Robust Foundation Models Empowered RAN Intelligence for Reliable Embodied Robot Scenarios

Anonymous authors
Paper under double-blind review

A Appendix

A.1 Related Work

We provide the state-of-the-art investigations and discussions for foundation model and AI-RAN in robust robot scenarios.

Foundation Model for robots-based scenarios: To ensure accurate robot navigation, the authors in Xia et al. (2024) proposed an agent-based foundation model as the new training paradigm for effective environmental sensing and robot navigation. The solution can generate cross-domain actions consistent with sensing information, paving the way to realize interactive and collaborative robots. However, the FM training can lead to the issue of security due to frequent information transmission between robots and edge RAN. The authors in Chen et al. (2025) developed a FASTNav to train a lightweight LLM, for reliable robot navigation by reduce the transmission data sizes. The proposed method contains three modules: fine-tuning, teacher-student iteration, and language-based multi-point robot navigation. The experiment results the solution can perform high secure and low latency robot navigation. To ensure reliable path planning, the authors in Qi et al. (2025) proposed a 3-step trajectory optimization framework for generating a jump motion for a humanoid robot. The framework can joint detect robot postures, centroidal angular momentum, and landing foot placement for reliable path planning. Nonetheless, the existing works neglect the communication resource constraint with high-frequency information transmission which can also cause lowly robust robot cooperation in 6G application scenarios.

RAN empowered reliable communication for robot cooperation: The reliable robot cooperation needs the support of suitable communication resources. It is important to schedule communication resources effectively by RAN. The authors in Baruffa et al. (2024) developed a testbed architecture that combines contemporary communication and cloud technologies to provide microservice-based mobile applications with the ability to offload part of their tasks to cloud/edge data centers connected by multi-RAT cellular networks. In addition, the authors in Bolla et al. (2023) presented an optimization problem to further explore redundant radio bearers for each robot. Such problem extends the current specification on redundant transmissions. Several heuristic methods have been developed to meet the time-scale requirements that cannot be achieved through exhaustive search. However, such methods might still lead to high communication costs due to high-frequency information exchanges among robots. The authors in Chinchilla-Romero et al. (2024) proposed a centralized control solution for automated mobile robots. A techno-economic analysis was designed to assess the total system cost in an Industry 4.0 robot environment. A sensitivity analysis was also included for the solution identifying the variables with great impact on the system cost. Unfortunately, the existing works ignore the undetermined numbers of robots and missions, which can expose significant pressures for real-time computing and reliable communications among robots.

Based on the discussions mentioned-above, the existing works mainly focus on the study of RAN optimization and FM training while few of investigations pay attention to variability of numbers of robots and missions. In addition, it is difficult to ensure reliable and robust FM training based on the limited computing resources of edge RAN. On the other hand, the existing work may also lead to low-efficiency robot cooperation due to the neglect of change in physical environments. In this case, a terminal-edge cooperative robust FM training is feasible to optimize RAN resources for reliable, accurate, and low-latency robot services.

A.2 Problem Formulation

We formulate corresponding objective function to optimize the robots-based parcel sorting and handling with given constraints.

A.2.1 Analysis of Data Collection

The robots can implement cooperative sensing by dynamically changing self-positions. We first formulate the sensing feasibility constraints based on robots' positions and data size:

$$x_{ist} \;\le\; \sum_{v\in\mathcal{N}(s)} m_{ivt}, \tag{1}$$

$$z_{ist} \;\le\; \bar{q}_s\, x_{ist}, \tag{2}$$

where $s \in \mathcal{S}$ is sensing site with site set $\mathcal{S}$; $v \in \mathcal{V}$ is the spatial position of robots with position set $\mathcal{V}$; $x_{ist} \in \{0,1\}$ and $m_{ivt} \in \{0,1\}$ are sensing decision indicator and position indicator, respectively. $x_{ist} = 1$ when robot i implements the sensing operation for site s at time t; $m_{ist} = 1$ denotes robot i is in the position v at time t; z_{ist} is a continued variance to present the data size (bits) that robot i collects at site s; $\bar{q}_s$ is the maximal data sizes that site s can provide. The two constraints can assist robots in optimizing self-positions for cooperative sensing.

In addition, we need to ensure comprehensive sensing by improving sensing coverage:

$$\sum_{i\in\mathcal{M}}\sum_{t\in\mathcal{T}} z_{ist} \;\ge\; Q_s, \tag{3}$$

where Q_s is the minimal acceptable sensing coverage.

A.2.2 Analysis of Parcel Sorting Accuracy

For the parcel sorting accuracy, we give detailed discussions considering delivery regions, destinations, and latency. Explicitly, in terms of the delivery region, we enable the R-FM-rApp to estimate the current sorting result is whether in the given delivery regions or not:

$$R_k = \mathbb{I}[\,\hat{z}_k \in \mathcal{Z}_k^{\mathrm{srv}}\,], \tag{4}$$

where $R_k \in \{0,1\}$ is an indicator; $\hat{z}_k$ is the estimation result; $\mathbb{I}$is an indicator function. Based on this, we can give the sorting accuracy A_{Ra} considering the delivery regions as follows:

$$A_{\mathrm{Ra}} = \frac{\sum_{k\in\mathcal{K}} w_k\, R_k}{\sum_{k\in\mathcal{K}} w_k} \ge A_{\mathrm{Ra,min}}, \tag{5}$$

where w_k is a weight of parcel k for different priorities; $A_{\mathrm{Ra,min}}$ is the acceptable minimal sorting accuracy.

Considering the delivery destinations, we formulate the corresponding accuracy model. Before that, we first formulate the destination accuracy indicator D_k:

$$D_k = \mathbb{I}[\,\hat{z}_k = g_k\,], \tag{6}$$

where g_k is the real delivery destination of parcel k. Based on this, we can formulate the destination accuracy model by

$$A_{\mathrm{dest}} = \frac{\sum_{k\in\mathcal{K}} w_k\, D_k}{\sum_{k\in\mathcal{K}} w_k}. \tag{7}$$

With the sorting accuracy for different delivery regions, the sorting accuracy for different destinations $A_{\mathrm{dest|Ra}}$ is formulated by

$$A_{\mathrm{dest|Ra}} = \frac{\sum_k w_k\, D_k\, R_k}{\sum_k w_k\, R_k + \varepsilon}, \tag{8}$$

where $\varepsilon > 0$ is a constant value.

In terms of delivery latency, we can further sort the parcels for accurate and real-time handling. Similarly, we first give the latency accuracy indicator S_k:

$$S_k = \mathbb{I}[\,t_k \le d_k\,], \tag{9}$$

where t_k and d_k are practical parcel sorting time and maximal acceptable parcel sorting time, respectively. Then, we give the joint optimization with accurate destination and required latency as follows:

$$A_{\text{SLA}} = \frac{\sum_{k\in\mathcal{K}} w_k\, D_k\, S_k}{\sum_{k\in\mathcal{K}} w_k}, \tag{10}$$

where A_{SLA} is the accuracy indicator for joint optimization of delivery destination and latency. In this case, we further give the whole sorting accuracy A_{mix}:

$$A_{\text{mix}} = \lambda_1 A_{\text{range}} + \lambda_2 A_{\text{dest}} + \lambda_3 A_{\text{SLA}} \geq A_{\text{mix,min}}, \tag{11}$$

where $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$; $A_{\text{mix,min}}$ is the acceptable minimal whole sorting accuracy.

A.2.3 Analysis of Parcel Handling Latency

We analyze the parcel handling latency with two parts: cooperative path planning and dynamic path adjustment. For the cooperative path planning, the latency mainly includes service latency τ^{svc} (namely the time of loading and unloading parcels), handling latency, and the waiting latency for giving way to other robots with collision avoidance. Specifically, the service latency for robot i, $\tau_i^{\text{svc}} > 0$, is regarded as a constant value with static loading and unloading time. Regarding the handling latency τ_i^{hand} of robot i, the planned path for robot i can be represented as PP_i. The average moving velocity of robot i is v_i. We can calculate the handling latency as

$$\tau_i^{\text{hand}} = \frac{\text{PP}_i}{v_i}. \tag{12}$$

The waiting latency is defined to avoid physical collisions based on planned paths. In this case, we can use a graph $G(V, E)$ to present the cooperative parcel handling, where V is the all robot node set and E is the handling path set. For robot i, we firstly need to guarantee $\sum_{i\in\mathcal{M}} m_{ivt} \;\; \leq \;\; 1$, namely each node matches one robot at the same time t. In addition, the edge capacity constraint can be formulated as

$$\sum_{i\in\mathcal{M}} \big(a_{iuv,t} + a_{ivu,t}\big) \;\leq\; \text{cap}_{uv}, \tag{13}$$

where $a_{iuv,t} = 1$ denotes robot i moves from node u to v, and verse visa. cap_{uv} is the maximal robot capability at the edge (u, v). In this case, when robot i and j move towards the same edge at the same time t, the waiting latency $W_{i,t}$ can be formulated by

$$W_i = \left(\sum_{i\neq j} \mathbb{I}[a_{i,j,t'} = 1] \cdot \Delta_{i,j,t'} \right), \tag{14}$$

where $\mathbb{I}[a_{i,j,t'} = 1$ is an indicator function to present there exist a conflict between robot i and j; $\Delta_{i,j,t'}$ is the waiting time of robot i. In this context, we can constrain the whole path planning latency of robot i by

$$t_i^{\text{PP}} = \tau_i^{\text{hand}} + \tau^{\text{svc}} + W_i. \tag{15}$$

The robots can re-adjust their moving paths to cope with the time-varying parcel handling scenarios. We analyze the dynamic path adjustment with two parts: the path adjustment latency $W_{\text{adj},i}$ and the revised handling completion latency $C_{\text{adj},t}$. The former can be formulated as

$$W_{\text{adj},i} = \sum_{t'\leq t} \mathbb{I}[a_{i,v,t'} = 0] \cdot \text{Id}(t'), \tag{16}$$

where $\mathbb{I}[a_{r,v,t'} = 0]$ is an indicator function, where $a_{r,v,t'} = 0$ denotes robot i is in the idle state in node v at time t. $\text{Id}(t')$ denotes the idle time of robot i. The latter can be represented as

$$C_{\text{adj}}(i) = C_i^u + \tau_{\text{rep}}, \tag{17}$$

where C_i^u is the current handling latency based on the previous path planning decision; τ_{rep} is the computing latency for path adjustment. In this context, the whole parcel handling latency t_i^{handle} is constrained by

$$t_i^{\text{handle}} = W_{\text{adj},i} + C_{\text{adj}}(i) + t_i^{\text{PP}} \leq t_{i,\max}^{\text{handle}}, \tag{18}$$

where $t_{i,\max}^{\text{handle}}$ is the maximal acceptable handling latency.

A.2.4 Objective Formulation

In addition to the parcel handling latency, the robots' energy consumption is another important metric due to battery-powered characteristic. We need to constrain the energy budget. Let $\upsilon_i(t)$ (Joules) denote the budget of the robot i at the time slot t that can be spent on parcel handling. The average cumulative budget Υ_i (Joules) is defined to restrict the energy consumption of robot i:

$$E_i^f(t) = \varphi(\upsilon_{i,t}), \tag{19}$$

where $\varphi(x)$ is a mapping function with nonlinear character; $E_i^f(t)$ is a variable due to the undetermined obstacle distributions. Based on this, we can formulate the optimization objective as

$$P1 : \min \left\{ \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \Big[\sum_{i=1}^{M} \sum_{k=1}^{K} a_1 \Delta T_k - a_2 \Delta A_{\text{mix}} \Big] \right\},$$

$$\text{s.t.} \begin{cases} C1 : equation\ 3, equation\ 5, equation\ 11, equation\ 13, equation\ 18 \\ C2 : r_i \geq r_{\text{min}}, \end{cases}$$

where a_1 and a_2 are weights in Lyapunov theory Matrosov (1962); Δ is the difference between the actual and virtual backlog. $\Delta T_k = b_{1,k} - b_{2,k}$, where $b_{1,k}$ and $b_{2,k}$ are the real handling latency and the expected handling latency for parcel k. We expect to minimize the difference value for real-time parcel handling. In terms of $C1$, equation 3, equation 5, equation 11, equation 13, and equation 18 ensure a comprehensive sensing cooperation, accurate parcel sorting with different delivery regions and destination, respectively, path conflict avoidance, and guarantee of low parcel handling latency. $C2$ can guarantee low-latency data transmission for real-time information exchanges among robots, cooperative FM training, and path planning computing.

A.3 Detailed Algorithm Description with Pseudocode

B Parcel Sorting and Handling Algorithm Designs with Robust FMs

In this section, we provide corresponding algorithms for accurate parcel sorting and handling, which consists of three parts: *Robust FM construction FM splitting for accurate parcel sorting* and *FM cooperation for real-time parcel handling*.

B.1 Robust FM construction

Foundation models are vulnerable to several security risks during training, including data poisoning (injection of malicious samples into large-scale corpora), privacy leakage, and backdoor insertion (hidden triggers that cause targeted misbehavior). Moreover, these models often exhibit high sensitivity to adversarial perturbations, which compromises their reliability in safety-critical applications. To mitigate these risks, we design a robust training methods: adversarial training formulates learning as a min–max optimization problem Razaviyayn et al. (2020):

$$\min_{\theta} \ \mathbb{E}_{(x,y)} \left[\max_{\delta \in \mathcal{S}} L\big(f_\theta(x+\delta), y\big) \right], \tag{20}$$

where θ is model parameters (weights of a neural network); (x, y) is the input sample x with its ground-truth label y; $f_\theta(x)$ is model prediction for input x under parameters θl $L(\cdot,\cdot)$ is cross-entropy measuring prediction error; δ is adversarial perturbation applied to input x.; $\mathcal{S}$ is a feasible perturbation set; $\mathbb{E}_{(x,y)}[\cdot]$ is expectation over the training data distribution. We explicitly enhance resistance against input perturbations. The differential privacy can be incorporated to limit information leakage from individual samples. The FM acquisition is illustrated in Fig. 1. The pre-training process is quantified as a Stochastic Game (SG) problem with a tuple $\{S_i, A_i, \mathcal{T}, r_i\}$, where $\mathcal{T}$ is a transfer function and r_i is a reward function. The R_i is formulated by

$$r_i(S_i, A_i) = \frac{1}{k_i} \sum_{k=1}^{k_i} [\Delta D_{j,k} + \Delta D_{i,k}], \tag{21}$$

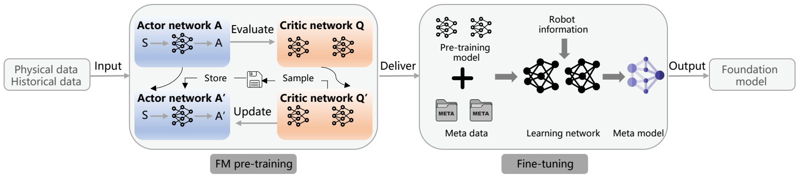


Figure 1: Illustration of cooperative FM construction.

where k_i is the number of parcel; $\Delta[f] = f(t-1) - f(t)$; α_i and β_i can be set as 5 and 0.05 usually, respectively. The robots can learn a feasible pre-training FM model from the critic network by minimizing the reward:

$$\mathrm{FM}_{\mathrm{pre}}(\theta^{\mu}) = E_{S,A \backsim \Omega}\Big[\frac{1}{M}\sum_{i} \nabla_{\theta^{\mu}} \mu(A_i|S_i) \nabla_{A_i} Q^{\mu}(S_i, (A_1, \cdots, A_M))|_{A_i=\mu(S_i)}\Big]. \tag{22}$$

Based on the pre-training FM model, A data sampling operation can be enabled to decompose the data for customized parcel sorting and handling:

$$\mathcal{D}_{\mathcal{K}}^{\text{sorting}} = \{(x_i, c_i, y_i)\}_{i=1}^{K}\,, \quad \mathcal{D}_{\mathcal{K}}^{\text{handling}} = \{(x_j, c_j, y_j)\}_{j=1}^{M}\,, \tag{23}$$

where (x_i, c_i, y_i) and (x_j, c_j, y_j) are information vectors to reflect to the parcel sorting and handling, respectively. We can obtain an accurate parcel sorting FM:

$$\mathrm{FM}_{\text{sorting}} = -\sum_{c=1}^{C} \log p_{\theta}(c \mid x, c), \tag{24}$$

where x is the input information; c is the training label; p is the prediction function. The parcel handling FM is formulated by

$$\mathrm{FM}_{\text{handling}} = \|\mathbf{p} - \hat{\mathbf{p}}\|_1 + \lambda_R\, d_{\mathrm{SO(3)}}(\mathbf{q}, \hat{\mathbf{q}}), \tag{25}$$

where p and $\hat{\mathbf{p}}$ are predictive and actual translation actions of robots, respectively; $\mathbf{q}$ and $\hat{\mathbf{q}}$ are predictive and actual orientation actions of robots, respectively.

B.2 FM Splitting for Accurate Parcel Sorting

We present the implementation process of parcel sorting illustrated in Fig. 2. We propose an adaptive model splitting algorithm that utilizes an attention mechanism to acquire multiple sub-FMs. Specifically, we first formulate the feature vector h_k for accurate parcel sorting with different delivery regions, destinations, and latency:

$$h_k = E_{\theta}(x_k, \mathrm{Env}), \tag{26}$$

where E_{θ} is the shared encoder network parameterized by θ. It maps the input pair (x_k, Env) into a latent representation (embedding); x_k is the raw input features of parcel k; Env is the environment information. The corresponding attention heads are then formulated as

$$p^{\mathrm{R}} = \mathrm{softmax}(W_{\mathrm{R}} h_k), \quad p^{\mathrm{D}} = \mathrm{softmax}(W_{\mathrm{D}} h_k), \quad p^{\mathrm{L}} = \mathrm{softmax}(W_{\mathrm{L}} h_k), \tag{27}$$

where W_{R}, W_{D}, and W_{L} are weight matrices. We employ the decoder to connect these attention heads with model parameters using a connection probability:

$$\mathcal{L}_{\mathrm{R-FM-rApp}} = \alpha\big(-\log p^{\mathrm{R}}_{y^{\mathrm{R}}}\big), \quad \mathcal{L}_{\mathrm{D-FM-rApp}} = \beta\big(-\log p^{\mathrm{D}}_{y^{\mathrm{D}}}\big), \quad \mathcal{L}_{\mathrm{L-FM-rApp}} = \gamma \sum_{k} C_{y^{\mathrm{L}},k}\, p^{\mathrm{L}}_{k}, \tag{28}$$

where α, β, and γ are weights controlling the contribution of the region classification loss. p^{R}, p^{D}, and p^{L}_{k} are probability distributions over all region classes; y^{R} and y^{D} are ground-truth class labels for the current samples. $C_{y^{\mathrm{L}},k}$ is the penalty of predicting class k when the true class is y^{L}. The latency-FM-rApp can assign appropriate robots to handle parcels based on distribution latency. Overall, the attention-based approach can reduce the time of model splitting while enhancing the accuracy of FM model implementation. We can transmit the sub-FMs to the Non-RT RICs to acquire various

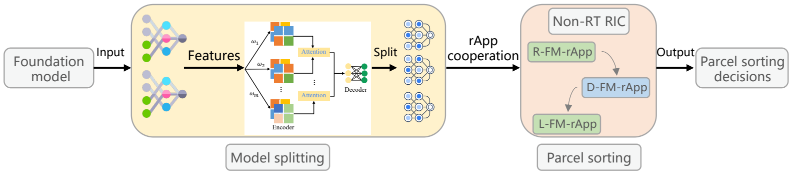


Figure 2: Illustration of parcel sorting.

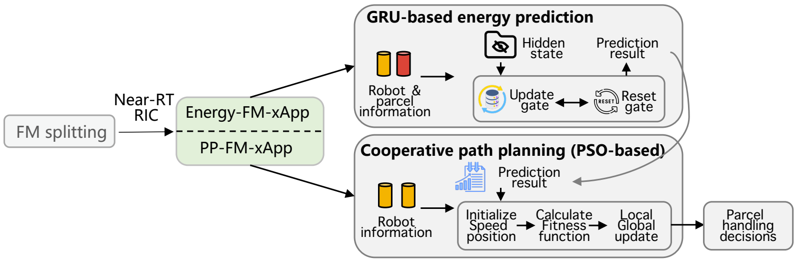

FM splitting
Near-RT RIC
Energy-FM-xApp
PP-FM-xApp
GRU-based energy prediction
Robot & parcel information
Hidden state
Prediction result
Update gate
RESET
Reset gate
Cooperative path planning (PSO-based)
Prediction result
Robot information
Initialize Speed position
Calculate Fitness function
Local Global update
Parcel handling decisions

Figure 3: Illustration of cooperative parcel handling with two kinds of xApps, energy-FM-xApp (E-FM-xApp) and Path Planning (PP-FM-xApp).

rApps. This approach ensures accurate parcel sorting through deep collaboration throughout the entire pipeline. The algorithm is shown in Algorithm 1.

In addition, our pipeline approach can effectively adapt to changes in distribution requirements. When the destination requirements change, we can enable the Non-RT RIC to use the current FM-rApps to generate new sorting decisions by re-estimating distribution destinations for accurate parcel sorting. However, when the latency requirements change, the current FM-rApps may not be able to provide real-time parcel sorting due to their limited FM generation capabilities. To address this, we can implement a re-training operation to update the FM-rApps using an attention mechanism. This method can significantly reduce implementation latency by leveraging historical experiences.

B.3 Model Cooperation for Real-Time Parcel handling

We propose a cooperative parcel handling algorithm based on information on robots and parcels with two main implementation steps shown in Fig. 3: Gated Recurrent Unit (GRU)-based energy prediction and Particle Swarm Optimization (PSO)-based cooperative path planning.

In the Near-RT RIC, we input the robot and parcel information into the input gate. We then activate the reset gate r_t using σ function to determine how much of the previous hidden state should be forgotten when calculating the new hidden state:

$$r_t = \sigma(W_r x_{\text{en},t} + U_r h_{t-1} + b_r)\,, \tag{29}$$

where W_r, $x_{\text{en},t}$, U_r, h_{t-1} are weight matrix, input feature vector at time t, weight matrix for hidden state h_t, and the previous hidden state, respectively. b_r is the bias vector. We can train the GRU network to acquire the accurate energy prediction result:

$$\mathcal{L}_{\text{pre}} = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{E}_t - E_t\right)^2, \tag{30}$$

where $\hat{E}_t = W_y h_t + b_y$. With the prediction results, for robot i at the dimension d, the position and velocity are updated as

$$V_{i,d} = \epsilon V_{i,d} + a_1 r_1 (X_d^{\text{g}} - X_{i,d}) + a_2 r_2 (X_d^{\text{p}} - X_{i,d}), \tag{31}$$

$$X_{i,d} = X_{i,d} + V_{i,d}, \tag{32}$$

where ϵ is the inertia parameter, a_1 and a_2 are acceleration coefficients; r_1 and r_2 are constrained in $[0, 1]$; X^{g} and X^{p} are global and local optimization results that are updated by

$$X_d^{\text{g}} = \arg\min_{X_d} g(X_{i,d}) = \sum_{j=1}^{M}\left\|X_{i,d} - h_j\right\|_2, \quad X_d^{\text{p}} = \arg\min_{X_d} g^{'}(X_{i,d}) = \sum_{j=1}^{M}\left\|X_{i,d} - h_j^{'}\right\|_2, \tag{33}$$

where h_j and $h_j^{'}$ are global and local prediction values, respectively. Consequently, we can implement real-time and accurate parcel handling decisions using our designed path planning algorithm with the aid of Near-RT RIC. The whole algorithm implementation is shown in Algorithm 1.

Our solution enables a cooperative multi-task implementation by collaborating with rApps and xApps. First, we design three types of rApps, area-FM-rApp, destination-FM-rApp, and latency-FM-rApp, to perform cooperative parcel sorting in a pipeline manner. The sorting results from the latency-FM-rApp can be transmitted to the Near-RT RIC via the AI interface for parcel handling. We propose that the energy-FM-xApp and PP-FM-xApp implement cooperative path planning to achieve low-latency parcel handling with energy savings. In turn, the parcel handling results can be transmitted to the Non-RT RIC to assist the rApps in optimizing current computing resource scheduling decisions. Consequently, we can ensure high-efficiency parcel sorting and handling through deep cooperation between the Non-RT RIC and Near-RT RIC in robot-based mobile scenarios.

Algorithm 1 Accurate parcel sorting and real-time handling with robust FMs.

```
// Definition: $\gamma = 0.99$.
Input: Network parameters $\theta$, state space $S_i$, action space $A_i$, update weight $\gamma$; replay buffer;
    system parameter set.
Output: Cooperative sorting and handling results.
1: Construct the FM cooperatively
2: for each episode in all the rounds do
3:     Design the reward function using equation 21
4:     for each time slot $t$ do
5:         for each agent $i \in \mathcal{M}$ do
6:             Obtain the feasible FM models by minimizing the $L$ using equation 22
7:             Implement the customized sorting and handling models using equation 23
8:             Obtain customized sub-FMs using 24 and 25
9:         end for
10:    end for
11: end for
12: Accurate parcel sorting
13: Formulate feature vector $h_k$ using equation 26
14: for each parcel requirement do
15:     Acquire attention heads using equation 27
16:     Obtain the output of sub-FMs using equation 28
17: end for
18: Real-time parcel handling
19: for each iteration do
20:     Calculate the hidden state of GRU using equation 29
21:     Obtain the prediction results using equation 30
22: end for
23: for each exploration time do
24:     Update the positions and velocities using equation 31 and equation 32
25:     Update the global and local optimization results using equation 33
26: end for
```

B.4 Detailed Experiment Results

Fig. 4 illustrates the training loss of the cooperative FM under different numbers of robots, ranging from 20 to 60. We see that across all robots' numbers, the training loss decreases rapidly before iteration 200. The performance reflects efficient convergence at the initial stage. As training progresses, the loss gradually flattens and stabilizes, indicating that the model has reached convergence. In addition, we observe that configurations with more robots achieve lower training losses compared to scenarios with fewer robots. This trend demonstrates that increasing the number of cooperating robots enhances the representational capacity and leads to improved convergence performance. The results imply that our solution can achieve satisfied FM acquisition for efficient parcel sorting and handling.

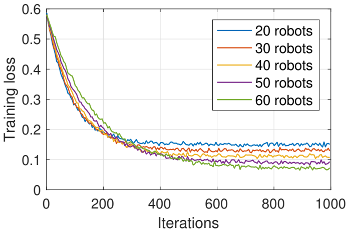


Figure 4: Training loss of cooperative FM with different numbers of robots.

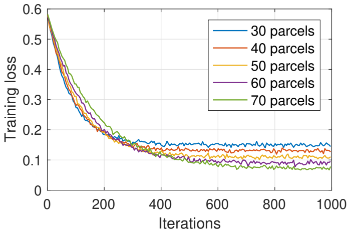


Figure 5: Training loss of cooperative FM with different numbers of parcels.

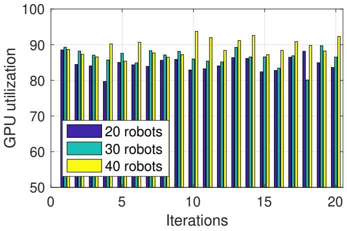


Figure 6: GPU utilization with different numbers of robots.

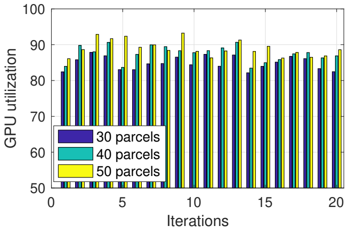


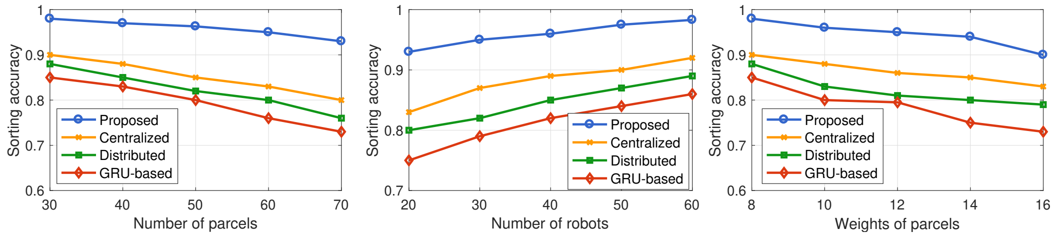
Figure 7: GPU utilization with different numbers of parcels.

Fig. 5 presents the training loss of the cooperative LAM model under different numbers of parcels. We see that the training loss for all solutions decreases sharply within the first 200 iterations followed by a gradual reduction and eventual stabilization. This behavior indicates that the model converges effectively under all workload conditions. Another important observation is that scenarios with a larger number of parcels attain lower training losses compared to cases with fewer parcels. This result demonstrates that increasing the task load can enhance final performance accuracy for accurate parcel sorting. Nevertheless, the convergence becomes slightly slower with higher parcel volumes, which reflects the trade-off between accuracy improvement and training efficiency. We can control the relations between the two indicators for high-efficiency sorting and handling.

Fig. 6 and Fig. 7 depict the GPU utilization of the cooperative FM framework under varying numbers of robots and parcels, respectively. In Fig. 6, the GPU utilization is compared across scenarios with 20, 30, and 40 robots. The results indicate that increasing the number of robots generally leads to higher and more stable GPU utilization. This suggests that larger cooperative groups are able to better exploit computational resources to minimize idle time and ensuring efficient parallel implementation with low execution latency. We can obtain the similar result in Fig. 7. We see that as the number of parcels increases, GPU utilization improves, reflecting the growing workload and enhanced resource occupancy. The higher task load ensures that the GPU remains consistently engaged, which improves training efficiency.

Fig. **??** illustrates the confidence error of FM training with respect to the number of iterations. The confidence error quantifies the gap between the predicted probability distribution and the ground truth, thereby reflecting the reliability of the decision-making. We see that all the solutions exhibit a decreasing confidence error with increasing iterations, indicating improved training stability. However, our solution consistently achieves lower confidence errors compared to other benchmarks. This demonstrates its ability to converge faster and maintain more reliable training predictions. Overall, the results highlight that our solution enhances both the convergence speed and the robustness of FM training, leading to safer and more reliable robot cooperation in parcel sorting and handling.

We compare parcel sorting accuracy in Fig. 8(a) under different numbers and types of parcels. With the random deployment of 40 robots with an average weight of 12 kg, our solution achieves the highest sorting accuracy compared to both benchmarks. This is due to our FM assists O-RAN in scheduling communication resources to ensure seamless cooperation among robots for efficient parcel sorting. Furthermore, our solution decouples missions into three sorting steps using our model-splitting method, which guarantees accurate parcel sorting through streamlined pipeline operations.

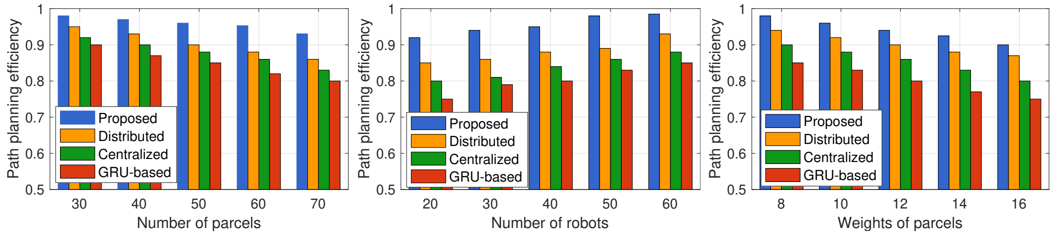


(a) Parcel sorting accuracy vs. number of parcels.

(b) Parcel sorting accuracy vs. number of robots.

(c) Parcel sorting accuracy vs. weights of parcels.

Figure 8: Performance evaluation of sorting accuracy under different numbers of robots, parcels, and weights of parcels.



(a) Path planning efficiency vs. number of parcels.

(b) Path planning efficiency vs. number of robots.

(c) Path planning efficiency vs. weights of parcels.

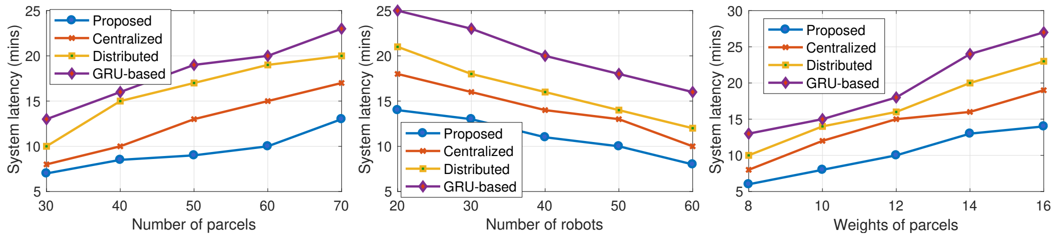
Figure 9: Performance evaluation of path planning efficiency under different numbers of robots, parcels, and weights of parcels.

The latency-FM-rApp also enables robots to sort parcels in real time for low latency requirements. Our solution improves the sorting accuracy by 7.1%, 13.8%, and 15% compared to centralized, distributed and GRU-based solutions, respectively.

Fig. 8(b) shows the performance of sorting accuracy under different numbers of robots. Given the 40 parcels with an average weight of 12 kg, our solution guarantees the highest sorting accuracy compared to all the benchmarks. This is because our FM can achieve deep collaboration among the R-FM-rApp, the D-FM-rApp, and L-FM-rApp to ensure accurate parcel sorting. In addition, our solution can collaborate computing resources of robots and edge RAN to improve the parcel sorting reliability for accurate parcel sorting. Furthermore, the latency-FM-rApp can conduct robots to optimize the sorting results considering different delivery priorities. Overall, our solution improves the sorting accuracy by 8%, 12.5%, and 15.9% compared to the centralized, distributed, and GRU-based algorithm, respectively.

We also illustrate the performance of sorting accuracy with different weights of parcels in Fig. 8(c). With 40 robots sorting 40 parcels, we can see that sorting accuracy reduces as the weights of parcels increase for all the solutions. This is because robots might cause a high recognition error for parcels with heavy weights due to large sizes of parcels. However, our solution still maintains a high sorting accuracy with up to 90% through robot cooperation. It is because our solution can assist robots in selecting feasible cooperators to implement cooperative sorting for high sorting accuracy. Additionally, our solution can collaborate feasible numbers of robots to match suitable numbers of parcels considering different weights of parcels. Overall, our solution improves the sorting accuracy by 9.2%, 15.9%, and 18.8% compared to the centralized, distributed, and GRU-based algorithm, respectively.

With the sorting results, we evaluate the efficiency of path planning, the ratio of the optimal path to the actual path with varying numbers of parcels, as illustrated in Fig. 9(a). Under the same deployment conditions as those in Fig. 8(a), our solution consistently achieves high path planning



(a) System latency vs. number of parcels.

(b) System latency vs. number of robots.

(c) System latency vs. weights of parcels.

Figure 10: Performance evaluation of system latency under different numbers of robots, parcels, and weights of parcels.

efficiency of up to 90% compared to both benchmarks. This is because our energy-FM-xApp allows robots to select feasible numbers of collaborators by estimating the energy consumption for cooperative parcel handling. Based on these estimation results, the PP-FM-xApp collaborates with the robots to implement cooperative path planning through information exchanges utilizing available communication resources. Our solution shortens the handling paths by 13.9% and 28.2% compared to distributed and centralized solutions, respectively.

With the same deployment as Fig. 8(b), Fig. 9(b) provides the path planning efficiency performance with different numbers of robots. The results clearly show that our solution consistently outperforms the other approaches across all robot configurations. In particular, our solution achieves efficiency levels close to or above 90%, even as the number of robots increases with high robustness and scalability. Another important observation is that increasing the number of robots slightly decreases the efficiency for all methods, reflecting the higher complexity of coordination in larger robot groups. Nevertheless, our method maintains a significant advantage over competing approaches, confirming its effectiveness for high-efficiency parcel handling. Our solution improve the path planning efficiency by 6.8%, 10.6%, and 17.5% compared to distributed, centralized and GRU-based solutions, respectively.

Under the same configuration as Fig. 8(c), we give the path planning efficiency performance with different weights of parcels in Fig. 9(c). It can be observed that our method consistently achieves the highest efficiency across all weight levels with up to 90% when the parcel weight is relatively low. Although efficiency decreases slightly as the parcel weight increases to 16, our solution still outperforms all alternatives by a significant margin. This also demonstrates that heavier parcels impose greater coordination challenges for all algorithms, but our method remains the most robust and scalable solution with the aid of terminal-edge cooperation. Our solution improve the path planning efficiency by 3.3%, 6.8%, and 17.5% compared to distributed, centralized and GRU-based solutions, respectively.

Finally, we compare system implementation latency under the different numbers of parcels shown in Fig. 10(a). With the same deployment scheme as shown in Fig. 8(a), we find that our solution consistently achieves the lowest implementation latency compared to both benchmarks. This improvement is attributed to our DDPG algorithm, which reduces the FM implementation time through a cooperative training manner by exchanging implementation actions. Additionally, our solution enables the O-RAN to collaborate with the computing resources of different robots to further accelerate FM training. We can dynamically schedule O-RAN communication resources to ensure reliable information exchanges for cooperative parcel sorting and handling. Our solution reduces system implementation latency by 15.8% and 30.4% compared to the distributed and centralized solutions, respectively.

With the same deployment as Fig. 8(b), Fig. 10(b) provides the system latency performance with different numbers of robots. We see that system latency decreases as the number of robots increases for all solutions, since more robots can share the workload and thus accelerate handling execution. However, our solution consistently achieves the lowest latency across all robot configurations, demonstrating superior scalability and efficiency. For example, with 60 robots, our solution reduces latency to nearly 5 minutes, significantly outperforming the centralized and distributed approaches, which remain above 10 minutes, and the GRU-based method, which exceeds 15 minutes. These results confirm that our solution not only leverages additional robotic resources more effectively but also minimizes communication and cooperation overhead.

Under the same configuration as Fig. 8(c), we give the system latency performance with different weights of parcels in Fig. 10(c). The results show that system latency increases with parcel weight for all solutions. This reflects the additional computational and cooperation overhead required for handling heavier parcels. Our solution consistently achieves the lowest latency across all weight values, maintaining latency below 15 minutes even for the heaviest parcels. These findings highlight that while heavier parcels impose greater system latency, our solution demonstrates superior robustness and scalability, effectively mitigating the latency increase compared to existing baselines. Our solution reduces system implementation latency by 33.3%, 41.2%, and 47.4% compared to the distributed, centralized, and GRU-based solutions, respectively.

REFERENCES

Giuseppe Baruffa, Andrea Detti, Luca Rugini, Francesco Crocetti, Paolo Banelli, Gabriele Costante, and Paolo Valigi. Ai-driven ground robots: Mobile edge computing and mmwave communications at work. *IEEE Open Journal of the Communications Society*, 5:3104–3119, 2024. doi: 10.1109/OJCOMS.2024.3399015.

Raffaele Bolla, Roberto Bruschi, Franco Davoli, Chiara Lombardo, Alireza Mohammadpour, Riccardo Trivisonno, and Wint Yi Poe. Adaptive reliability for the automated control of human–robot collaboration in beyond-5g networks. *IEEE Transactions on Network and Service Management*, 20(3):2489–2503, 2023. doi: 10.1109/TNSM.2023.3298171.

Yuxuan Chen, Yixin Han, and Xiao Li. Fastnav: Fine-tuned adaptive small-language- models trained for multi-point robot navigation. *IEEE Robotics and Automation Letters*, 10(1):390–397, 2025. doi: 10.1109/LRA.2024.3506280.

Lorena Chinchilla-Romero, Jonathan Prados-Garzon, Ramya Vasist, and Meysam Goordazi. Economic feasibility of 5g-based autonomous mobile robots solutions for industry 4.0. *IEEE Communications Magazine*, 62(11):52–59, 2024. doi: 10.1109/MCOM.005.2400125.

V.M Matrosov. On the stability of motion. *Journal of Applied Mathematics and Mechanics*, 26(5):1337–1353, 1962. ISSN 0021-8928. doi: https://doi.org/10.1016/0021-8928(62)90010-2. URL https://www.sciencedirect.com/science/article/pii/0021892862900102.

Haoxiang Qi, Zhangguo Yu, Xuechao Chen, Qingqing Li, Yaliang Liu, Chuanku Yi, Chencheng Dong, Fei Meng, and Qiang Huang. A three-step optimization framework with hybrid models for a humanoid robot's jump motion. *IEEE Transactions on Automation Science and Engineering*, 22:14120–14132, 2025. doi: 10.1109/TASE.2025.3557162.

Meisam Razaviyayn, Tianjian Huang, Songtao Lu, Maher Nouiehed, Maziar Sanjabi, and Mingyi Hong. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Processing Magazine*, 37(5):55–66, 2020. doi: 10.1109/MSP.2020.3003851.

Bingyi Xia, Peijia Xie, and Jiankun Wang. Smart mobility with agent-based foundation models: Towards interactive and collaborative intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 9(7):5130–5133, 2024. doi: 10.1109/TIV.2024.3457759.