Risk-Aware Energy-Constrained UAV-UGV Cooperative Routing using Attention-Guided Reinforcement Learning

Md Safwan Mondal[1,†], Subramanian Ramasamy[1], James D. Humann [2], Jean-Paul F. Reddinger[3], James M. Dotterweich[3], Marshal A. Childers[3], Pranav Bhounsule[1]

***Abstract*— Maximizing the endurance of unmanned aerial vehicles (UAVs) in large-scale monitoring missions spanning over large areas requires addressing their limited battery capacity. Deploying unmanned ground vehicles (UGVs) as mobile recharging stations offers a practical solution, extending UAVs' operational range. This introduces the challenge of optimizing UAV-UGV routes for efficient mission point coverage and seamless recharging coordination. In this paper, we present a risk-aware deep reinforcement learning (Ra-DRL) framework with a multi-head attention mechanism within an encoder-decoder transformer architecture to solve this cooperative routing problem. Our model minimizes mission time while accounting for the stochastic fuel consumption of UAV, influenced by environmental factors like wind velocity, ensuring adherence to a risk threshold to avoid mid-mission energy depletion. Extensive evaluations on various problem sizes show that our method significantly outperforms nearest-neighbor heuristics in both solution quality and risk management. We validate the Ra-DRL policy in a Gazebo-ROS SITL environment with a PX4-based custom UAV and Clearpath Husky UGV. The results demonstrate the robustness and adaptability of our policy, making it highly effective for mission planning in dynamic, uncertain scenarios.**

I. INTRODUCTION

In mission-critical operations involving unmanned aerial vehicles (UAVs), such as disaster response, surveillance, and border security [1]–[5], efficiently covering large areas poses a significant challenge due to UAVs' limited endurance. Frequent recharging disrupts missions and reduces overall effectiveness. To overcome this, collaborative systems utilizing unmanned ground vehicles (UGVs) as mobile recharging depots have emerged. These systems leverage the complementary strengths of UAVs and UGVs, enabling flexible, coordinated operations that extend mission durations and enhance performance across diverse environments [6], [7]. In such dynamic settings, operational risks like failures from stochastic factors, such as wind and energy consumption variations, must be addressed to ensure reliability and efficiency [8], [9].

[1]Md Safwan Mondal, Subramanian Ramasamy and Pranav A. Bhounsule are with the Department of Mechanical and Industrial Engineering, University of Illinois Chicago, IL, 60607 USA. mmonda4@uic.edu, sramas21@uic.edu, pranav@uic.edu [2]James D. Humann is with DEVCOM Army Research Laboratory, Los Angeles, CA, 90094 USA.james.d.humann.civ@army.mil [3]Jean-Paul F. Reddinger, James M. Dotterweich, Marshal A. Childers are with DEVCOM Army Research Laboratory, Aberdeen Proving Grounds, Aberdeen, MD 21005 USA. jean-paul.f.reddinger.civ@army.mil, james.m.dotterweich.civ@army.mil, marshal.a.childers.civ@army.mil

† Corresponding author, *This work was supported by ARO grant W911NF-24-2-0018.

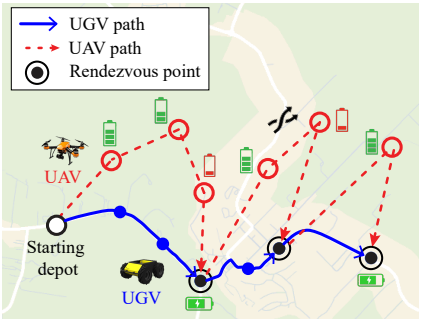

Fig. 1: Illustration of the fuel-constrained UAV-UGV cooperative routing problem: The UAV visits mission points and lands on the UGV to recharge. The objective is to plan routes for both vehicles to minimize the total mission time while accounting for the UAV's stochastic fuel consumption and other vehicular constraints of both the UAV and UGV.

This work addresses a cooperative routing problem where a UAV-UGV team must visit a set of mission points in the shortest possible time. The UAV, limited by battery life, relies on the UGV for recharging, while the UGV, constrained by road networks and speed, must coordinate its movements to support the UAV. Our deep reinforcement learning framework introduces a risk-aware policy (Ra-DRL) that optimizes mission time while accounting for the stochastic nature of UAV fuel consumption under varying wind conditions. Extensive testing across different problem sizes highlights the importance of incorporating operational risks to ensure reliable and efficient UAV-UGV collaboration.

A. Related works

Extensive research has explored various UAV-UGV cooperative routing problems [10], [11] in literature. Maini et al. [12], [13] tackled a routing problem where a fuel-constrained UAV visits mission points while recharging on a UGV traveling along a road network, using Mixed Integer Linear Programming (MILP) to optimize paths. Li et al. [14] studied cooperative UAV-UGV path planning, where the UAV maps terrain and relays obstacle data to the UGV, using a hybrid path planning algorithm for optimization. Ramasamy et al. [15]–[17] approached cooperative UAV-UGV routing by solving the UGV route with K-means clustering and then applying the Vehicle Routing Problem (VRP) with local search heuristics for UAV routing. While many routing problems are modeled as MILP, these become intractable for large-scale instances due to their NP-Hard nature [18]. To address this, various heuristics like genetic algorithms, tabu search, and ant colony optimization are used to produce high-quality approximate solutions within

reasonable time frames [19], [20]. However, these heuristics are often tailored to specific problems, making them labor-intensive and limiting their generalizability and effectiveness in solving large-scale problems with some compromise in solution optimality.

In recent years, learning-based methods have shown significant potential in addressing vehicle routing problems and other combinatorial optimization challenges [21]–[23]. Kool et al. [24] introduced an encoder-decoder Transformer architecture that surpassed classical heuristics in routing tasks, while Li et al. [25] applied DRL with attention mechanisms to the heterogeneous capacitated vehicle routing problem, achieving superior solution quality and efficiency. Fan et al. [26] combined multi-head attention with a DRL policy to optimize routes for energy-constrained UAVs. Despite these advancements, RL-based risk-aware routing remains largely unexplored. In these cases, stochastic programming (SP) can address uncertainty but becomes impractical for large-scale problems due to the high number of variables [27]. Shi et al. [28] used Constrained Markov Decision Process (CMDP) with linear programming for the UAV-UGV rendezvous problem but relied on pre-planned routes for determining sorties. Given that recharging is closely linked to sortie planning, our focus is on jointly optimizing both route planning and recharging rendezvous while accounting for stochastic UAV fuel consumption. To this end, our key contributions are:

1. We tackle the risk-aware, energy-constrained UAV-UGV cooperative routing problem using a CMDP. This is solved through reinforcement learning with Lagrangian relaxation and a policy gradient method, leveraging an encoder-decoder Transformer with attention layers.
2. We evaluate the framework across various problem sizes and risk tolerances, demonstrating strong generalization capabilities. Our approach outperforms baselines in both solution quality and risk management.
3. We develop and validate the policy in a Gazebo simulation environment using PX4 SITL for the UAV and Clearpath Husky for the UGV, incorporating realistic environmental conditions and vehicle dynamics.

II. Problem formulation

A. Problem overview

Consider a collaborative UAV-UGV system, where a fuel-constrained UAV U_a and a recharging UGV U_g are tasked with visiting mission points $\mathcal{M} = \{m_0, m_1, ..., m_n\}$ distributed across a scenario (see Fig. 1). These points are divided into ground points $\mathcal{M}_g$, accessible to both UAV and UGV via a road network G, and UAV-only points $\mathcal{M}_a$, reachable only by the UAV. The UAV, with limited battery capacity F^a and higher velocity v^a, has stochastic fuel consumption influenced by environmental factors like wind, adding uncertainty to its routing. The UGV operates solely on roads at a slower velocity v^g. To avoid fuel depletion, the UAV must periodically rendezvous with the UGV at ground points for recharging, spending a service time T_R before resuming its mission. The mission starts at the depot and ends once all mission points are visited, with the UAV completing a final recharge.

Problem 1: Given mission points $\mathcal{M}$, develop a risk-aware strategy to optimize routes for both the UAV and UGV, accounting for the UAV's stochastic fuel consumption influenced by environmental conditions. The strategy must determine optimal routes and recharging rendezvous between UAV and UGV, minimizing total mission time while ensuring the probability of UAV mid-mission fuel depletion stays below a specified risk threshold.

B. Risk-Aware MDP Formulation

To model the risk-aware UAV-UGV cooperative routing problem as a Markov Decision Process (MDP), incorporating the stochastic nature of the UAV's fuel consumption, we have extend the MDP by introducing risk constraints. Based on [28], the problem is framed as a Constrained Markov Decision Process (CMDP), where the objective is to minimize mission time while ensuring that the likelihood of entering failure states, measured by a risk function remains within acceptable limits δ. This threshold sets the maximum permissible rate of mission failures. The CMDP is defined by the tuple $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} >$ as follows:

1. **State Space** $(\mathcal{S})$**:** The state includes the current positions of the UAV and UGV, the remaining UAV fuel level, and the status of mission points. At each step, the state is $s_t = (p_t, f_t, q_t)$, where $p_t = \{x_t, y_t\}$ represents the position, f_t is the fuel level, and $q_t = \{x^i, y^i, d_t^i\}$ tracks the coordinates and visitation status of mission points $m_i \in \mathcal{M}$, with $d_t^i = 1$ if a point is visited, otherwise $d_t^i = 0$. The failure state s_f represents the UAV running out of fuel.
2. **Action Space** $(\mathcal{A})$**:** Actions $a_t \in \mathcal{A}$ involve selecting mission points for visiting or recharging: $\mathcal{A} = \{\mathcal{M}_g \text{ (recharging)}, \mathcal{M}_g \text{ (visiting)}, \mathcal{M}_a \text{ (visiting)}\}$. Infeasible actions are masked out based on the current state s_t at every decision-making step t.
3. **Reward** $(\mathcal{R})$**:** The reward is the negative of the total mission time, $\mathcal{R} = -\sum_{t=0}^{T} r_t$, where $r_t = r(s_t, a_t) = t_{\text{travel}} + T_R$, representing travel time plus recharge time (for recharging action only) at every step t.
4. **Transition Function** $(\mathcal{T})$**:** State transitions are considered deterministic. The next state's agent position p_{t+1} is updated to the selected location, and fuel is updated as $f_{t+1} = f_t - f_{\text{travel}}$ for visiting, or reset to F^a after recharging. The mission point visitation status is updated for visited mission points accordingly. If the UAV runs out of fuel, it transitions to the failure state s_f.

Definition 1: The risk is defined as the expected number of failures, i.e., the number of times the UAV transitions into the failure state s_f during the mission. For a given policy π, the risk starting from the initial state s_0 is defined as:

$$\rho^{\pi}(s_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} \bar{C}(s_t, \pi(s_t))\right] \tag{1}$$

where T is the finite trajectory length (i.e., the end of the mission), and the risk function $\bar{C}(s_t, \pi(s_t))$ is an indicator

function:

$$\bar{C}(s_t, \pi(s_t)) = \begin{cases} 1 & \text{if } s_t = s_f \text{ (failure)} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In this case, $\bar{C}(s_t, \pi(s_t))$ indicates whether a failure occurs at step t, returning 1 if the UAV transitions to the failure state s_f and 0 otherwise.

Objective: The goal is to find the optimal policy π^* that minimizes the expected mission time while ensuring the number of failures remains below the threshold δ:

$$\pi^* = \arg\min_{\pi} \mathbb{E}\left[\sum_{t=0}^{T-1} C(s_t, \pi(s_t))\right] \tag{3}$$

subject to:

$$\rho^{\pi}(s_0) \leq \delta \tag{4}$$

Here, $C(s_t, \pi(s_t))$ is the cost function representing the mission time at step t.

III. Reinforcement learning framework

In this section, we propose an encoder-decoder transformer network combined with reinforcement learning to learn a risk-aware routing policy π_θ, where θ represents the trainable parameters. Starting from the initial state s_0, the policy π_θ selects an action a_t at each timestep t, determining whether the UAV should visit a mission point or recharge based on the current state s_t. This continues until either the terminal state s_T or failure state s_f is reached. To mitigate UAV failure risk (e.g., fuel depletion leading to s_f), we introduce a Lagrangian-based formulation that balances mission time minimization with risk control, maintaining failure probability within a specified threshold δ. The cooperative route $\mathcal{T}$, defined as the sequence of mission points selected by the UAV to visit or rendezvous for recharging at the UGV, follows the joint probability distribution:

$$\mathbb{P}(\mathcal{T}; \theta) = \prod_{t=0}^{T-1} \pi_\theta(a_t \mid s_t) \tag{5}$$

where T is the total number of timesteps. Assuming deterministic transitions, we have:

$$\mathbb{P}(s_{t+1} \mid s_t, a_t) = 1 \tag{6}$$

In this risk-aware framework, the total cost includes a Lagrangian multiplier λ, which penalizes the policy if the risk $\rho^{\pi}(s_0)$ exceeds the threshold δ. The objective function is:

$$L(\theta, \lambda) = \mathbb{E}\left[\sum_{t=0}^{T-1} \left(C(s_t, a_t) + \lambda \cdot (\rho^{\pi}(s_0) - \delta)\right)\right] \tag{7}$$

where $C(s_t, a_t)$ represents the mission time cost, λ is the Lagrangian multiplier, and $\rho^{\pi}(s_0)$ is the expected number of failures from the initial state s_0. The Lagrangian multiplier λ penalizes the policy when $\rho^{\pi}(s_0) > \delta$, enforcing the risk constraint during optimization. The goal is to find the optimal policy π_θ^* that minimizes mission time while controlling failure risk, formulated as:

$$\theta^* = \arg\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda) \tag{8}$$

Here, θ^* represents the optimal policy parameters, and $\lambda \geq 0$ is iteratively updated to satisfy the risk constraint $\rho^{\pi}(s_0) \leq \delta$. This Lagrangian dual approach balances mission time optimization with risk tolerance, ensuring that UAV failure risk remains within acceptable limits.

A. Encoder-Decoder Transformer architecture

To learn the routing policy π_θ, we use an encoder-decoder transformer architecture (see Fig. 2), adapted from Kool et al. [24] for cooperative routing. The encoder transforms the mission point coordinates into high-dimensional embeddings, and the decoder uses these embeddings to make decisions based on the current scenario state.

1) Encoder

We incorporate a multi-head attention (MHA) mechanism in the encoder to extract higher-dimensional representations from the mission point features. The input is a 3D vector $X = (o_i = \{x_i, y_i, b_i\}, \forall m_i \in \mathcal{A})$, where (x_i, y_i) are normalized coordinates and b_i is a binary variable indicating recharging eligibility. The inputs are linearly projected to $h_i^0 = W^0 o_i + b^0$, with an embedding dimension $d_h = 128$. The embedding is then processed through $L = 3$ multi-head attention layers to create an advanced embedding h_i^L, capturing the relationships among mission points. In each attention layer $l \in L$, *Query*, *Key*, and *Value* vectors are computed from the previous layer embedding h_i^{l-1}, with dimensions $d_q = d_k = d_v = \frac{d_h}{M}$, where $M = 8$ is the number of attention heads. For each head j, the attention scores Z_j^l are calculated as:

$$q_{i,j}^l = h_i^{l-1} W_{q,j}^l, \quad k_{i,j}^l = h_i^{l-1} W_{k,j}^l, \quad v_{i,j}^l = h_i^{l-1} W_{v,j}^l \tag{9}$$

$$Z_j^l = \text{softmax}\left(\frac{q_{i,j}^l {k_{i,j}^l}^T}{\sqrt{d_k}}\right) v_{i,j}^l \tag{10}$$

$$\text{MHA}(h_i^{l-1}) = \text{Concat}(Z_1^l, Z_2^l, ..., Z_M^l) \tag{11}$$

Here, $q_{i,j}^l, k_{i,j}^l, v_{i,j}^l$ are the *Query*, *Key*, and *Value* vectors in head j, and $W_{q,j}^l, W_{k,j}^l, W_{v,j}^l$ are the trainable parameter matrices. The attention output $\text{MHA}(h_i^{l-1})$ is followed by a feed-forward layer (FF) with ReLU activation. Residual skip connections and Batch-Normalization (BN) are applied in both MHA and FF sublayers:

$$\hat{h}_i^l = \text{BN}(h_i^{l-1} + \text{MHA}(h_i^{l-1})) \tag{12}$$

$$h_i^l = \text{BN}(\hat{h}_i^l + \text{FF}(\text{ReLU}(\hat{h}_i^l))) \tag{13}$$

After processing through L attention layers, the final node embedding h_i^L is passed to the decoder for action selection.

2) Decoder:

During each decision-making step, the decoder determines the probability of selecting each available node based on the encoder's node embedding h_i^L and a **context** vector that captures the current scenario state. At step t, the context vector h_c^t is constructed using the agent's current position h_j^L for $j \in p_t$, the agent's fuel level f_t, and the encoder node embedding h_i^L:

$$\bar{h}^t = \frac{1}{n} \sum_{i=0}^{n} h_i^L \tag{14}$$

$$h_c^t = \bar{h}^t W_g + \text{Cat}(h_j^L, f_t) W_c \tag{15}$$

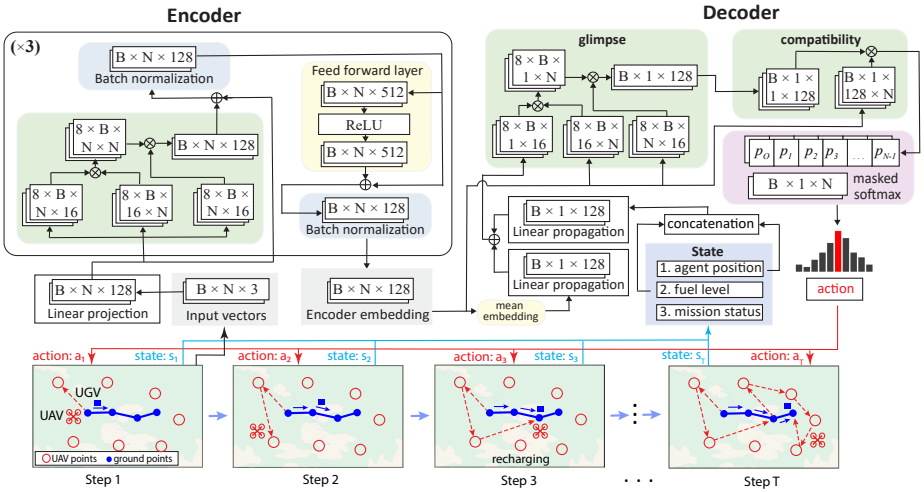

Fig. 2: Architecture of the proposed Transformer network. The encoder consists of three attention layers to generate input embeddings from raw data, while the decoder creates a context vector based on the current state. Both the input embedding and context vector pass through multi-head and single-head attention layers to determine the action, sequentially forming the cooperative route as shown.

Here, W_g and W_c are trainable parameters. The decoder uses the context vector h_c^t as the *Query* and the encoder embeddings h_i^L as *Key/Value* in a multi-head attention (MHA) layer to compute the glimpse h_g^t:

$$h_g^t = \text{MHA}(h_c^t, h_i^L W_k^g, h_i^L W_v^g) \tag{16}$$

Once glimpse h_g^t is obtained, it is used in *Query* q^t and h_i^L in *Key* k^t in a single-head attention layer to compute the compatibility h^t. Infeasible actions are masked out based on constraints, such as previously visited mission points or unreachable points due to fuel levels. The compatibility is calculated as:

$$q^t = h_g^t W_q, \quad k^t = h_i^L W_k \tag{17}$$

$$h^t = \begin{cases} C_p \cdot \tanh\left(\frac{q^t k^{t^T}}{\sqrt{d_q}}\right), & \text{if feasible} \\ -\infty, & \text{else} \end{cases} \tag{18}$$

Here, W_q, W_k are trainable matrices and $C_p = 10$ is a clipping parameter. The output probabilities for the actions are calculated using the softmax function:

$$\pi_\theta(a_t \mid s_t) = \text{softmax}(h^t) \tag{19}$$

Following this process, the decoder selects actions sequentially until the mission is completed or terminated. We use a sampling decoding strategy during training and both greedy and sampling strategies are applied during evaluation.

B. Training method

The training algorithm (see Algorithm 1) is based on the REINFORCE policy gradient method [29] and includes: 1) the policy network π_θ, which selects actions based on a probability distribution, and 2) the baseline network π_ϕ, which selects greedy actions (maximum probability). At each iteration, routes and rewards are computed for a batch, while baseline rewards come from the greedy rollout of the baseline network. The policy parameters θ are updated using the policy gradient algorithm, and the baseline parameters ϕ are updated if they underperform in a paired t-test. To manage risk, we introduce a Lagrangian multiplier λ to balance mission time minimization with risk control. The Lagrangian penalizes the policy if the risk $\rho^\pi(s_0)$ exceeds a predefined risk threshold δ, with λ updated iteratively via gradient ascent. The risk loss is computed encouraging the policy to reduce risk when it exceeds δ. REINFORCE is selected for its ability to learn directly from interactions, maintaining linear complexity in terms of epochs, steps, and gradient computations, making it practical for large-scale problems.

Algorithm 1: Policy network training using REINFORCE with Lagrangian-based risk management

Input: Policy network π_θ, Baseline network π_ϕ, epochs E, Number of batches N, batch size B, episode length T, Risk threshold δ, Lagrangian multiplier λ
Output: Trained policy network $\pi_{\theta'}$

for *epoch in* $1 \ldots E$ **do**
  Sample N batches from dataset
  for *iteration in* $1 \ldots N$ **do**
    for *instance* b *in* $1 \ldots B$ **do**
      Initialize $s_{0,b}$ at $t = 0$
      while $t < T$ **do**
        Get action $a_{t,b} \sim \pi_\theta(a_{t,b}|s_{t,b})$
        Obtain reward $r_{t,b}$, risk $r_{\text{risk},t,b}$, and $s_{t+1,b}$
        $t = t + 1$
      $\mathcal{R}_b = \sum_{t=0}^{T-1} \boldsymbol{r}_{t,b}$
      Risk return $R_{\text{risk},b} = \sum_{t=0}^{T-1} r_{\text{risk},t,b}$
      Baseline reward $\mathcal{R}_b^\phi$, Baseline risk $R_{\text{risk},b}^\phi$ from greedy rollout with π_ϕ
    Compute losses:
      $$\text{Reinforce Loss} = \frac{1}{B}\sum_{b=1}^{B}(\mathcal{R}_b - \mathcal{R}_b^\phi)\log \pi_\theta(s_{T,b} \mid s_{0,b})$$
      $$\text{Risk Loss} = \frac{1}{B}\sum_{b=1}^{B}\lambda(R_{\text{risk},b} - R_{\text{risk},b}^\phi)\log \pi_\theta(s_{T,b} \mid s_{0,b})$$
      $$\text{Total Loss} = \text{Reinforce Loss} + \text{Risk Loss}$$
    Compute gradient:
      $$\nabla_\theta J \leftarrow \nabla_\theta \text{Total Loss}$$
    Update $\theta \leftarrow \theta + \alpha \nabla_\theta J$
    if *Risk threshold violated* **then**
      Update Lagrangian multiplier: $\lambda \leftarrow \lambda + \alpha_\lambda(R_{\text{risk},b} - \delta)$
      Clamp: $\lambda = \max(0, \lambda)$
  if *OneSidedPairedTTest*$(\pi_\theta, \pi_\phi) < 0.05$ **then**
    $\phi \leftarrow \theta$

IV. RESULTS

A. Dataset

We have evaluated the effectiveness of our risk-aware UAV-UGV cooperative routing algorithm through simulations over a 20 km × 20 km area using a single UAV-UGV system. The UAV, traveling at 12 m/s with a fuel capacity of 150 kJ, follows a stochastic fuel consumption model, while the UGV moves at 4.5 m/s on a fixed road network G. UAV mission points ($\mathcal{M}_a$) are uniformly sampled around recharging stops ($\mathcal{M}_g$) on the road network. The UAV's energy consumption accounts for both weight and wind velocity contributions to longitudinal airspeed. Based on [28], the energy consumption model is modeled as: $P(v_\infty) = b_0 + b_1 v_\infty + b_2 v_\infty^2 + b_3 v_\infty^3 + b_4 w + b_5 v_\infty w$ where v_∞ denotes the airspeed and w is the fixed weight of the UAV (2.3 kg). The airspeed v_∞ is the sum of the vehicle's ground speed v^a and the wind velocity component parallel to the vehicle's ground speed, as expressed by:$v_\infty = |v^a + \cos(-\psi) \cdot \xi_{a,b}|$. Here, $\xi_{a,b}$ represents the wind speed, modeled using a Weibull distribution with shape parameters $a = 3$ m/s and $b = 3$, which corresponds to mild steady wind near ground level. The wind direction ψ is uniformly distributed between 0° and 360°. We have trained the model on three problem sizes, **U15G5** (15 UAV and 5 ground points), **U30G10**, (30 UAV and 10 ground points) and **U45G15** (45 UAV and 15 ground points) with 5.12 million instances each, evaluated against two risk thresholds: $\delta = 0.5$ and $\delta = 0.1$. Training is conducted using the Adam optimizer (learning rate 10^{-4}, decay rate 0.995) for 100 epochs with 256 instances per batch on an RTX 2080 Ti GPU. The average training time per epoch is 1.8 minutes for U15G5, 3.2 minutes for U30G10, and 5.1 minutes for U45G15. The model's performance is evaluated across different decoding strategies, risk thresholds, and generalization tests on larger problem sizes and extended road networks.

B. Comparative analysis

Given the complexity of the problem, no standard benchmarks are available, and finding an exact solution becomes computationally intractable as mission points increase. To evaluate model performance, we test on 256 test samples under two risk thresholds ($\delta = 0.05$ and $\delta = 0.1$), comparing the RL-based risk-aware model **(Ra-DRL**) to a nearest neighbor heuristic baseline **(Ra-NN)**, which greedily selects the nearest mission points and triggers recharging when fuel level drops below a threshold γ to avoid failure. For fair comparison, we apply similar masking during action selection. We also compare two DRL decoding strategies: greedy decoding (selecting the highest-probability action) and sampling decoding, where $\mathbb{N}$ trajectories are sampled and the best is chosen. We evaluate with $\mathbb{N} = 1024$ (DRL(1024)) and $\mathbb{N} = 10240$ (DRL(10240)).

The Table I shows that **Ra-NN** exhibits a tradeoff between mission time and risk, with lower fuel thresholds (γ) leading to more conservative decisions and fewer failures. However, reducing the fuel threshold also shortens mission time at the cost of higher failure rates. In contrast, the RL model consistently outperforms Ra-NN, achieving **45-60% shorter mission times** while maintaining **lower risk levels**. This trend becomes more evident as problem sizes increase, with the RL model excelling in both objective value and risk control. Despite Ra-NN's faster computations due to its greedy nature, it sacrifices mission success rates. In the **Ra-DRL policies**, sampling-based decoding reduces mission time by **5-15%** compared to greedy decoding, though it slightly increases risk due to its exploratory nature. Larger sample sizes (1024 vs. 10240) offer diminishing returns in mission time but significantly raise computational costs. As risk tolerance decreases, mission times increase to meet stricter thresholds, particularly in smaller scenarios like U15G5 and U30G10, with less impact in larger scenarios like U45G15. Overall, the results demonstrate the **scalability and efficiency of the RL model**, balancing mission time and risk across varying problem sizes and thresholds.

TABLE I: Comparison evaluation of the DRL policy across problem sizes

Method	U15G5			U30G10			U45G15		
	Obj. (min.)	Time (sec)	Risk (%)	Obj. (min.)	Time (sec)	Risk (%)	Obj. (min.)	Time (sec)	Risk (%)
Ra-DRL (greedy, δ=0.05)	178	0.5	0.03	218	1.11	0.04	260	2.06	0.01
Ra-DRL (1024, δ=0.05)	155	1.65	0.03	186	4.57	0.05	249	6.89	0.02
Ra-DRL (10240, δ=0.05)	151	28.18	0.02	181	48.28	0.04	249	75.51	0.02
Ra-DRL (greedy, δ=0.1)	176	0.5	0.091	210	1.09	0.05	260	2.19	0.04
Ra-DRL (1024, δ=0.1)	141	3.1	0.06	182	4.41	0.07	249	7.95	0.07
Ra-DRL (10240, δ=0.1)	139	34.01	0.08	178	42.32	0.08	248	78.6	0.07
Ra-NN (γ=60%)	344	0.03	0.12	457	0.06	0.13	574	0.13	0.13
Ra-NN (γ=40%)	330	0.03	0.18	435	0.06	0.38	543	0.13	0.45
Ra-NN (γ=20%)	322	0.03	0.30	426	0.06	0.47	531	0.13	0.63

C. Generalization

To assess the generalization of the proposed DRL framework, we evaluate it on larger problem instances by: 1) increasing the number of mission points, and 2) expanding the road network. We generate test cases for configurations: **U60G20** (60 UAV and 20 ground points) and **U75G25** (75 UAV and 25 ground points). The policy trained on U45G15 is applied to these larger instances to gauge its adaptability to more complex and large scenarios.

TABLE II: Performance across larger scenarios

Method	U60G20			U7525		
	Obj. (min.)	Time (sec)	Risk (%)	Obj. (min.)	Time (sec)	Risk (%)
Ra-DRL (greedy, δ=0.05)	264	1.55	0.03	265	1.68	0.04
Ra-DRL (1024, δ=0.05)	250	8.38	0.05	250	9.74	0.03
Ra-DRL (10240, δ=0.05)	244	84.56	0.07	245	96.5	0.06
Ra-DRL (greedy, δ=0.1)	265	1.32	0.06	266	1.66	0.05
Ra-DRL (1024, δ=0.1)	250	8.6	0.10	250	9.85	0.10
Ra-DRL (10240, δ=0.1)	244	86.7	0.12	245	98.3	0.11
Ra-NN (γ=60%)	676	0.27	0.14	790	0.48	0.22
Ra-NN (γ=40%)	632	0.26	0.49	708	0.47	0.59
Ra-NN (γ=20%)	596	0.25	0.76	673	0.45	0.80

Table II shows that the trained risk-aware policy effectively maintains risk thresholds in larger problem instances (U60G20 and U75G25), even though it is originally trained on U45G15. Sampling-based decoding results in approximately 5% shorter mission times compared to greedy decoding but at significantly higher computational cost, similar to the trends observed in Table I. Both $\delta = 0.05$ and $\delta = 0.1$ models manage risk well, demonstrating scalability and robustness, with slight deviations in the $\mathbb{N} = 10240$ sampling strategy. Mission costs remain similar across risk thresholds. Larger sampling sizes (10240 vs. 1024) provide minimal mission time improvements but greatly increase compu-

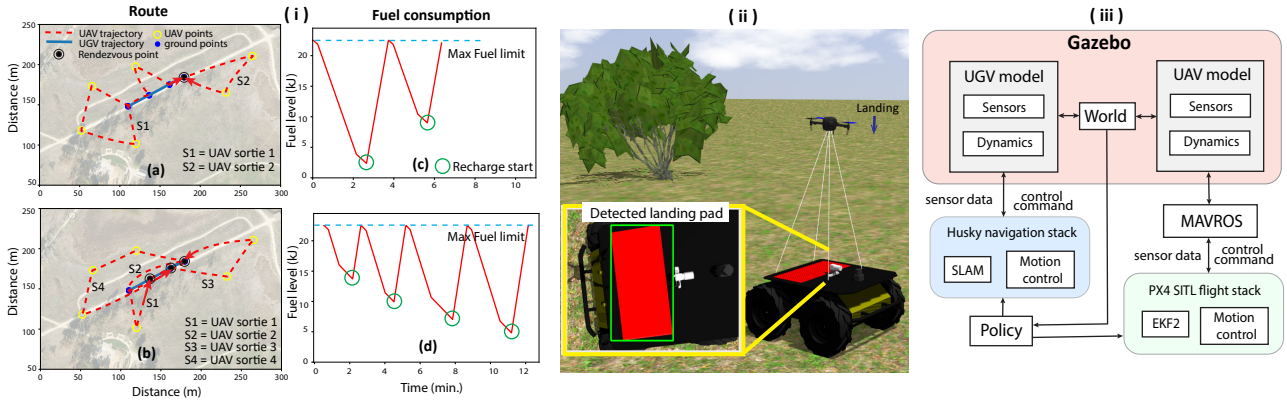

Fig. 3: Gazebo deployment: (i) (a) Mission routes and (c) fuel consumption profile with Ra-DRL($\mathbb{N}$=1024, $\delta = 0.10$) policy, and (b) mission routes and (d) fuel consumption profile with Ra-DRL($\mathbb{N}$=1024, $\delta = 0.05$) policy. Animation can be found at `http://tiny.cc/jg6mzz` (ii) Autonomous UAV landing on the UGV for recharging (iii) The simulation architecture for UAV-UGV cooperative mission.

tational time, consistent with results for smaller problem instances. The Ra-NN baseline shows a higher failure rate and significantly longer mission times compared to our Ra-DRL policies, underscoring the effectiveness of the RL-based approach. Using curriculum learning, we can fine-tune any pretrained model for lower risk thresholds in fewer training epochs (e.g., $\delta = 0.05$ models can be used for warmup when training for $\delta = 0.03$). More details and route animations are available in our repository at `http://tiny.cc/jg6mzz`.

V. Experiments

In this section, we deploy the proposed risk-aware UAV-UGV collaborative routing model in the Gazebo simulation environment to evaluate its performance and risk mitigation capabilities. The simulation involves collaboration between a Clearpath Husky UGV and a custom quadrotor UAV, focusing on task completion, energy consumption, and recharging operations.

1. **UGV platform**: The Clearpath Husky UGV is equipped with a Hokuyo UST10 Laser Scanner, Intel RealSense camera, GPS module, BlackflyS camera, and IMU for localization. The UGV navigates at a speed of 0.5 m/s, using its sensor suite for obstacle detection and navigation.
2. **UAV platform**: A custom-built quadrotor UAV (2.3 kg) with a rear-facing camera, IMU, GPS, and range sensor flies at 3.25 m/s with a maximum fuel level of 22 kJ. The UAV's stochastic energy consumption model is integrated into the UAV model.
3. **Environment model**: Wind velocity, as described earlier, simulates realistic ground-level conditions with an average speed of 2.5 m/s. Dynamic perturbations in wind speed and direction are introduced to mimic unpredictable patterns, challenging the UAV's stability and stochastic energy consumption.
4. **Control architecture**: The simulation runs on an Intel Core i9 system with Ubuntu 20.04 and ROS Noetic. The UAV and UGV communicate via the ROS Master. The risk-aware policy is implemented in Python, generating mission plans based on mission points, wind data, and real-time UAV-UGV positions. The UAV is controlled via PX4 SITL using MAVLink through MAVROS, while the Husky UGV uses gmapping SLAM for navigation (see Fig. 3iii). For precise UAV landing, vision based autonomous landing is employed (see Fig. 3ii), with the UGV recharging the UAV upon successful landing. The implementation code can be found at `http://tiny.cc/jg6mzz`.

In Gazebo, we deploy the Ra-DRL(1024, $\delta = 0.05$) and Ra-DRL(1024, $\delta = 0.1$) policies on the UAV-UGV system at Baylands Park area with 12 mission points. The Ra-DRL($\delta = 0.05$) policy results in a safer route with 4 recharging rendezvous, completing the mission in 12 minutes. In contrast, the Ra-DRL($\delta = 0.1$) policy finishes in 7 minutes with only 2 recharging instances, but the UAV's fuel drops to 11% of capacity, compared to 22% for the safer policy (see Fig. 3i).

VI. Conclusions and Future Work

In this study, we introduce a risk-aware deep reinforcement learning (Ra-DRL) framework that leverages a transformer network with multi-head attention layers to solve the UAV-UGV cooperative routing problem under stochastic fuel consumption. Our approach efficiently optimizes UAV-UGV routes while managing risk thresholds, to minimize mission time and prevent UAV energy depletion failures. The transformer architecture comprising an encoder for generating input embeddings and a state-aware decoder for sequential decision-making, facilitates effective collaboration between the UAV and UGV. The evaluation of our Ra-DRL framework demonstrates several key accomplishments: 1) It outperforms nearest-neighbor heuristic baselines by 45-60% in solution quality and exhibits superior risk management, achieving a more favorable risk-cost tradeoff. 2) It shows strong generalization across larger and more complex problem instances, consistently delivering high-quality solutions. 3) We successfully deploy the framework in a Gazebo-ROS-SITL environment, incorporating real-world vehicle dynamics, proving its viability for realistic mission planning. For future work, we plan to deploy our framework in outdoor real-world settings, addressing vehicle dynamics and real-time constraints. We also aim to explore metaheuristic baselines and extend the framework for multi-UAV-UGV collaboration to enhance performance in diverse mission scenarios.

REFERENCES

[1] Yao Liu, Zhihao Luo, Zhong Liu, Jianmai Shi, and Guangquan Cheng. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. *IEEE Access*, 7:63504–63518, 2019.

[2] Daniel H Stolfi, Matthias R Brust, Grégoire Danoy, and Pascal Bouvry. Uav-ugv-umv multi-swarms for cooperative surveillance. *Frontiers in Robotics and AI*, 8:616950, 2021.

[3] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. *IEEE transactions on robotics*, 32(6):1498–1511, 2016.

[4] Yu Wu, Shaobo Wu, and Xinting Hu. Cooperative path planning of uavs & ugvs for a persistent surveillance task in urban environments. *IEEE Internet of Things Journal*, 8(6):4906–4919, 2020.

[5] Md Safwan Mondal, Subramanian Ramasamy, James D Humann, James M Dotterweich, Jean-Paul F Reddinger, Marshal A Childers, and Pranav Bhounsule. A robust uav-ugv collaborative framework for persistent surveillance in disaster management applications. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1239–1246. IEEE, 2024.

[6] Parikshit Maini and PB Sujit. On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications. In *2015 international conference on unmanned aircraft systems (ICUAS)*, pages 1370–1377. IEEE, 2015.

[7] Mingjia Zhang, Huawei Liang, and PengFei Zhou. Cooperative route planning for fuel-constrained ugv-uav exploration. In *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pages 1047–1052. IEEE, 2022.

[8] Ahmad Bilal Asghar, Guangyao Shi, Nare Karapetyan, James Humann, Jean-Paul Reddinger, James Dotterweich, and Pratap Tokekar. Risk-aware resource allocation for multiple uavs-ugvs recharging rendezvous. *arXiv preprint arXiv:2209.06308*, 2022.

[9] Ahmad Bilal Asghar, Guangyao Shi, Nare Karapetyan, James Humann, Jean-Paul Reddinger, James Dotterweich, and Pratap Tokekar. Risk-aware recharging rendezvous for a collaborative team of uavs and ugvs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5544–5550. IEEE, 2023.

[10] Yao Liu, Zhong Liu, Jianmai Shi, Guohua Wu, and Witold Pedrycz. Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(12):7450–7465, 2020.

[11] Satyanarayana G Manyam, Kaarthik Sundar, and David W Casbeer. Cooperative routing for an air–ground vehicle team—exact algorithm, transformation method, and heuristics. *IEEE Transactions on Automation Science and Engineering*, 17(1):537–547, 2019.

[12] Parikshit Maini, Kaarthik Sundar, Mandeep Singh, Sivakumar Rathinam, and PB Sujit. Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3016–3028, 2019.

[13] Parikshit Maini, Kaarthik Sundar, Sivakumar Rathinam, and PB Sujit. Cooperative planning for fuel-constrained aerial vehicles and ground-based refueling vehicles for large-scale coverage. *arXiv preprint arXiv:1805.04417*, 2018.

[14] Jianqiang Li, Genqiang Deng, Chengwen Luo, Qiuzhen Lin, Qiao Yan, and Zhong Ming. A hybrid path planning method in unmanned air/ground vehicle (uav/ugv) cooperative systems. *IEEE Transactions on Vehicular Technology*, 65(12):9585–9596, 2016.

[15] Subramanian Ramasamy, Jean-Paul F Reddinger, James M Dotterweich, Marshal A Childers, and Pranav A Bhounsule. Coordinated route planning of multiple fuel-constrained unmanned aerial systems with recharging on an unmanned ground vehicle for mission coverage. *Journal of Intelligent & Robotic Systems*, 106(1):30, 2022.

[16] Subramanian Ramasamy, Md Safwan Mondal, Jean-Paul F Reddinger, James M Dotterweich, James D Humann, Marshal A Childers, and Pranav A Bhounsule. Solving vehicle routing problem for unmanned heterogeneous vehicle systems using asynchronous multi-agent architecture (a-teams). In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 95–102. IEEE, 2023.

[17] Md Safwan Mondal, Subramanian Ramasamy, James D. Humann, Jean-Paul F. Reddinger, James M. Dotterweich, Marshal A. Childers, and Pranav A. Bhounsule. Cooperative multi-agent planning framework for fuel constrained uav-ugv routing problem, 2023.

[18] Diego Cattaruzza, Nabil Absi, and Dominique Feillet. Vehicle routing problems with multiple trips. *4or*, 14:223–259, 2016.

[19] Fernando Ropero, Pablo Muñoz, and María D R-Moreno. Terra: A path planning algorithm for cooperative ugv–uav exploration. *Engineering Applications of Artificial Intelligence*, 78:260–272, 2019.

[20] Mengqing Chen, Yang Chen, Zhihuan Chen, and Yanhua Yang. Path planning of uav-ugv heterogeneous robot system in road network. In *Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part VI 12*, pages 497–507. Springer, 2019.

[21] Yoav Kaempfer and Lior Wolf. Learning the multiple traveling salesmen problem with permutation invariant pooling networks. *arXiv preprint arXiv:1803.09621*, 2018.

[22] Quinlan Sykora, Mengye Ren, and Raquel Urtasun. Multi-agent routing value iteration network. In *International Conference on Machine Learning*, pages 9300–9310. PMLR, 2020.

[23] Steve Paul, Payam Ghassemi, and Souma Chowdhury. Learning scalable policies over graphs for multi-robot task allocation using capsule attention networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8815–8822. IEEE, 2022.

[24] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[25] Jingwen Li, Yining Ma, Ruize Gao, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 52(12):13572–13585, 2021.

[26] Mingfeng Fan, Yaoxin Wu, Tianjun Liao, Zhiguang Cao, Hongliang Guo, Guillaume Sartoretti, and Guohua Wu. Deep reinforcement learning for uav routing in the presence of multiple charging stations. *IEEE Transactions on Vehicular Technology*, 2022.

[27] Bin Du, Dengfeng Sun, Satyanarayana Gupta Manyam, and David W Casbeer. Cooperative air-ground vehicle routing using chance-constrained optimization. In *2020 American Control Conference (ACC)*, pages 392–397. IEEE, 2020.

[28] Guangyao Shi, Nare Karapetyan, Ahmad Bilal Asghar, Jean-Paul Reddinger, James Dotterweich, James Humann, and Pratap Tokekar. Risk-aware uav-ugv rendezvous with chance-constrained markov decision process. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 180–187. IEEE, 2022.

[29] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.