Decentralized Multi-Robot Shared Perception for Worker Action Inference in Industrial Facilities

Ali Imran[1], Giovanni Beltrame[2] and David St-Onge[1]

***Abstract*— Many industrial tasks, including machine tending and assembly operations, can significantly enhance their efficiency by leveraging mobile robotic aids. However, a primary challenge that must be fully addressed for the widespread adoption of these deployments is ensuring that robots are aware of the positions and actions of workers. Moreover, predicting human intent is essential for the safe deployment of robots in tasks where humans and robots work closely together.**

Our paper introduces a multi-robot shared perception framework designed to capture spatiotemporal interactions between humans and surrounding objects. This framework utilizes graph neural networks (GNNs) for spatial analysis and draws inspiration from swarm intelligence for inter-robot information sharing. The collected information is then fed into a gated recurrent unit (GRU) for modeling temporal relations. The development of such a system represents a significant advancement toward achieving effective shared perception, thereby paving the way for future research in dynamic, real-world multi-robot deployments. A visual overview of the system can be accessed here.

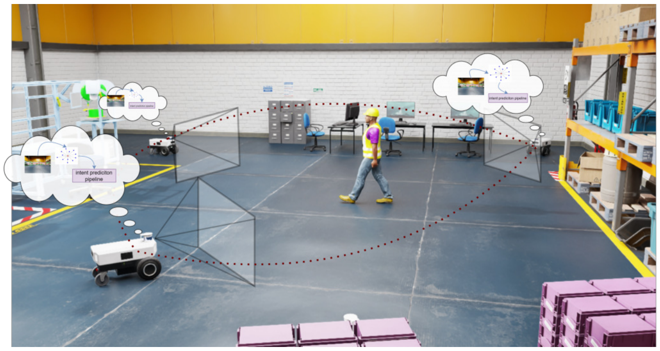
intent prediciton pipeline
intent prediciton pipeline
intent prediciton pipeline

Fig. 1: Deployment of a multirobot system in Isaac Sim for human intent prediction. The robots create a graph representation for spatial understanding, incorporate information from the neighbors, and deploy a GRU for temporal understanding.

I. Introduction

Humans possess a remarkable ability for complex reasoning, relying on multi-level perception to infer the behavior of other entities, particularly other individuals. This ability is robust even in the face of high levels of uncertainty, generating reliable and redundant information models [1]. As mobile robotic systems increasingly integrate into society, it becomes imperative to instill such capabilities within them. This will enable them to gain a deeper understanding of dynamic and unstructured environments, ultimately leading to the development of safer and more trustworthy robotic systems.

Robotic arms have revolutionized industrial manufacturing by executing repetitive and sometimes hazardous tasks with exceptional precision and efficiency, making them indispensable in numerous modern industrial settings. To further enhance their beneficial impact on the industry, the next step is to facilitate efficient and safe collaboration between mobile units and human operators in shared spaces.

Currently, strategies for mobile robotic industrial aids either rely on facilities exclusively designed for robots [2], or single operator-guided systems (AGV). However, greater flexibility is necessary to increase adoption rates. A robotic system comprised of multiple mobile units must be resilient to failures and ensure the safety of workers at all times, thereby establishing trustworthiness for human users. We believe that the initial step in this direction is the development of an efficient shared perception methodology. By creating an efficient perception module, the performance of downstream modules such as decision-making and navigation in cluttered environments can be significantly enhanced.

We have developed an efficient perception system that capitalizes on the multitude of viewpoints in a multi-robot deployment. GNN have recently gained more attention for multirobot path planning [3], collaboration perception [4] and environmental reasoning for navigation [5]. However, these implementations lack temporal reasoning on detected workers' actions, which is crucial for ensuring safety. GRUs have been shown to be powerful strategies for inferring pedestrian upcoming actions for autonomous vehicles [6].

Our shared perception intent prediction pipeline harnesses the power of graphs to facilitate information exchange between nodes. Implemented in ROS, our pipeline integrates information about the same scene from different robots. It comprehends the spatial relations between humans in the scene and nearby objects using GNN and utilizes GRUs to understand temporal relations, ultimately predicting the intentions of human workers. The multirobot system enhances robustness by enabling robots to share data, compensating for individual sensor failures and ensuring accurate human intent prediction, which is critical for safety in industrial environments. Additionally, accurate intent prediction would support downstream tasks, such as robot navigation and multirobot task allocation, eventually improving workflows in an industrial setup.

*This work was supported by NSERC and CoRoM

[1] are with the Lab of INIT Robots, Mechanical Engineering, Ecole de technologie supérieure, 1100 Notre-Dame W., Canada `name.surname@etsmtl.ca`

[2] is with MISTLab, Computer and Software Engineering, Polytechnique Montréal, Canada `name.surname@polymtl.ca`

II. System Architecture

A. Image processing and graph creation

The first step is to learn a model on the scene object's relations to the subject's actions. The designed feature vectors contain the most relevant information about the scene objects. Table I contains the objects of interest for the task. We extract these objects of interest from the robot camera images. In real-world deployment, an instance of YOLO [7] will be deployed, however, this experiment relies on the object detection and tracking feature of Isaac Sim [8] that provides accurate tightly and loosely bound 2D bounding boxes.

TABLE I: List of Objects of Interest

Category	**Objects**
Storage Area	Crates, Boxes, Palettes
Workstation	Desks, Chairs, Storage Drawers, Computers
Assembly Station	Assembly Desk, Chair
Manufacturing Station	Machine, Table

We then transform the camera images and the bounding boxes into encoded vectors using a RESNET50 [9] backbone that yields a flattened output vector of length 512. These encoded vectors are then transformed into node features using an approach inspired by [10]. Our graph is star-shaped, with the central node representing the human worker: a 1024-length node feature vector created with its first 512 elements extracted from the raw image and the last 512 elements from the human operator bounding box in the scene. All other nodes are scene objects using the same concatenation strategy. The edge's attributes are computed from the Euclidean distances between the human operator and the respective object in the 2D image.

B. Graph Convolution

To ensure our graph representation is compact enough for sharing over multiple robots and running onboard, we select Graph Convolution Networks [11].

We denote a graph as $G = (V, E)$, where V is the set of nodes and E is the set of edges. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ for a graph with N nodes is defined as $A_{ij} = 1$ if there is an edge from node i to node j, and $A_{ij} = 0$ otherwise. The degree matrix $D \in \mathbb{R}^{N \times N}$ is a diagonal matrix with the i^{th} diagonal element being the degree of node i, i.e., the number of edges connected to node i. To account for self-loops, we modify our adjacency matrix and degree matrix. The adjacency matrix with self-loops becomes $A' = A + I$, where I is the identity matrix. The adjacency matrix looks like:

$$A_{ij} = \begin{cases} 1 & i = j, \\ d_j & i = 1, j \neq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The degree matrix with self-loops becomes $D' = D + I$. We denote the matrix of node features at layer l as $H^{(l)} \in \mathbb{R}^{N \times F^{(l)}}$, where N is the number of nodes and $F^{(l)}$ is the number of features per node at layer l. The input feature matrix $H^{(0)}$ has dimensions $N \times F^{(0)}$, and the output feature matrix $H^{(L)}$ has dimensions $N \times F^{(L)}$, where L is the number of layers. The weight matrix at layer l is denoted as $W^{(l)} \in \mathbb{R}^{F^{(l)} \times F^{(l+1)}}$. The operation of a GCN layer can be represented as follows:

$$H^{(l+1)} = \sigma\left(\left(D'^{-1/2} A' D'^{-1/2}\right) H^{(l)} W^{(l)}\right) \tag{2}$$

In this equation, $D'^{-1/2} A' D'^{-1/2}$ is the normalized adjacency matrix, $H^{(l)}$ is the matrix of node features at the current layer, $W^{(l)}$ is the matrix of weights for the current layer, and σ is a non-linear activation function: we selected ReLU. The message passing is done according to the following expression:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathrm{N}(i)} \frac{1}{\sqrt{D_{ii} D_{jj}}} h_j^{(l)} W^{(l)}\right) \tag{3}$$

For the sake of this experiment, we use 2 layers of this graph convolution operations. The successful implementation of this step in the pipeline completes the spatial understanding module. Once the robot has an understanding of the relationships of the human subject with the nearby objects, the next step to predict the intent is to understand the relation of the states in successive frames.

C. Temporal GRU

For the temporal analysis of the scene, we had a choice between LSTMs [12], and GRUs [13]. We selected GRUs, since, compared to LSTMs, they have proven to be more efficient, consume less memory, and are faster to implement [14]. The input vector is created using the graph representation of the scene discussed in the previous step. This graph is passed through our pre-trained GNN, having its last fully connected layer removed. This gives us an output vector of shape 1x128. Concurrently, the image is passed through YOLOv8 in order to extract the pose information of the human in the scene. The output from the YOLO is flattened, thus, giving us a vector of shape 1x34. Since GRU primarily requires inputs in the form of sequences, the combination of the node embedding and the pose information forms the basis of the input to the GRU. Depending on whether the prediction is being done on a single robot or multiple robots, the length and the composition of the input changes.

Figure 3 shows the composition of the input when we have just a single robot making a prediction about the intent of the human in the scene. In this case, we concatenate the node embeddings from previous time frames, the node embedding, and pose information from the current time frame and pass it through the GRU to conduct a prediction. Our network has been trained on multiple test cases, accounting for various number of successive time frames as described in Sec. III.

D. Decentralized Information Sharing

When dealing with multiple robots, the decision-making process relies not only on its own node embeddings but also on those of neighboring robots. The objective at this stage is to facilitate decentralized information sharing among the

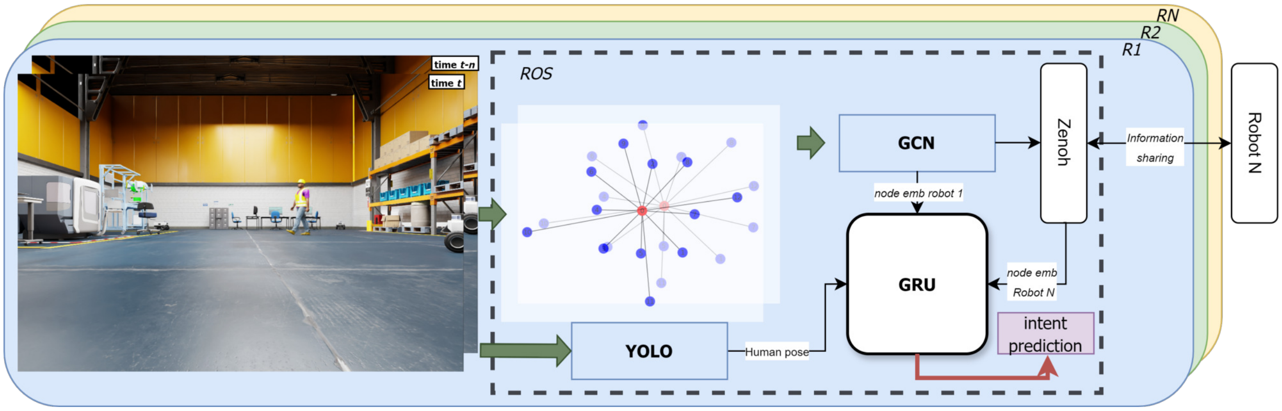

Fig. 2: System architecture: Image parsing, graph formation, spatial prediction, pose information integration, multirobot information sharing and temporal prediction

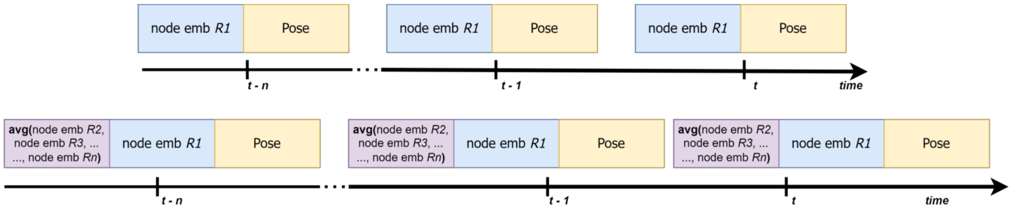

Fig. 3: Variations in the feature vector for temporal predictions

robots. ROS2 offers various existing packages such as nimbro_network [15] and FastDDS to support decentralized communication. However, as the number of robots increases, configuring these tasks becomes increasingly complex. Therefore, we opted for Zenoh [16], a publish/subscribe/query protocol that operates with a set of abstractions in the 4th and 5th layers of the OSI model.

The robots acquire node embeddings from neighboring robots through message passing via Zenoh. The final construction of the input sequence to GRU is illustrated in Fig. 3. Depending on the number of robots involved, the node embeddings from neighboring robots are aggregated and then concatenated with the node embedding and pose information of the main robot. The sequence length is determined based on the number of frames considered for temporal analysis. Subsequently, the GRU is trained on a dataset of these sequences to predict the intent of the human within the scene.

III. Simulation Setup

We evaluated our approach using a realistic simulation setup provided by Nvidia's Isaac Sim [8]. This platform offers a lifelike industrial environment complete with ROS support, human trajectory planing, synthetic dataset generation capabilities. The simulation environment we created is depicted in Figure 1. It features four potential stations for the human worker: storage shelves, workstation, assembly station, and manufacturing unit, resulting in a 4-way classification problem. To simulate human motion, we utilized Isaac Sim's default custom motion model [17], which accounts for both static and dynamic obstacles using reactive collision avoidance and velocity-based position estimation as well. During experimentation, we maintained a constant goal position while randomly selecting the initial position each time the simulation was initiated. We deployed multiple Carter robots [18], numbering from 1 to 5, from Isaac Sim's library. These robots are equipped with RGB and depth sensors, although for this specific experiment, we relied solely on the RGB cameras.

IV. Results and Discussion

Our choice to deploy a multirobot system in a controlled, simple environment serves as a foundational experiment to validate the model's effectiveness before scaling to complex industrial setups. Modeling objects as nodes using GNNs allows us to capture deeper scene dynamics by understanding the spatial relationships between the human and surrounding objects, which is crucial for enhancing predictive accuracy in real-world scenarios.

Extensive testing has been conducted to determine the optimal structure of the model and its features, ensuring a rich representation of the scene while maintaining a lightweight design. Table II presents the results from these tests. Model accuracy was assessed using an independent test set comprising data sequences not utilized during training. A clear trend can be observed in the results: the system performs better when pose information is incorporated into the input, when longer sequences are considered, and when a greater number of robots are deployed. Real-time testing was done with moving and stationary robots, comprising a higher number of unseen scenarios, thus explaining the drop in performance. During real-time experimentation, tests were conducted with various initial positions for the human and the robots. Moreover, the robots moved at a constant velocity, altering the appearance of the scene compared to

TABLE II: Results of the spatiotemporal analysis, comparing a single robot (with and without pose) and multiple robots (including real-time simulation testing). Columns represent the time lengths considered in the past as shown in Fig 3 *Tests not conducted due to high computational requirements of Isaac Sim

1 robot	**T-1**	**T-2**	**T-3**
GNN+GRU+P	74.52	80	83.46
GNN+GRU	62.65	63.5	62.5
2 robots			
Test	90.70	94.6	97.60
Real-time	81.23	81.57	84.76
3 robots			
Test	95.50	97.8	98.80
Real-time	82.22	83.17	85.91
4 robots			
Test	97.05	99	99.5
Real-time	82.46	83.18	88.14
5 robots			
Test	98.0088	99.8	99.8
Real-time	*	*	*

the training dataset, which mostly comprised images from stationary robots. In the case of deploying five robots, we experienced system performance degradation and crashing applications due to the high computational demands of Isaac Sim. Our system uses a Core i9 CPU with 64GB of RAM and an RTX 4070 GPU with 24GB VRAM. Although these specifications are generally sufficient for running complex simulations, the simultaneous movement of multiple robots and the storage of large data volumes can cause system crashes. We intend to test this specific case by running the application in headless mode in the future. The following paragraphs dive deeper into the ablation studies.

A. Dataset

The dataset is designed to capture the motion, appearance, and state of humans within the scene from the perspective of multiple mobile robots positioned at various locations. Each robot captures images at a rate of 30 frames per second (fps). The dataset is organized into sequences, where each sequence represents the human's movement toward a specific goal position. Sequence lengths range from 80 to 400 frames.

Each frame containing a human presence is converted into a graph representation. The dataset comprises a total of 28 042 images, corresponding to an equal number of graphs used for training. Despite variations in sequence lengths based on different goals, efforts were made to maintain a uniform distribution across the dataset. A standard split of 60/20/20 is implemented for training, validation, and testing.

B. Model Structure Variations

1) Graph Structure: We conducted multiple experiments to assess the network's performance while varying the number of layers, trainable parameters, adjacency matrix connectivity, and weight sharing. Our experiments involved varying the number of layers from 1 to 5. Although we observed some improvement in network performance with an increased number of layers, the difference was not substantial enough to justify the corresponding increase in trainable parameters. Given our aim to maintain a compact model suitable for onboard deployment, we settled on two layers.

Additionally, we investigated the structure of the adjacency matrix, noting differences in performance based on the presence or absence of self-loops. The model exhibited better performance when self-loops were included. Since we opted to keep the number of layers constant at 2, weight sharing did not significantly impact performance.

2) GRU Configuration: In multirobot scenarios, the structure of the input sequence must be unified, often achieved through concatenation or aggregation. A straightforward concatenation method involved combining the node embeddings received from neighboring robots in the order they were received. However, this approach posed limitations as the length of the input sequence varied based on the number and order of robots.

Alternatively, aggregating the node embeddings from neighboring robots ensured a consistent input sequence length regardless of the number or order of robots. Surprisingly, both methods produced similar performance results. Consequently, we opted for concatenated vectors as the hidden states of the GRU, utilizing the last hidden layer to predict the intention of the human operator.

V. Conclusion & Future Work

The paper introduces an intent prediction pipeline for decentralized multirobot systems, showing promising initial results from simulation experiments. However, several components are currently under development to achieve a fully integrated shared perception pipeline.

One critical aspect involves ensuring consensus among all robots regarding intent prediction. As some robots may encounter occluded vision or unreliable sensor data, it's essential to enable decentralized sharing of output classification. This facilitates robots with faulty vision to stay updated on predictions made by others in the system. To address this, we plan to leverage Buzz [19], which provides pre-defined constructs for virtual stigmergy and information sharing, crucial for conflict detection and resolution within this task scope.

Additionally, performance evaluation of this system is necessary against benchmarks and current state-of-the-art human motion prediction algorithms. Comparison with non-learning based approaches such as constant velocity models [20] and other human motion models [21], [22] is also essential. While conventional methods may suffice in simple scenarios, we anticipate our deep learning approach to outperform in more complex scenarios [6]. We further hypothesize that incorporating the multirobot aspect can enhance overall system robustness and accuracy.

Currently, our GNN utilizes Euclidean distances as edge attributes obtained from 2D images. To enhance the system's robustness, we aim to increase the dimensionality of edges to incorporate more contextual information. Additionally, the system is designed to detect a single human in the scene, but future work will involve detecting multiple humans and predicting their intent.

References

[1] C. Thorpe and H. Durrant-Whyte, "Field robots," in *Robotics Research: The Tenth International Symposium*. Springer, 2003, pp. 329–340.

[2] L. Liu, F. Guo, Z. Zou, and V. G. Duffy, "Application, development and future opportunities of collaborative robots (cobots) in manufacturing: A literature review," *International Journal of Human–Computer Interaction*, vol. 40, no. 4, pp. 915–932, 2024.

[3] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.

[4] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno, "Multi-robot collaborative perception with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2289–2296, 2022.

[5] X. Ji, H. Li, Z. Pan, X. Gao, and C. Tu, "Decentralized, unlabeled multi-agent navigation in obstacle-rich environments using graph neural networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8936–8943.

[6] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[7] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics (version 8.0. 0)[computer software]," *YOLO by Ultralytics (Version 8.0. 0)[Computer software]*, 2023.

[8] "NVIDIA Isaac Sim," NVIDIA Corporation, accessed: 2024-03. [Online]. Available: https://developer.nvidia.com/isaac-sim

[9] B. Koonce and B. Koonce, "Resnet 50," *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, pp. 63–72, 2021.

[10] K.-H. Zeng, S.-H. Chou, F.-H. Chan, J. Carlos Niebles, and M. Sun, "Agent-centric risk assessment: Accident anticipation and risky region localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2222–2230.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[12] R. C. Staudemeyer and E. R. Morris, "Understanding lstm–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[13] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 2017, pp. 1597–1600.

[14] R. Cahuantzi, X. Chen, and S. Güttel, "A comparison of lstm and gru networks for learning symbolic sequences," in *Science and Information Conference*. Springer, 2023, pp. 771–785.

[15] Max Schwarz, "nimbro network github page," https://github.com/AIS-Bonn/nimbro_network, 2024, accessed: 2024-03-19.

[16] A. Corsaro, L. Cominardi, O. Hecart, G. Baldoni, J. Enoch, P. Avital, J. Loudet, C. Guimarães, M. Ilyin, and D. Bannov, "Zenoh: Unifying communication, storage and computation from the cloud to the microcontroller," vol. DSD 2023, 09 2023.

[17] NVIDIA Corporation, "Warehouse logistics - ext omni anim people — omniverse isaac sim documentation," 2024, accessed: 2024-04-07. [Online]. Available: https://docs.omniverse.nvidia.com/isaacsim/latest/features/warehouse_logistics/ext_omni_anim_people.html

[18] NVIDIA Isaac Sim, "USD Assets - Robots," 2024, accessed: 2024-03-27. [Online]. Available: https://docs.omniverse.nvidia.com/isaacsim/latest/features/environment_setup/assets/usd_assets_robots.html

[19] C. Pinciroli and G. Beltrame, "Buzz: An extensible programming language for heterogeneous swarm robotics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3794–3800.

[20] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 236–243.

[21] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2543–2549.

[22] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.