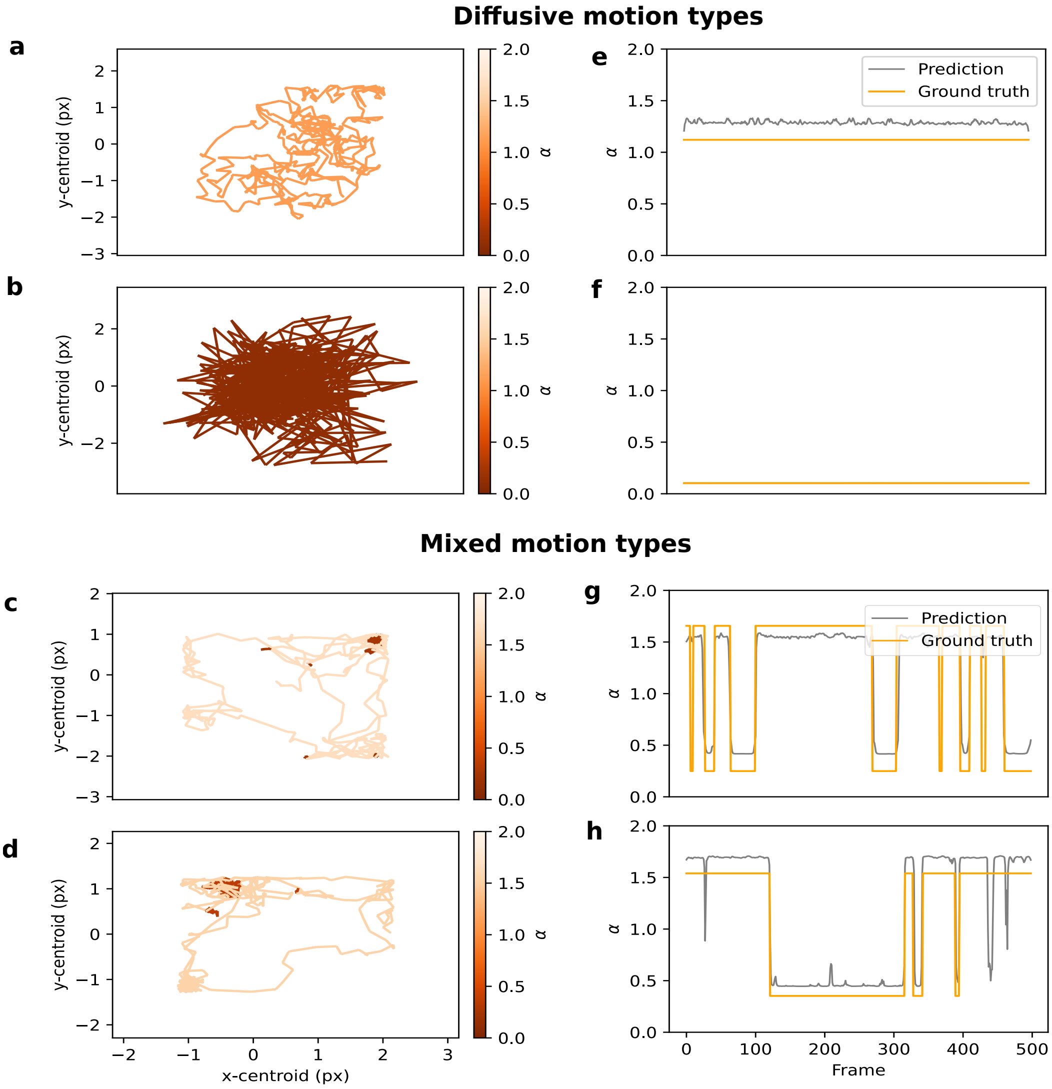


**Figure 4.2: Predicted diffusion exponents on synthetic trajectories. a–d**, Examples of individual 2D simulated trajectories undergoing changes of diffusion exponents. The color of the segments represents the value of the diffusion exponent. Specifically, **a–b** display purely diffusive motion, while **c–d** show mixed motion. **e–h**, Time traces of diffusion exponent values along the same trajectories. The ground-truth value used in the simulations is shown by the orange curve. The predictions obtained by MAGIK at the single-node level are shown in grey. **e–f** correspond to purely diffusive motion; **g–h** correspond to mixed motion.

The key results from validation datasets are estimations of diffusion exponent values, which are shown in Figure 4.2 **.** Figure 4.2 **a–d** displays two distinct trajectories from validation datasets containing purely diffusive motions, alongside the corresponding estimations of diffusion exponents. Figure 4.2 **e-h** presents two distinct trajectories from validation datasets containing mixed diffusive motions, alongside the corresponding estimations of diffusion exponents.

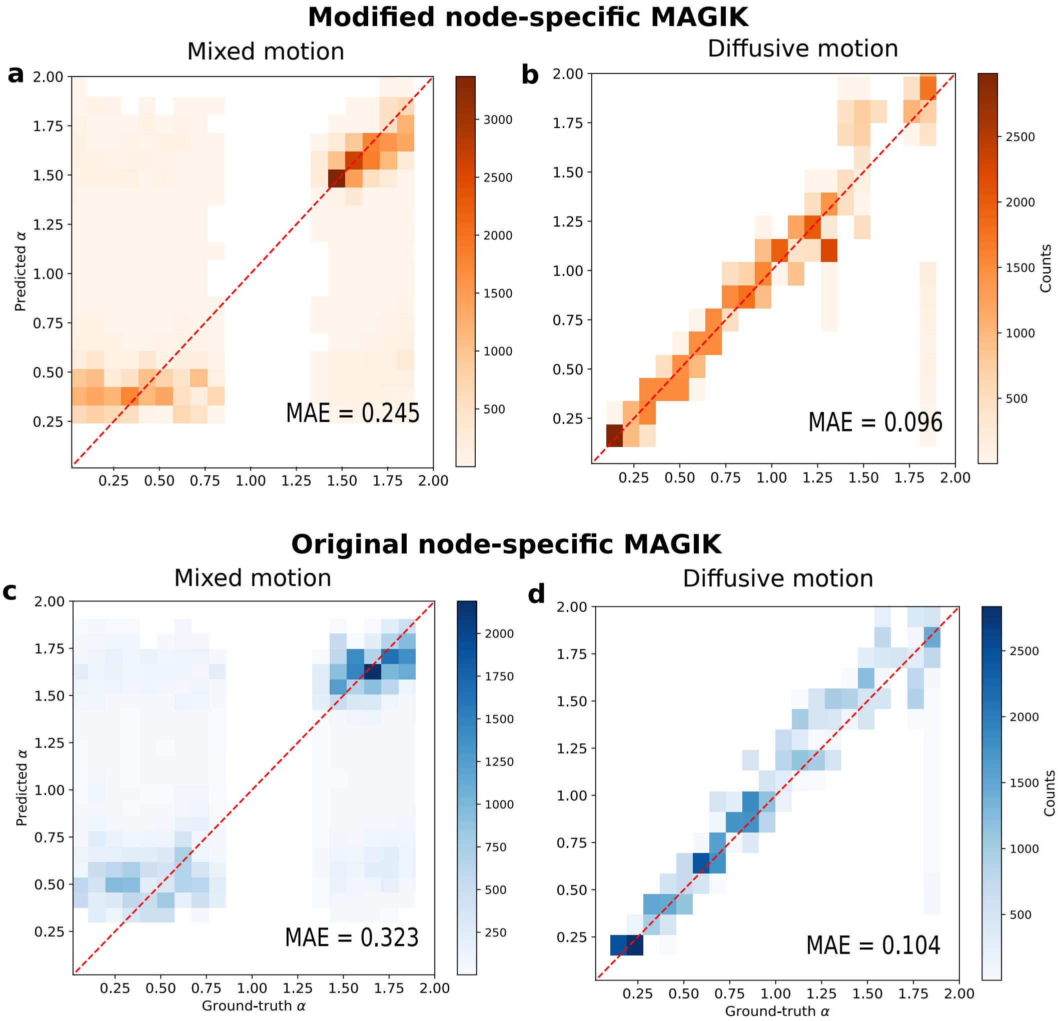


**Figure 4.3: Performance comparison between the modified and original node-specific MAGIK models on synthetic data. a-b**, Probability distribution of predicted diffusion exponents using the modified node-specific MAGIK model versus ground-truth for simulated trajectories with mixed motion (**a**) and with purely diffusive motion (**b**). **c-d**, Probability distribution of predicted diffusion exponents using the original node-specific MAGIK model versus ground-truth for simulated trajectories with mixed motion (**c**) and with purely diffusive motion (**d**).

To evaluate the model performance on synthetic datasets, I employed the mean absolute error (MAE) metric, which measures the precision of our estimations. This metric is mathematically defined as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{\alpha}_i - \alpha_i| \tag{4.6}$$

where $N$ represents the total number of frames across all validation trajectories, $\hat{\alpha}_i$ denotes the predicted diffusion exponent for frame $i$, and $\alpha_i$ represents the corresponding ground truth.

For my comparative analysis, I selected 200 individual trajectories each from validation datasets containing purely diffusive motions and mixed motions, respectively. We then compute the mean absolute errors (MAE) for predictions of diffusion exponents derived from both our modified node-specific MAGIK model and the original node-specific MAGIK model. Figure 4.3 **a-b** presents the probability distribution of predicted diffusion exponents ($\alpha$) from our modified node-specific MAGIK model plotted against ground truth values. Correspondingly, Figure 4.3 **c-d** presents the probability distribution of predicted diffusion exponents from the original node-specific MAGIK model against ground truth values.

Analysis of Figure 4.3 and Figure 4.2 leads to the following significant conclusions:

1. The modified node-specific MAGIK model shows higher accuracy in predicting diffusion exponents for trajectories with mixed motion types (MAE = 0.245) compared to the original model (MAE = 0.323). This substantial improvement validates our model modifications, which not only accelerated the training process but also enhanced prediction accuracy.
2. For trajectories with purely diffusive motion, the modified node-specific MAGIK model achieves similar accuracy (MAE = 0.096) to the original model (MAE = 0.104). This result is expected, as predicting constant diffusion exponents is a straightforward task that does not necessarily benefit from our modifications.
3. Within the performance of the modified node-specific MAGIK model, we observe particularly strong performance in predicting strongly superdiffusive motions ($\alpha \geq 1.5$) compared to other motion regimes. This enhanced accuracy for strong superdiffusive trajectories close to our research objectives, as this thesis particularly emphases on directed motion ($\alpha \geq 1.5$) of vesicle-encapsulated nanoparticles as potential diagnostic biomarkers.

### 4.1.4 Results on experimental data

After validating the predictive performance of our modified node-specific MAGIK model on synthetic datasets, I apply it to experimental microscopy videos capturing nanoparticle dynamics under four distinct experimental conditions. These conditions, same as those described in the previous chapter, consisted of: ethylene glycol chain-coated gold nanoparticles (EG6-AuNPs) and polyethylene glycol-coated gold nanoparticles (PEG-AuNPs), each tracked within two prostate cancer cell lines—LNCaP and PC3. All microscopy videos are recorded at 1 frame per second (FPS) with a consistent scale of 1 pixel = 0.1 $\mu$m. For each condition, I first reconstruct particle trajectories using the pipeline introduced in the previous chapter, then calculate inferred diffusion exponent values at each point along each reconstructed trajectory.

Figure 4.4 presents detailed results from two selected conditions: EG6-AuNPs within PC3 and LNCaP prostate cancer cell lines. Figure 4.4 **a-c** displays three distinct reconstructed trajectories of EG6-AuNPs within PC3 cells, with their corresponding temporal changes of predicted diffusion exponents and instantaneous velocities. Similarly, Figure 4.4 **d-f** shows three representative trajectories of EG6-AuNPs within LNCaP cells with their respective diffusion exponent and velocity

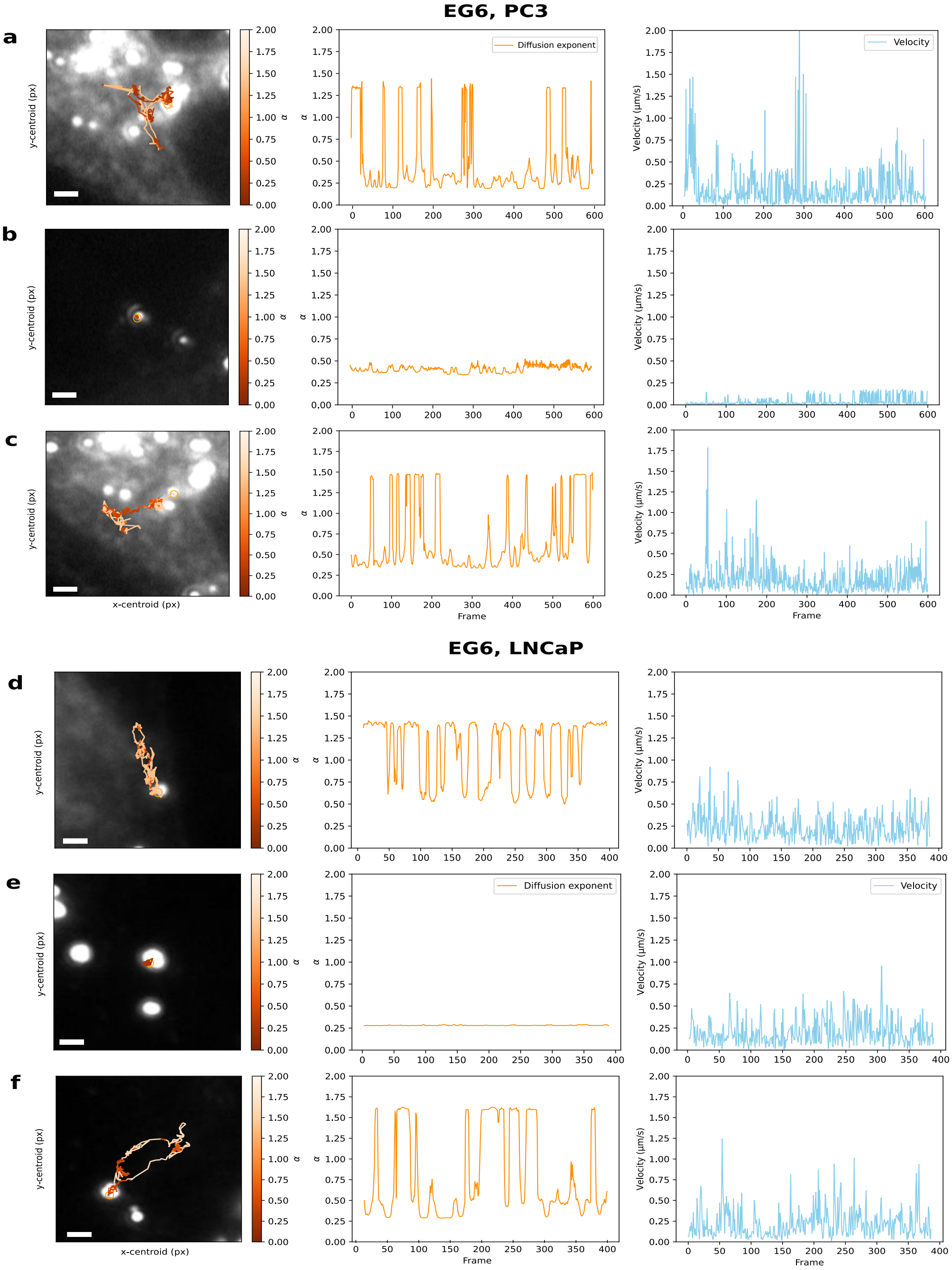


**Figure 4.4: Predicted diffusion exponents on reconstructed trajectories from experimental videos.** Left panels show 2D trajectories in experimental videos and the color of their segments represents the diffusion exponent values; middle panels present time traces of diffusion exponent values along the same trajectories (orange trace); right panels present time traces of nanoparticle velocities along the same trajectories (blue trace). **a-c**, Three example trajectories of EG6-coated AuNPs within PC3 cells; **d-f**, Three example trajectories of EG6-coated AuNPs moving within LNCaP cells.

profiles.

The predicted diffusion exponents present two primary patterns: (1) constant values indicating purely diffusive behavior, and (2) time-varying values indicating transitions between purely diffusive behavior and directed motion. This demonstrates the modified node-specific MAGIK model's capability to identify both pure diffusive motion types and mixed motion regimes within a single trajectory. Notably, trajectory segments with higher diffusion exponents exhibit more directed motion (fewer angular changes), aligning with theoretical expectations of anomalous diffusion processes.

### 4.1.5 Validation of inferred diffusion exponents on experimental datasets

While ground truth diffusion exponent values for experimental trajectories are unavailable, I validated our predictions by examining their consistency with established biophysical observations. Previous research by S Mogre et al. [2] has demonstrated that directed motion typically exhibits higher motility. Figure 4.5 presents this validation analysis. Figure 4.5 **a-b** shows the distribution of predicted diffusion exponents and their correlation with instantaneous velocities for EG6-AuNPs in PC3 cells. Figure 4.5 **c-d** presents the same analysis for EG6-AuNPs in LNCaP cells. These distributions revealed the following findings:

- The distributions of diffusion exponents under both conditions exhibit bimodal patterns with distinct peaks at approximately $\alpha \approx 0.3$ and $\alpha \approx 1.5$, corresponding to subdiffusive and superdiffusive (directed) motion, respectively. This bimodality aligns with the biological expectation of distinct transport mechanisms within the cellular environment.
- The correlation between predicted diffusion exponents and particle velocities varies by cell type. For EG6-AuNPs in PC3 cells, we observed a moderate positive correlation (correlation coefficient = 0.412). In contrast, EG6-AuNPs in LNCaP cells exhibited a weaker correlation (correlation coefficient = 0.107, see Table A.1 in Appendix A.2). This is likely due to the smaller size of LNCaP cells, which restricts the spatial range available for particle movement, thereby reducing the differences between fast and slow motility.

These validation analyses indicate that our model's predictions align with established biophysical principles governing intracellular particle transport, providing confidence in the biological relevance of our computational approach.

## 4.2 Extracting differentially expressed dynamic indices from nanoparticle trajectories

As discussed in previous chapters, the ultimate aim of this research is to identify dynamic indices from reconstructed trajectories that show differential expression across cancer cell lines, potentially serving as biomarkers. Based on the diffusion exponent predictions presented in the last section, I focus specifically on velocity measurements during periods of directed motion as the dynamic indices.

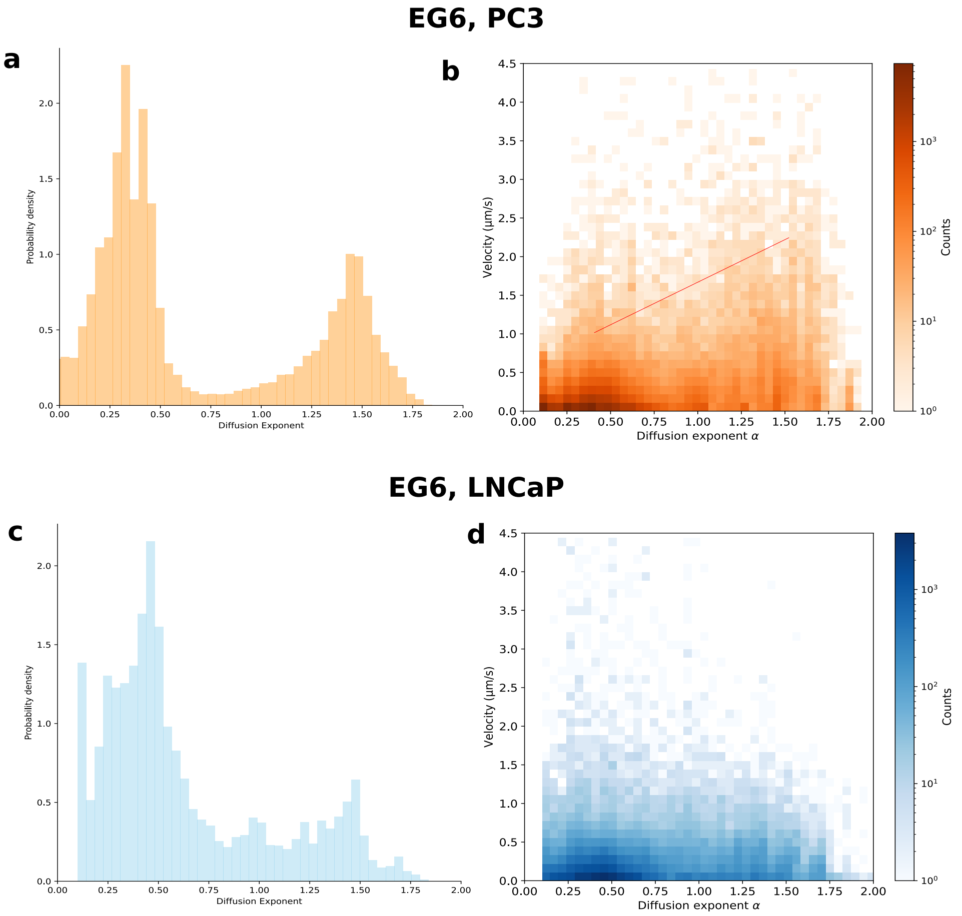


**Figure 4.5: Validation of inferred diffusion exponents on experimental data. a-b**, Results for experimental videos recording EG6-coated AuNPs within PC3 cells. **a**, Distribution of point-wise diffusion exponent values across all reconstructed trajectories under this condition. **b**, Joint probability distribution of point-wise diffusion exponents versus corresponding particle velocities. **c-d**, Results for experimental videos capturing EG6-coated AuNPs within LNCaP cells. **c**, Distribution of point-wise diffusion exponent values across all reconstructed trajectories under this condition. **d**, Joint probability distribution of point-wise diffusion exponents versus corresponding particle velocities.

To compute velocities during the directed motion, I first make a quantitative definition of **directed motion** based on the predicted diffusion exponent values. Based on established anomalous diffusion theory in previous chapters, trajectory segments with diffusion exponent values exceeding $\alpha \geq 1.5$ are classified as directed motion. This threshold corresponds to strongly superdiffusive behavior characteristic of active transport processes within cells. For both EG6-AuNPs and PEG-AuNPs, I compared the velocity distributions during periods of directed motion between PC3 and LNCaP prostate cancer cell lines. These distributions, presented in Figure 4.6, are derived from the analysis of four independent experimental videos for each experimental condition to ensure statistical robustness.

Figure 4.6 **a** illustrates the differential directed motility distributions of vesicle-

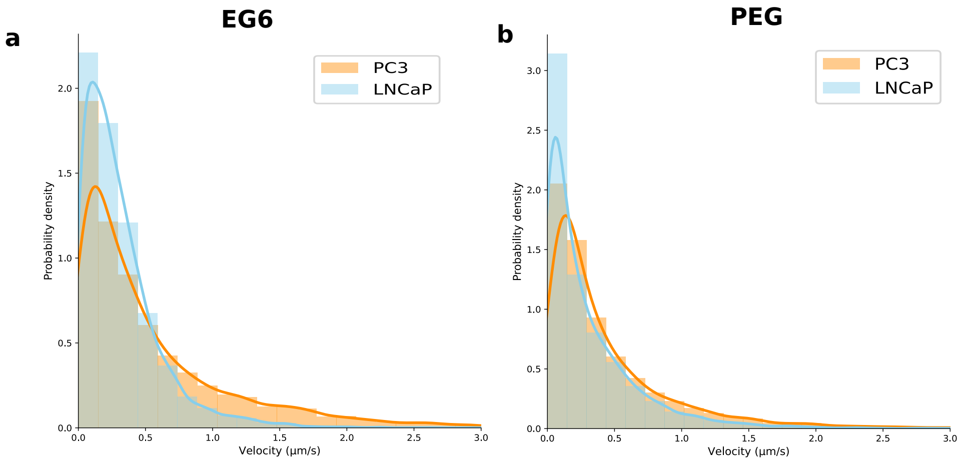


**Figure 4.6: Identified dynamic indices from reconstructed trajectories that show differential expression across cancer cell lines.** **a**, Distribution comparison of nanoparticle velocities during directed motion under two conditions: EG6-coated AuNPs in PC3 cells (orange) versus EG6-coated AuNPs in LNCaP cells (blue). **b**, Distribution comparison of nanoparticle velocities during directed motion under two conditions: PEG-coated AuNPs in PC3 cells (orange) versus PEG-coated AuNPs in LNCaP cells (blue).

encapsulated EG6-AuNPs between PC3 and LNCaP cells. Figure 4.6 **b** presents the same comparison for PEG-AuNPs. Statistical analysis of these distributions revealed following significant findings:

- Both nanoparticle types exhibited consistently higher directed motility velocities in PC3 cells compared to LNCaP cells. This cell-line-dependent velocity differential was statistically significant, suggesting fundamental differences in the cytoskeletal transport mechanisms or intracellular trafficking pathways between these two prostate cancer cell lines.
- The magnitude of this cell-line velocity differential was substantially greater for EG6-AuNPs compared to PEG-AuNPs. This indicates that the surface chemistry of the nanoparticles could influence their interaction with cellular transport machinery.

These findings suggest that EG6-AuNPs represent a more sensitive probe design for capturing distinct intracellular transport indices between prostate cancer cell lines. This observation highlights the importance of nanoparticle surface chemistry optimization when developing dynamic biomarker approaches for cancer cell phenotyping.

## 4.3 Discussion and further directions

This chapter demonstrated the application of my modified node-specific MAGIK model to predict diffusion exponent values in nanoparticle trajectories and extract differentially expressed directed motion velocities as potential biomarkers. The model performed effectively on both synthetic and experimental data. Its results from prostate cancer cell lines align with established biological understanding of intracellular transport mechanisms. In addition, the directed motion velocity analysis reveals significant differences between PC3 and LNCaP cells, with EG6-AuNPs showing greater sensitivity to cell line differences than PEG-AuNPs. This suggests that nanoparticle surface chemistry plays a crucial role in capturing distinct cellular phenotypes.

Despite these strengths, several limitations remain. First, the analysis only focuses on diffusion exponents, neglecting other important parameters such as spatially distributed diffusion coefficients. Second, beyond velocity measurements during directed motion, additional dynamic indices that might serve as effective biomarkers remain unexplored. Third, the current analysis includes nanoparticles that adhere to the glass surface and exhibit static behavior in the videos. Since these particles are not the aimed particles for the analysis, their inclusion introduces potential errors into the final results.

By expanding the range of diffusion parameters and dynamic indices analyzed, this approach holds significant potential for developing novel functional biomarkers based on intracellular transport characteristics of polymer-coated nanoparticles.

# 5

# Conclusion

In the previous chapters, I have explored how to reconstruct trajectories of vesicle-encapsulated gold nanoparticles (AuNPs) from large-scale scattering microscopy videos and how to extract dynamic indices from these trajectories as potential biomarkers to distinguish between different prostate cancer cell lines. Here, I summarize the methods and results discussed in these chapters and explore potential paths for future expansion.

The ultimate aim of this thesis is to identify novel biomarkers capable of differentiating prostate cancer cell lines by analyzing the intracellular behavior of polymer-coated AuNPs. Since these nanoparticles are internalized by vesicles within cells, they provide a high precision to track vesicular dynamics. Additionally, prior research has shown that vesicle trafficking machinery is altered in different prostate cancer cell lines [8], which narrows the search for biomarkers to those related to vesicle dynamics. In this thesis, I introduced a single-particle tracking (SPT) framework to analyze nanoparticle movements and extract diffusion-related properties as candidate biomarkers. The SPT framework comprises two main tasks: trajectory reconstruction, as discussed in Chapter 3, and diffusion property extraction, as discussed in Chapter 4.

The experimental data analyzed in this thesis are obtained by capturing the motion of polymer-coated AuNPs using a scattering microscope. As shown in the microscopy images (Figure 3.1 **a**), the observed bright spots represent vesicle-encapsulated AuNPs. All analyzed videos shared the same frame rate (1 FPS) and scale (1 pixel = 0.1 $\mu$m).

The first major step is reconstructing the nanoparticle trajectories. Given the large size of the experimental datasets (over 500 frames and more than 100 detected nanoparticles per frame), applying graph neural networks directly is computationally limited. To address this problem, Chapter 3 introduced a robust pipeline that segments input videos into smaller sequences, processes each sequence via the MAGIK model [9] separately, merges the predicted graphs, and constructs trajectories from the integrated graph (Figure 3.3 **a-d**). This approach reduces GPU demands while maintaining high accuracy. Furthermore, rather than relying on manually labeled data, the MAGIK model in the pipeline is trained on simulated trajectories designed to mimic the motion types observed in the experimental data. This design decreases the need for manual annotation in biological research. When applied to experimental datasets, this pipeline successfully reconstructs trajectories (Figure 3.5) , revealing multiple particle trajectories within each video.

Once the trajectories are reconstructed, Chapter 4 focuses on the second step, predicting diffusion properties from these trajectories. Given that prior studies on vesicle motion emphasize the importance of diffusion exponents [2, 12], this thesis concentrated on predicting point-wise diffusion exponent values for each trajectory. To achieve this, I modified the MAGIK model [9]. This enhanced model demonstrates improved accuracy—specifically, a lower Mean Absolute Error (MAE) for high diffusion exponent values estimation on simulated validation datasets (Figure 4.3). Subsequently, I apply this pre-trained model to experimental data to infer the diffusion exponent values, revealing a range of motion types that align with expectations (Figure 4.4).

Building on these predictions, I finally explore dynamic indices as potential biomarkers. Motion types, such as directed motion and subdiffusive motion, are classified based on the corresponding diffusion exponent values from prior research [2, 22]. One key index analyzed is the velocity of nanoparticles during directed motion, which exhibited distinct distributions across prostate cancer cell lines (Figure 4.6). These findings suggest the promise of this index as a discriminative biomarker.

In conclusion, this thesis presents an integrated approach for reconstructing nanoparticle trajectories and characterizing dynamic indices from trajectories, offering a promising path for identifying dynamic-related biomarkers. Future work may expand this framework by incorporating additional diffusion properties (such as spatially distributed diffusion coefficients) and exploring a broader array of dynamic indices with strong biological relevance. Through these efforts, we may further enhance the potential of nanoparticle-based analyses for biomarker discovery.

# Bibliography

[1] Hongjie Guo, Chen Zhou, Mingming Zheng, Jie Zhang, Honghai Wu, Qiaojun He, Ling Ding, and Bo Yang. Insights into the role of derailed endocytic trafficking pathway in cancer: From the perspective of cancer hallmarks. *Pharmacological Research*, 201:107084, 2024.

[2] Saurabh S Mogre, Aidan I Brown, and Elena F Koslover. Getting around the cell: physical transport in the intracellular world. *Physical Biology*, 17(6):061003, oct 2020.

[3] Rebecca L Siegel, Kimberly D Miller, Nikita Sandeep Wagle, and Ahmedin Jemal. Cancer statistics, 2023. *CA: a cancer journal for clinicians*, 73(1):17–48, 2023.

[4] J. Ferlay, M. Colombet, I. Soerjomataram, D. M. Parkin, M. Piñeros, A. Znaor, and F. Bray. Cancer statistics for the year 2020: An overview. *International Journal of Cancer*, 2021.

[5] Young-Eun Choi, Ju-Won Kwak, and Joon Won Park. Nanotechnology for early cancer detection. *Sensors*, 10(1):428–455, 2010.

[6] Seyed Mohammadali Dadfar, Karolin Roemhild, Natascha Ingrid Drude, Saskia von Stillfried, Ruth Knuchel, Fabian Kiessling, and Twan Lammers. Iron oxide nanoparticles: Diagnostic, therapeutic and theranostic applications. *Advanced Drug Delivery Reviews*, 138:302–325, 2019.

[7] Alaaldin M. Alkilany and Catherine J. Murphy. Toxicity and cellular uptake of gold nanoparticles: what we have learned so far? *Journal of Nanoparticle Research*, 12(7):2313–2333, 7 2010.

[8] B. D. Nturubika, C. M. Guardia, D. C. Gershlick, et al. Altered expression of vesicular trafficking machinery in prostate cancer affects lysosomal dynamics and provides insight into the underlying biology and disease progression. *British Journal of Cancer*, 131:1263–1278, 2024.

[9] J. Pineda, B. Midtvedt, H. Bachimanchi, et al. Geometric deep learning reveals the spatiotemporal features of microscopic motion. *Nat. Mach. Intell.*, 5:71–82, 2023.

[10] F. Simon, L.E. Weiss, and S. van Teeffelen. A guide to single-particle tracking. *Nat Rev Methods Primers*, 4:66, Sept. 2024.

[11] W. Cai, Y. Hu, X. Qu, et al. Machine learning analysis of anomalous diffusion. *Eur. Phys. J. Plus*, 140:183, Mar. 2025.

[12] M. Liu, Q. Li, L. Liang, et al. Real-time visualization of clustering and intracellular transport of gold nanoparticles by correlative imaging. *Nature Communications*, 8, 2017.

[13] G. Muñoz-Gil, G. Volpe, M.A. Garcia-March, et al. Objective comparison of methods to decode anomalous diffusion. *Nat Commun*, 12:6253, Oct. 2021.

[14] Javier Duarte and Jean-Roch Vlimant. *Graph Neural Networks for Particle Tracking and Reconstruction*, chapter 12, pages 387–436. 2022.

[15] B. Midtvedt, J. Pineda, F. Skärberg, et al. Single-shot self-supervised object detection in microscopy. *Nat Commun*, 13:7492, Dec. 2022.

[16] K. Löffler, T. Scherr, and R. Mikut. A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction. *PLoS ONE*, 16:e0249257, 2021.

[17] H. Verdier et al. Learning physical properties of anomalous random walks using graph neural networks. *Journal of Physics A: Mathematical and Theoretical*, 54:234001, 2021.

[18] G. Muñoz-Gil et al. Andi datasets: Anomalous diffusion challenge 2020. Zenodo, v1.0, 2021.

[19] Gernot Guigas, Claudia Kalla, and Matthias Weiss. Probing the nanoscale viscoelasticity of intracellular fluids in living cells. *Biophysical Journal*, 93(1):316–323, July 2007.

[20] N. Monnier, Z. Barry, H. Park, et al. Inferring transient particle transport dynamics in live cells. *Nat Methods*, 12:838–840, Aug. 2015.

[21] Paolo Pierobon, Sonia Achouri, Sylvain Courty, Alexander R. Dunn, James A. Spudich, Maxime Dahan, and Giovanni Cappello. Velocity, processivity, and individual steps of single myosin V molecules in live cells. *Biophys. J.*, 96(10):4268–4275, May. 2009.

[22] Avi Caspi, Rony Granek, and Michael Elbaum. Diffusion and directed motion in cellular transport. *Phys. Rev. E*, 66:011916, Jul 2002.

[23] B. Requena, S. Masó-Orriols, J. Bertran, M. Lewenstein, C. Manzo, and G. Muñoz Gil. Inferring pointwise diffusion properties of single trajectories with deep learning. *Biophys. J.*, 122(22):4360–4369, 2023.

[24] A. Argun, G. Volpe, and S. Bo. Classification, inference and segmentation of anomalous diffusion with recurrent neural networks. *Journal of Physics A: Mathematical and Theoretical*, 54(29):294003, 2021.

[25] H. Verdier et al. Learning physical properties of anomalous random walks using graph neural networks. *J. Phys. A: Math. Theor.*, 54:234001, 2021.

[26] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

[27] L. Mill, D. Wolff, N. Gerrits, et al. Synthetic image rendering solves annotation problem in deep learning nanoparticle segmentation. *Small Methods*, 5(7):2100223, 2021.

[28] D. Posey, P. Blaisdell-Pijuan, S. K. Knoll, et al. Small-scale displacement fluctuations of vesicles in fibroblasts. *Sci Rep*, 8:13294, 2018.

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

[30] G. Muñoz-Gil, M. A. Garcia-March, C. Manzo, J. D. Martín-Guerrero, and M. Lewenstein. Single trajectory characterization via machine learning. *New Journal of Physics*, 22(1):013010, 2020.

[31] Mattias Sjöberg, Mokhtar Mapar, Antonius Armanious, Vladimir P. Zhdanov, Björn Agnarsson, and Fredrik Höök. Time-resolved and label-free evanescent light-scattering microscopy for mass quantification of protein binding to single lipid vesicles. *Nano Letters*, 21(11):4622–4628, 2021.

# A

# Supplementary Information

## A.1 Computation of node attributes for input graph construction

For each fixed-size image patch containing a detected nanoparticle, three node attributes are extracted: centroid position $(x, y)$, raw intensity $I_r$, and background intensity $I_b$. To calculate the intensity values, the image patch is first processed with a Gaussian filter to estimate the background signal. Two derived patches are then created: one containing only the background and another obtained by subtracting the background from the original image patch.

The raw intensity $I_r$ is computed as the mean of the integrated intensities in the background-subtracted patch, representing the signal of nanoparticles. The background intensity $I_b$ is calculated as the mean intensity in the background-only patch, reflecting local background levels. This method is described by Sj"oberg et al. [31].

## A.2 Pearson correlation coefficients computed from Figure 4.5

Pearson correlation coefficients are computed to quantify the linear relationship between the point-wise diffusion exponent values $(\alpha)$ and the corresponding instantaneous velocity $(v)$ of nanoparticles.

The instantaneous velocity at each point is calculated as:

$$v_i = \frac{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|}{\Delta t}, \tag{A.1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are the 2D positions at time points $i$ and $i+1$, and $\Delta t$ is the time interval between frames.

The Pearson correlation coefficient $r$ between two variables $\alpha$ (diffusion exponent) and $v$ (velocity) is computed as:

$$r = \frac{\sum_{i=1}^{n} (\alpha_i - \bar{\alpha})(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n} (\alpha_i - \bar{\alpha})^2} \sqrt{\sum_{i=1}^{n} (v_i - \bar{v})^2}}, \tag{A.2}$$

where $\bar{\alpha}$ and $\bar{v}$ are the mean values of $\alpha$ and $v$, respectively, and $n$ is the number of data points. The significance of the correlation is assessed using the associated p-value. Table A.1 summarizes the computed Pearson correlation coefficients and their corresponding p-values for the two experimental conditions shown in Figure 4.5:

**Table A.1:** Pearson correlation coefficients between point-wise diffusion exponent values and velocities for EG6-AuNPs in different prostate cancer cell lines.

| **Condition** | **Correlation Coefficient** ($r$) | **P-value** |
|---|---|---|
| EG6-AuNP, PC3 | 0.412 | $1 \times 10^{-4}$ |
| EG6-AuNP, LNCaP | 0.107 | $1 \times 10^{-4}$ |

**DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY