Graph Neural Network-Based Real-Time 3D Tracking for Micro-Agent Control

Yuxin Jin, Franco Pinan Basualdo, Antonio Marino, Yongfeng Mei, Paolo Robuffo Giordano, *Senior Member, IEEE*, Claudio Pacchierotti, *Senior Member, IEEE* and Sarthak Misra, *Senior Member, IEEE*

***Abstract*— Micro-surgical robotic systems are gaining prominence in minimally invasive surgery within the medical field. However, accurately tracking the position of the moving agents at the micro-scale remains a significant challenge, particularly for multi-agent systems operating in cluttered and unknown environments. Traditional image analysis methods can falter when confronted with issues such as mutual and obstacle occlusion, especially in dynamic and unstructured scenarios. In order to address this issue, this study introduces a graph-based multi-agent 3D tracking algorithm for a micro-agent control system. This algorithm integrates image information with the control inputs used to navigate the micro agents. We combine the power of Convolutional Neural Networks and Graph Neural Networks to effectively extract features from image sources, and combine them with historical data and control inputs. The primary novelty of this algorithm is its ability to make predictions when the target is occluded in the 2D detection results. The proposed system achieved a tracking error of 0.15 mm, outperforming standard model-based tracking techniques.**

I. Introduction

Untethered micro-agents have showcased significant effectiveness across various domains, including minimally-invasive surgery, constructing microstructures, precise handling or positioning of objects, and approaching challenging locations to accomplish designated tasks [1], [2]. The accurate control and maneuverability of the micro-agents in confined spaces offer advantages in safeguarding surrounding tissues from potential damage. The efficacy of the micro-agent systems heavily depends on precise tracking mechanisms to provide micro-agent positions, enabling controllers to define appropriate control inputs. The absence of accurate position feedback poses challenges to closed-loop controllers, leading to potential errors and complicating system control. However, achieving real-time and robust computation of micro-agent locations remains a significant challenge.

The transition from standard image-based tracking to machine learning techniques in the field of object tracking has been driven by the limitations of traditional methods and the advancements offered by neural network-based techniques. Traditional image processing dominated the field of object detection previously. Such approaches relied on distinct image attributes, like color or shape, to anticipate object locations in a 2D plane [3], [4]. Wang *et al.* introduced methods involving the comparison of consecutive images to discern moving objects, albeit contingent upon a static background [5]. Similarly, the work proposed by Cheung and Kamath recurrently updates a single background model based on input frames [6]. However, these algorithms struggled with dynamic backgrounds and changes in lighting and were limited in handling dynamic scenarios effectively [7].

In recent years, Convolutional Neural Networks (CNNs) have emerged as a dominant paradigm in object detection tasks due to their ability to learn features from raw data [8] automatically. Various CNN architectures such as Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector) have been proposed to address different aspects of object detection, including accuracy, speed, and robustness [9], [10], [11]. Object detection serves as an important precursor to object tracking, providing initial bounding boxes around objects of interest, where the association step between different frames during tracking is needed. Deep Simple Online and Realtime Tracking (SORT) improves the object tracking performance by adding a Kalman filter-based motion model [12]. Despite the remarkable progress achieved by CNN-based methods in object detection, challenges remain in handling occlusions, scale variations, and cluttered scenes, prompting ongoing research into novel network architectures and training strategies. The emergence of Graph-based Neural Networks (GNNs) has garnered significant attention, capitalizing on their ability to discern interrelationships among nodes and edges [13]. Gori *et al.* were pioneers in outlining the concept of GNNs in

This work is supported by the European Union's Horizon Europe Research and Innovation Programme under Grant Agreement 101070066 (project: RĔGO). This project has also received funding from the European Research Council through the European Union's Horizon2020 Research and Innovation Programme under Grant 866494 (project: MAESTRO).

Yuxin Jin and Franco Pinan Basualdo are with the Surgical Robotics Laboratory, Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands y.jin@utwente.nl; f.n.pinanbasualdo@utwente.nl.

Antonio Marino is with Université de Rennes, CNRS, Inria, IRISA – Rennes, France. antonio.marino@irisa.fr

Yongfeng Mei is with Department of Materials Science & State Key Laboratory of Molecular Engineering of Polymers, Fudan University, Shanghai 200433, P. R. China, and with International Institute of Intelligent Nanorobots and Nanosystems, Fudan University, Shanghai 200433, P. R. China. He is also with Shanghai Frontiers Science Research Base of Intelligent Optoelectronics and Perception, Fudan University, Shanghai 200433, P. R. China. yfm@fudan.edu.cn

Claudio Pacchierotti and Paolo Robuffo Giordano are with CNRS, Université de Rennes, Inria, IRISA-Rennes, France. claudio.pacchierotti@irisa.fr; prg@irisa.fr.

Sarthak Misra is with the Surgical Robotics Laboratory, Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands and also with the Surgical Robotics Laboratory, Department of Biomaterials and Biomedical Technology Engineering, University of Groningen and University Medical Centre Groningen, 9713 GZ Groningen, The Netherlands s.misra@utwente.nl

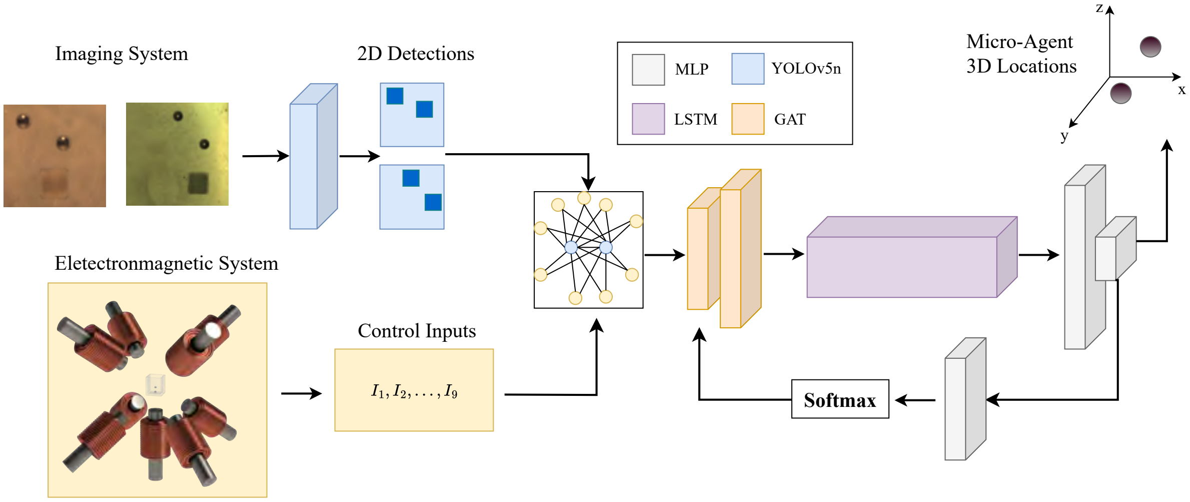

Fig. 1: Overview of the proposed pipeline: YOLO Version 5 (YOLOv5) is initially employed to predict 2D detections from input images. Control inputs $(I_1, I_2, ..., I_9)$ are currents to the electromagnetic system are obtained. We combine detection results (blue) with the control inputs (yellow) to form a graph. The network architecture of the proposed graph neural network consists of Graph Attention Neural Networks (GAT), Long Short Term Memory (LSTM), and Multi-Layer Perceptron (MLP). The output of the GAT, encoded node information, is passed to LSTM which can make the temporal prediction. Afterward, a multi-layer perception will map the temporal-encoded information to a 3D location. Meanwhile, a Softmax function updates the attention matrix by using the final outputs.

2005, a concept further developed by Scarselli *et al.* [14], [15]. Additionally, Zhang *et al.* have initiated exploration into the use of GNNs for object tracking, leveraging their capacity to discern interactions among targets for precise predictions [16]. However, current methodologies predominantly focus on spatial-temporal relationships, overlooking application- and system-specific relationships.

This paper introduces a novel GNN-based real-time object tracking algorithm, designed specifically for wireless magnetic multi-agent systems at the microscale. Our approach utilizes dual images and control inputs of the micro-agent system as the inputting data to forecast the 3D locations of micro-agents. By integrating a CNN with an attention-based GNN, we establish a novel tracking network proficient in 3D tracking based on disparate 2D image viewpoints [17]. While CNN scrutinizes visual features from input images, the GNN predicts target locations leveraging CNN outcomes, historical data, and control inputs. This fusion harnesses visual information alongside physical inputs, augmenting prediction accuracy. The GNN assumes a pivotal role during occlusion events and unexpected scenarios, comprehending physics models and computing frame features to infer reliable predictions. This paper uses micro-sized particles steered in 3D space by a set of nine electromagnetic coils. For this reason, we use the currents driving the electromagnetic coils as the control input for our tracking algorithm. Our contributions are as follows:

- We propose a novel integrated framework with CNN and GNN, to learn 2D to 3D mapping at the micro-scale.
- We add the micro-agent control input (e.g, the current input to the robot system) to increase the robustness of the predictions.
- We prove the effectiveness of the proposed method by evaluating the 3D error rates concerning the ground truth. The results are compared against other GNN-based algorithms.

II. Methods

Our objective is to predict the 3D location of micro-agents given a sequence of previous graph snapshots containing 2D location and control inputs. The data can be characterized as a dynamic graph $(\mathcal{G} = (\mathcal{V}, \mathcal{E}, t))$ where $\mathcal{V}$ contains the node features $(\mathcal{X})$ for each of N nodes at timestamp (t). $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set which demonstrates the connection between each node. We can assign different weights to the edge to define the relationship between nodes. The task at timestamp (t) is, given the most recent timestamps $\mathcal{V}_{t-L}, ..., \mathcal{V}_{t-1}$ with length L, to predict the current agent location $(R_t \in \Re^N \times 3)$.

In order to track the 3D location of the micro-agents, the proposed system utilizes a combination of CNN, and GNN with Long Short Time Memory (LSTM) layers to exploit spatial-temporal relationships in the data and predict the 3D coordinates in the world frame as the output. The architecture of the proposed networks is summarized in Figure 1. Our approach considers edge features in the Graph Attention Neural Networks (GAT) layers, passing them to the LSTM layer to extract temporal information. Finally, a softmax function updates the weights of the graph edges based on attention scores.

2D Object Tracking: Two images, taken from the robot workspace's top view and side view, serve as input to the 2D object detector. YOLO version 5 (YOLOv5), an improved version of YOLO, is a single-stage object detection model with CSPDarknet53 as the backbone [18], [10]. In order to balance accuracy and inferencing speed, we chose the model YOLOv5n, a smaller version among the YOLOv5 family, to process the image and locate the bounding box in the 2D frames. The output from the object detector is given by

$$\mathcal{O}_b = [\mathcal{B}_0, ..., \mathcal{B}_i]^T, \tag{1}$$

$$\mathcal{B}_i = [b_x^i, b_y^i, b_w^i, b_h^i], \tag{2}$$

where $\mathcal{O}_b$ is the set of bounding boxes, and $\mathcal{B}_i$ represents the i-the bounding box with coordinates (b_x^i as center (x) pixel location, b_y^i as center (y) pixel location, b_w^i as width, and b_h^i as height).

Graph Structure: Among all the control inputs of the electromagnetic system, we choose the current input to the system as it directly drives the micro-agents. We combine control information from the robot system and image frames to form the graph, as shown in Figure 1. It consists of 11 nodes with two different node classes: two 2D location nodes and nine robot control nodes. Blue nodes contain the location values of the 2D bounding boxes calculated by the object detector in the top and side view images. The remaining yellow nodes represent the control input to the microrobot system, i.e., in our case, there are nine current values used to actuate the nine coils respectively.

Graph Attention Neural Networks: Graph Attention Neural Networks (GATs) use self-attention to assign importance to each node's neighbors and combine their features [17], [19]. This allows GATs to handle complex and large graphs without depending on the graph structure. The graph convolution operation from layer (l) to layer $(l+1)$ is calculated by:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} A_t W^l h_j^{(l)}\right), \tag{3}$$

$$\text{ReLU}(x) = \max(0, x), \tag{4}$$

where $\mathcal{N}(i)$ is the set of its one-hop neighbors, $c_{ij} = \sqrt{|N(i)|}\sqrt{|N(j)|}$ is a normalization constant, σ is the Rectified Linear Unit (ReLU) activation function, A_t is the attention score calculated from the final outputs and W^l is a shared weight matrix for node-wise feature transformation.

LSTM: Employing an LSTM layer, a sophisticated variant of recurrent neural networks, proves instrumental in the overarching architecture, seamlessly integrating into the model to meticulously capture and discern the intricate temporal relationships inherent in the dynamic input graphs [20].

Model Output: The output of the model is 3D locations ($R_t \in \Re^{N\times 3}$) of N agents at time t.

Softmax Function: After obtaining the agent locations from Multi-layer Perceptrons (MLP), we apply another MLP layer to convert the $R_t \in \Re^{N\times 3}$ to $A_t \in \Re^{N\times 10}$. Subsequently, attention scores are computed using the softmax function, which calculates the probability p_i for each element $a_i^t \in A_t$ as follows:

$$p_i = \frac{e^{a_i^t}}{\sum_{j=1}^{N} e^{a_j^t}}, \tag{5}$$

where e is the base of the natural logarithm (Euler's number), and the denominator is the sum of the exponentiated values of all elements in the input vector. These attention scores are then multiplied with the corresponding node features to emphasize their significance for subsequent calculations.

III. Experimental Results

A. Experimental Setup

In order to test the performance of the proposed method, we use BatMag, an electromagnetic system to perform the experiments with two moving agents in Figure 2 [21]. The BatMag electromagnetic system allows independent 3D control of pairs of identical and non-identical spherical micro-agents. The motion of the micro-agents is induced by magnetic fields and gradients generated by nine electromagnetic coils, positioned to satisfy specific workspace accessibility and force exertion constraints.

The details of the experimental setup are listed below:

- Electromagnetic system: The system comprises nine Vacoflux-core coils, configured with all coils positioned 30 mm from a shared center — eight at the cube's corners and the ninth at the bottom as shown in the computer-aided design (CAD) drawing in Figure 2. Third-party servo drives (Elmo Motion Control, Petach-Tikva, Israel) regulate all electromagnetic coils.
- Imaging system: The imaging configuration consists of two Grasshopper 3 cameras (Teledyne FLIR LLC, USA) capturing side and overhead images. Operating at a

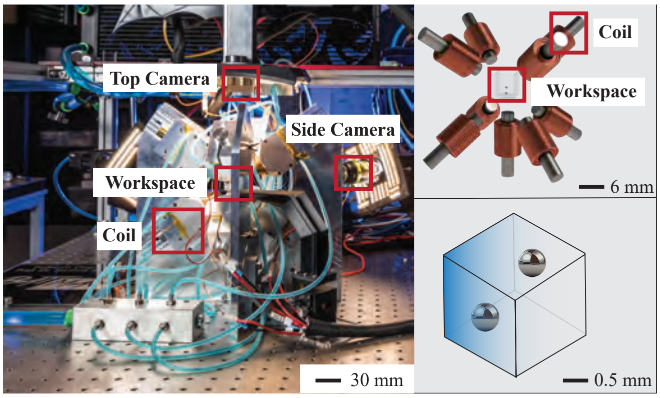

Fig. 2: Electromagnetic system: BatMag consists of nine coils for actuating the micro-agents. On the left side, there is an image of BatMag with two cameras installed on the top and side of the system. In the top-right corner is a 3D computer-aided design (CAD) drawing of BatMag with the workspace in the middle. In the bottom-right corner, there is a drawing showing two micro-agents moving within the workspace. The currents driving the nine electromagnets and the 2D images retrieved by the cameras are the inputs of our 3D tracking system.

resolution of 2048 × 2048 pixels, equivalent to a pixel size of approximately 10 μm, these cameras record at a rate of 10 Hz.

- Agents: The magnetic agents employed in the experiments are 0.5 mm diameter AISI 420C stainless steel spheres.
- Test environment: All experiments were conducted in M1000 silicon oil with a viscosity of 1 Pas.

To compare with the previous study by Basualdo *et al.*, we replicate the experiment outlined in [22]. The current experiment involves actuating two steel spheres using the magnetic field within the workspace to predefined targets. The input data for the 3D tracking algorithm consists of visual information captured by the imaging system from both the top and side views, along with the current inputs to the micro-agent system.

B. Data Collection and Training

The quality of the dataset significantly influences the efficacy of a neural network model. The data collection process can be categorized into two main components: the collection of the image datasets and the currents driving the electromagnets.

For the image collection, the previously mentioned imaging system is utilized to record three ten-minute videos of the movement of two spheres within the electromagnetic system. In order to ensure the randomness of the recorded trajectories, multiple arbitrary targets located across the workspace are chosen for the micro-agent to approach during the data generation phase. These videos are subsequently segmented into 18000 images because we record the video at 10 frames per second. Afterward, the images are allocated into training, validation, and testing datasets at ratios of 0.6, 0.3, and 0.1, respectively.

The collection of micro-agent-related information is registered by the BatMag system during the micro-agents' movement. The current input to the electromagnetic system can be extracted upon retrieving the necessary data. This information is then combined with the visual features derived from the collected images to construct the graphs. Concurrently, by using the locations of 2D bounding boxes labeled manually, we calculate the 3D location using the triangulation method [23]. This information serves as the ground truth for training the neural networks.

In order to bolster the system's robustness, we train under occlusion or misdetection scenarios. Occlusion occurs when the two micro-agents overlap within a single view, causing partial visibility of the micro-agents in the images. Misdetection arises when the object detector identifies fewer objects in the image frame than are present. We randomly remove some existing detections to simulate misdetection scenarios during experimentation. Additionally, we deliberately induce occlusion ten times during data collection. This methodology empowers neural networks to acquire predictive capabilities when camera views are obscured or detection results are flawed. The training process is elaborated in Algorithm 1. During the training, we separate the graph dataset ($\mathcal{D}$) into a continuous sequence of length (l), then we apply a moving window (W) with length (l_w) to graph sequences to assign previous graphs together used for prediction. During each epoch, we first build graphs (g_b) using the visual and control data in the batched dataset. Then, for each window, we compute the feature using GNN for each graph inside the window. Then, we stack the outputs from the GNN to LSTM to get a temporal prediction. Finally, a multilayer perception (MLP) layer is applied to compute the 3D location. Meanwhile, the output is passed through a Softmax function to get the probability of updating the attention matrix (A).

Algorithm 1 Training Algorithm

```
Require: dataset $\mathcal{D}$ composed by data sequences of length $l$ of $N$ agents motions
  initialize dataset $\hat{\mathcal{D}} = \{\}$
  for sequence in $\mathcal{D}$ do                                  ▷ Preparation steps
      initialize batch Tensor $b = \{\}$
      for $i = 0 \dots (l - l_w)$ do
          extract window $W$ of length $l_w < l$
          Append window to the batch tensor $b \leftarrow b \cup \{W\}$
      end for
      Append batch to the dataset $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}} \cup \{b\}$
  end for
  for $i = 1 \dots$ epochs do                                       ▷ Training steps
      for $b$ in $\hat{\mathcal{D}}$ do
          Extract visual features over the images in $b$
          Build the graphs structures $g_b$ over the batch and windows
          Initialize attention $A = [1]^{N \times N}$
          for $W$ in $b$ and $g$ in $g_b$ do
              compute features $h \leftarrow GNN(g, A)$ for each element in the window
              $h_w \leftarrow LSTM(W, h), \quad h_w \in \mathbb{R}^{N \times F}$
              $\hat{p} \leftarrow MLP(h_w), \quad \hat{p} \in \mathbb{R}^{N \times 3}$
              Update Attention $A \leftarrow Softmax(MLP(\hat{p}))$
          end for
          Compute MSE loss $L(p, \hat{p})$
          Update model weights based on $L$
      end for
  end for
```

C. Experiment and Comparisons

We assess the efficacy of the proposed 3D tracking method under various circumstances, including occlusion and misdetection. Further, we provide an ablation study to validate our design choice by comparing it with the variations of the proposed neural network architecture.

1) 3D tracking performances: In our evaluation, we assess the effectiveness of the proposed algorithm across 60 sequences of length 100 featuring two spheres navigating a workspace without occlusion. As detailed in Figure 4(a), the resulting tracking error, corresponding to the Euclidean distance between predicted and ground truth values, varies in the range $[0.04 - 0.28]$ mm, with an av-

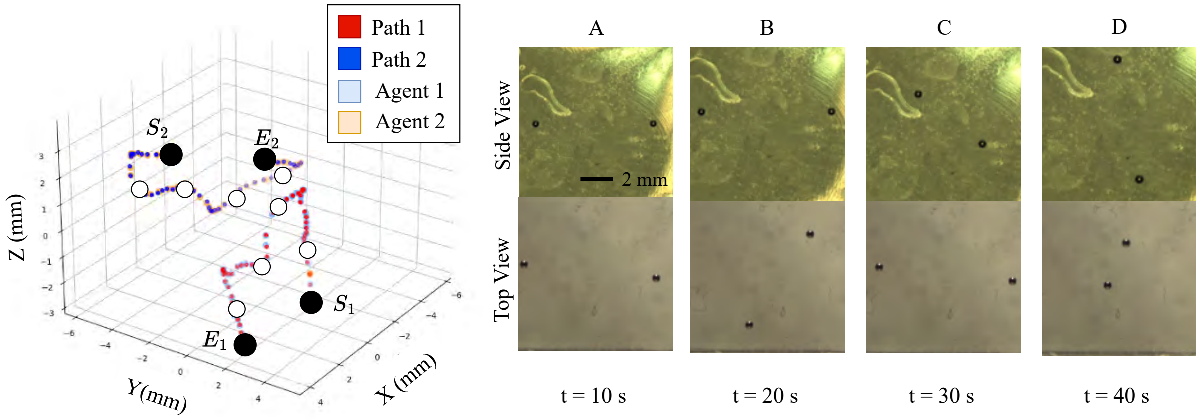

Fig. 3: 3D plot on the left indicates the 3D trajectory of the two moving micro-agents from $t = 0$ s to $t = 60$ s. The black circles mark the start and end points, and the white circles are four intermittent points along the trajectory. The red dots indicate the trajectory of agent 1 while S_1 and E_1 are the starting and ending points, respectively. Blue dots indicate the trajectory of agent 2 while S_2 and E_2 are the starting and ending points, respectively. The orange and light blue dots are the predicted trajectories by the proposed neural network model of agent 1 and agent 2 respectively. On the right side of the figure, we provide four image frames captured at four different timesteps.

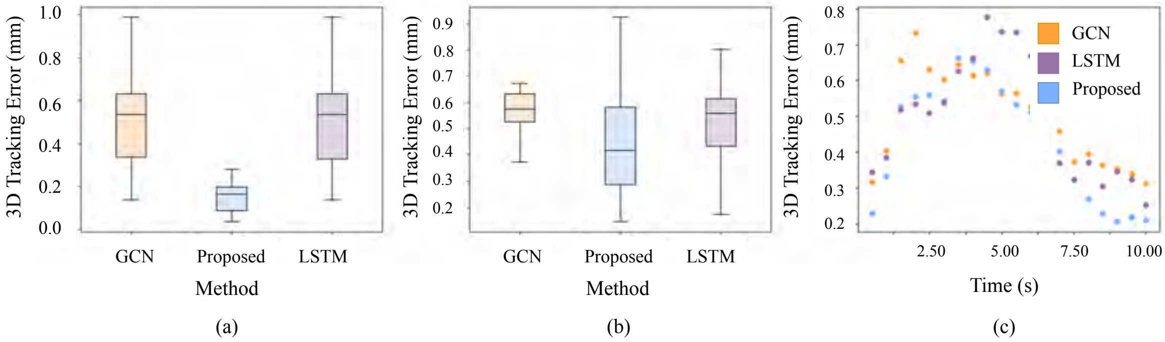

Fig. 4: 3D tracking performance of proposed method against GCN- and LSTM-only method: (a) 3D tracking error when there is no misdetection, (b) 3D tracking error when 20% of 2D detection frames are missing (c) 3D tracking error when we stop providing 2D detection results at $t = 0$ s and resume at $t = 4$ s.

erage of 0.15 mm. In Figure 3, we depict a representative trajectory of two moving agents from $t = 0$ s to $t = 60$ s. The red and blue dots denote the actual measurements and the orange and light blue dots indicate the predicted values from the proposed neural network model. The initial coordinates are $(-0.54, 2.31, -2.75)$ mm and $(1.71, -2.99, 2.81)$ mm for micro-agent 1 and micro-agent 2 respectively, with ending points at $(0.82, 4.52, -3.56)$ mm and $(-3.36, -4.40, 1.59)$ mm. On the right side of Figure 3, we provide four image frames captured at $t = 10$ s, 20 s, 30 s, and 40 s along the trajectory from the top and side views of the imaging system.

2) 3D Localization with Missing Detection: To further evaluate the robustness of the proposed GNN model, we aim to showcase its performance in scenarios where 2D detections are absent. Various factors such as errors in the 2D detector's output, communication failures between the imaging system and the main computer, or accidental obstruction of the camera's view can result in the loss of 2D detection. During the experiment, we intentionally removed some of the detection outputs to emulate missing object detections. As depicted in Figure 4(c), the initial 3D tracking error is recorded at 0.23 mm. When the detection results start to diminish at $t = 0$ s, the error only escalates to 0.65 mm when frames are missing for 4s consecutively. Subsequently, the error begins to decline upon recovery of the detection outputs to the neural network at $t = 4$ s. The tracking error evolution where the detection is omitted at the $t = 8$ s. This experiment highlights the model's capability to maintain stable tracking performance even in the absence of 2D detection, showcasing its resilience and adaptability in real-world scenarios characterized by intermittent detection failures.

3) Ablation Study: Meanwhile, we undertake an experimental investigation into variations of the proposed neural network architecture, including models based exclusively on GCN or LSTM components. To ensure methodological consistency, all models undergo training with identical datasets. Subsequently, their performance is evaluated using the same dataset used in the previous experiment by using 60 sequences. The 3D tracking error, with and without misdetection, is illustrated in Figures 4(a) and 4(b) respectively. Comparing the tracking performance between LSTM and GCN-only models against our proposed integrated GAT-LSTM approach unveils a substantial enhancement in accuracy and stability for predicting both temporal and spatial relationships. Through a deliberate omission of 20% of the 2D detection results, our integrated approach yields a tracking error of 0.43 mm, outperforming the 0.57 mm error of the GCN-only model and the 0.54 mm error of the LSTM-only model. This improvement suggests that the synergy between LSTM's ability to capture temporal relationships and important features in time series effectively and GAT's spatial representation in our approach significantly contributes to strengthening the robustness, surpassing the individual contributions of each model in isolation.

IV. Conclusion

In this paper, we have introduced an innovative 3D tracking algorithm that integrates LSTM networks with GNNs to locate magnetic-actuated micro-agents in real-time scenarios. Our proposed method offers significant advancements in providing the control system with precise and robust micro-agent localization, even in the face of unexpected challenges such as misdetection and object occlusion. By leveraging the combined power of CNNs, GNNs, and LSTM networks, our model adeptly captures intricate spatial-temporal patterns inherent in location-control graphs, thus enhancing the overall tracking performance. The low 3D tracking error achieved by our model underscores its accuracy in tracing micro-agents during normal agent movement. Moreover, the ability of estimate micro-agent locations accurately during occlusion events highlights its robustness in handling unexpected situations, thereby further validating its utility in dynamic environments.

Looking forward, we envision extending our algorithm to handle scenarios involving an increased number of objects, thereby addressing more complex tracking challenges in dynamic environments. By scaling our model to accommodate multiple objects simultaneously, we aim to enhance its versatility and applicability across a broader range of real-world scenarios. Additionally, further research efforts will focus on testing our algorithm using control inputs, enabling comprehensive validation of its performance under diverse operational conditions.

References

[1] H. Zhu, B. Xu, Y. Wang, X. Pan, Z. Qu, and Y. Mei, "Self-powered locomotion of a hydrogel water strider," *Science Robotics*, vol. 6, no. 53, p. 7925, 2021.

[2] B. Xu, B. Zhang, L. Wang, G. Huang, and Y. Mei, "Tubular micro/nanomachines: from the basics to recent advances," *Advanced Functional Materials*, vol. 28, no. 25, p. 1705872, 2018.

[3] S. M. Abdullah-Al-Wadud, M. H. Kabir, M. A. A. Dewan, and O. Chae, "An image processing technique for color detection and distinguish patterns with similar color: An aid for color blind people," in *the Proceedings of the International Conference on Computer and Communication Engineering*, 2008, pp. 82–86.

[4] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[5] R. S. Rakibe and B. D. Patil, "Background subtraction algorithm based human motion detection," *International Journal of Scientific and Research Publications*, vol. 3, no. 5, pp. 2250–3153, 2013.

[6] S. C. Sen-Ching and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *the Proceeding of Visual Communications and Image*, vol. 5308. SPIE, 2004, pp. 881–892.

[7] S. R. Balaji and S. Karthikeyan, "A survey on moving object tracking using image processing," in *the Proceedings of 11th International Conference on Intelligent Systems and Control*, 2017, pp. 469–474.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *in the Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *the Proceedings of European Conference on Computer Vision*. Springer, Cham, 2016, pp. 21–37.

[12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *the Proceedings of IEEE International Conference on Image Processing*, 2017, pp. 3645–3649.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[14] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *the Proceedings of the International Joint Conference on Neural Networks*, vol. 2, 2005, pp. 729–734.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[16] Y. Zhang, S. Li, J. Wang, and T. Liu, "Gnntrack: Graph neural networks for object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2810–2823, 2021.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[18] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, and L. Changyu, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] F. Ongaro, S. Pane, S. Scheggi, and S. Misra, "Design of an electromagnetic setup for independent three-dimensional control of pairs of identical and nonidentical microrobots," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 174–183, 2019.

[22] F. N. Piñan Basualdo and S. Misra, "Collaborative magnetic agents for 3d microrobotic grasping," *Advanced Intelligent Systems*, vol. 5, no. 12, p. 2300365, 2023.

[23] I. Vite-Silva, N. Cruz-Cortés, G. Toscano-Pulido, and L. G. de la Fraga, "Optimal triangulation in 3d computer vision using a multi-objective evolutionary algorithm," in *Applications of Evolutionary Computing*. Springer, 2007, pp. 330–339.