# LEARNING HIGHLY ACCURATE CONSERVATIVE STRATEGY BASED ON GRAPH NEURAL NETWORK FOR THE FEASIBILITY PREDICTION OF MULTI-UAV TASK PLANNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

High-throughput multi-UAV task-planning systems must evaluate diverse task instances under limited computational budgets. However, calling a search-based planner for every candidate instance is costly, whereas rejecting a planner-solvable instance is operationally more consequential. Therefore, we study pre-planner screening for a fixed planning pipeline with three budget-dependent outcomes: Feasible, Infeasible, and Unresolved. Since retaining Unresolved as a distinct outcome prevents budget-limited outcomes from being conflated with infeasibility, we propose CONS-GNN, which represents each candidate instance as a relation-aware heterogeneous UAV–task–region graph and predicts the reference planner's outcome using only pre-planner information. CONS-GNN rejects an instance only when its predicted probability of infeasibility exceeds a validation-selected threshold; all other instances are forwarded to the planner. We evaluate CONS-GNN on three independently seeded test sets, each containing 10,000 3D multi-UAV task instances. CONS-GNN achieves mean accuracy, macro-F1, and macro-averaged one-vs-rest PR-AUC scores of 0.9517, 0.9388, and 0.9831, respectively. At the selected operating points, it avoids an average of 41.36% of planner calls while rejecting 3 of 12,449 planner-feasible test instances, corresponding to an observed deployment FRR of 0.0241%. Ablation and scalability analyses identify the contributions of individual graph components and assess performance across UAV team sizes. Together, these results indicate that planner-conditioned screening can reduce planner use with few observed false rejections under the evaluated conditions.

## 1 INTRODUCTION

Planning for multiple unmanned aerial vehicles (UAVs) supports inspection, emergency response, surveillance, and logistics. Requests may combine time windows, energy and capacity limits, no-fly regions, and collision avoidance (Otto et al., 2018; Song et al., 2023; Yao et al., 2019). In a discretized 3D environment, these requirements induce multi-agent path finding (MAPF) and Safe Interval Path Planning (SIPP) subproblems (Stern et al., 2019; Phillips & Likhachev, 2011). Services handling repeated planning requests must allocate limited planner time across candidate instances.

Search-based planners explicitly reason about conflicts and time-dependent free space (Sharon et al., 2015; Li et al., 2021a; Phillips & Likhachev, 2011; Li et al., 2022; Huang et al., 2022; Okumura, 2023; Okumura et al., 2019), but every invocation incurs cost, even when it returns a negative result or exhausts the planning budget. Pre-planner screening is asymmetric: passing a planner-infeasible task instance wastes one call, whereas rejecting a planner-feasible one discards an instance the planner could solve.

Existing learning-based methods predict geometric feasibility, identify important objects, or guide planning (Wells et al., 2019; Silver et al., 2021; Khodeir et al., 2023); the Geometric Reasoning Networks (GRN) approach predicts task-and-motion feasibility with a graph model (Ait Bouhsain et al., 2025). Unlike these within-search signals, pre-planner screening predicts a specified planner's outcome under a fixed budget rather than planner-independent physical feasibility.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

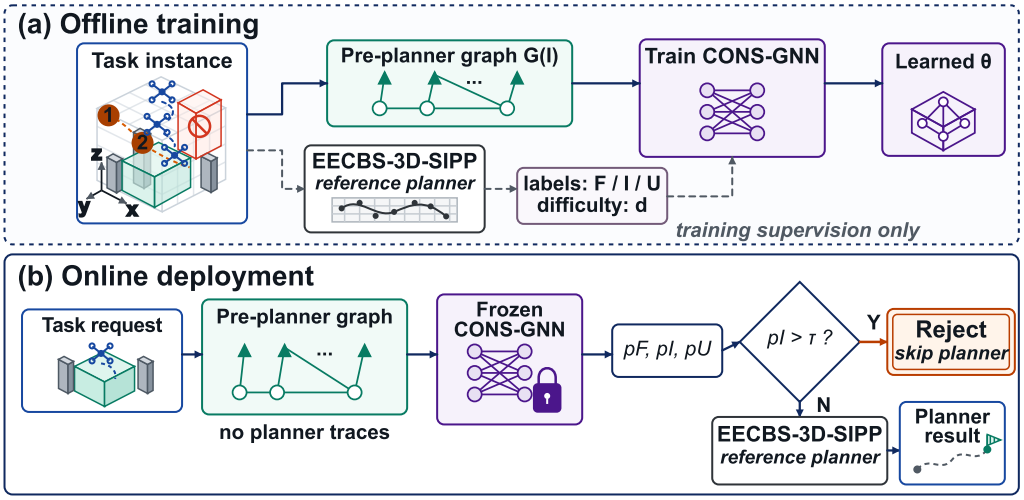



Figure 1: Overview of CONS-GNN. Offline, pre-planner graphs are paired with feasible (F), infeasible (I), and unresolved (U) labels and auxiliary difficulty targets from EECBS–3D-SIPP. Online, CONS-GNN accesses no planner traces; it rejects only when $p_I > \tau$ and otherwise calls the reference planner.

This planner-conditioned setting distinguishes FEASIBLE, INFEASIBLE, and UNRESOLVED, where UNRESOLVED means no definitive decision within budget. Merging UNRESOLVED with INFEASIBLE may lead a screening model to reject planner-solvable instances. We study whether learned screening can reduce planner calls while keeping false rejections rare.

CONS-GNN addresses this question with a typed UAV–task–region graph capturing assignment compatibility, task interactions, airspace use, and regional connectivity. A relation-aware network predicts the planner outcome with auxiliary difficulty supervision, while a validation-selected gate forwards candidates unless their infeasibility scores exceed its threshold.

We make three contributions:

- We formulate multi-UAV pre-planner screening as selective planner-call allocation under asymmetric rejection costs, modeling FEASIBLE, INFEASIBLE, and UNRESOLVED as distinct outcomes of a fixed planner–budget pair rather than as intrinsic feasibility labels.
- We introduce CONS-GNN, which encodes assignment, temporal, airspace, and regional interactions in a relation-aware UAV–task–region graph. It couples difficulty supervision with validation-selected rejection and uses no planner traces at deployment.
- We show that separating UNRESOLVED improves conservative screening: CONS-GNN avoids 41.36% of planner calls with 3 false rejections among 12,449 planner-feasible test instances (observed FRR: 0.0241%), whereas binary label collapse increases FRR tenfold. Ablation, hard-subset, scalability, and ROS 2/Gazebo analyses examine this safety–efficiency tradeoff.

## 2 RELATED WORK

**Multi-agent and safe-interval planning.** Multi-agent path finding formalizes coordination in shared space (Stern et al., 2019). Existing solvers span conflict-based search, safe-interval planning, neighborhood repair, and decentralized coordination, with complementary quality–scalability tradeoffs (Sharon et al., 2015; Li et al., 2021a; Phillips & Likhachev, 2011; Li et al., 2022; Huang et al., 2022; Okumura, 2023; Okumura et al., 2019). Recent variants improve this tradeoff through adaptive anytime large neighborhood search (LNS), reinforcement-learning-assisted repair, and large-scale imitation learning (Phan et al., 2024; Wang et al., 2025; Andreychuk et al., 2025). These methods allocate effort after MAPF search begins; CONS-GNN instead screens candidates before search and leaves the final decision to the reference planner.

**Multi-UAV task planning.** UAV planning couples assignment, routing, timing, resource, and spatial constraints (Otto et al., 2018; Song et al., 2023; Yao et al., 2019). Prior work primarily generates assignments or trajectories under these constraints. We ask which candidates should be submitted to a fixed planner when a service handles repeated requests. We evaluate planner-call savings, false rejections, planner-feasible retention, latency, and throughput alongside classification metrics.

**Learning to guide expensive planning.** Learning accelerates task-and-motion planning by predicting feasibility, identifying important objects, guiding stream search, or estimating geometric cost (Garrett et al., 2021; 2020; Wells et al., 2019; Silver et al., 2021; Khodeir et al., 2023; Kim et al., 2022; Bradley & Roy, 2024). GRN is especially relevant because it uses graph neural networks to predict action and grasp feasibility in robot task-and-motion planning (Ait Bouhsain et al., 2025). Its original task does not address planner-call allocation. We adapt GRN to the same three-way outcome space and pre-planner screening task; the experimental setup specifies the matched inputs and validation-only selection protocol.

**Graph learning and selective deployment.** Graph neural networks support structured combinatorial problems (Bengio et al., 2021; Khalil et al., 2017). Message passing, attention, relational convolution, and heterogeneous transformers model rich node and edge types (Gilmer et al., 2017; Veličković et al., 2018; Schlichtkrull et al., 2018; Hu et al., 2020), while learning-augmented routing and local search allocate expensive computation (Hottung & Tierney, 2020; Li et al., 2021b). CONS-GNN preserves UAV–task–region interactions before planning. Its gate is related to selective classification and learned abstention (Geifman & El-Yaniv, 2017; 2019), but abstention here means explicitly forwarding the candidate to the reference planner. We therefore use validation scores to choose a conservative threshold rather than assuming perfectly calibrated probabilities (Guo et al., 2017).

## 3 Problem Setting

A multi-UAV task instance is represented as

$$\mathcal{I} = (\mathcal{U}, \mathcal{T}, \mathcal{M}, \mathcal{C}), \tag{1}$$

where $\mathcal{U}$ denotes the set of UAVs, $\mathcal{T}$ the set of tasks, $\mathcal{M}$ a discretized 3D airspace map, and $\mathcal{C}$ the temporal, energy, range, no-fly, capacity, and collision constraints. Each UAV has an initial state and resource limits. Each task specifies a start, goal, release time, deadline, priority, and lower-bound path estimate. The fixed reference pipeline first applies deterministic pre-checks and assignment logic, then performs EECBS high-level conflict search (Li et al., 2021a) with constrained 3D SIPP (Phillips & Likhachev, 2011) as the low-level planner.

The planner-conditioned label is

$$y(\mathcal{I}) \in \{\textsc{Feasible}, \textsc{Infeasible}, \textsc{Unresolved}\}. \tag{2}$$

The label FEASIBLE indicates that the reference planner returns collision-free paths within its budget, whereas INFEASIBLE indicates a definitive negative result from the pipeline. The label UNRESOLVED indicates a timeout or another failure to reach a definitive result. These labels describe the behavior of a specific planner under a specified budget, not the intrinsic physical feasibility of a task instance. The UNRESOLVED class may therefore contain both difficult feasible and difficult infeasible instances.

At deployment, the model chooses between PASS and REJECT. Let $a(\mathcal{I})$ denote this action. The objective is to reduce planner calls while controlling the rejection of task instances that the planner could solve. We measure this error using the deployment false-rejection rate:

$$\mathrm{FRR} = \frac{|\{\mathcal{I} : y(\mathcal{I}) = \textsc{Feasible}, a(\mathcal{I}) = \textsc{Reject}\}|}{|\{\mathcal{I} : y(\mathcal{I}) = \textsc{Feasible}\}|}. \tag{3}$$

We select the deployment threshold under an FRR constraint because false rejections discard planner-solvable instances, whereas unnecessary passes incur only an additional planner call.

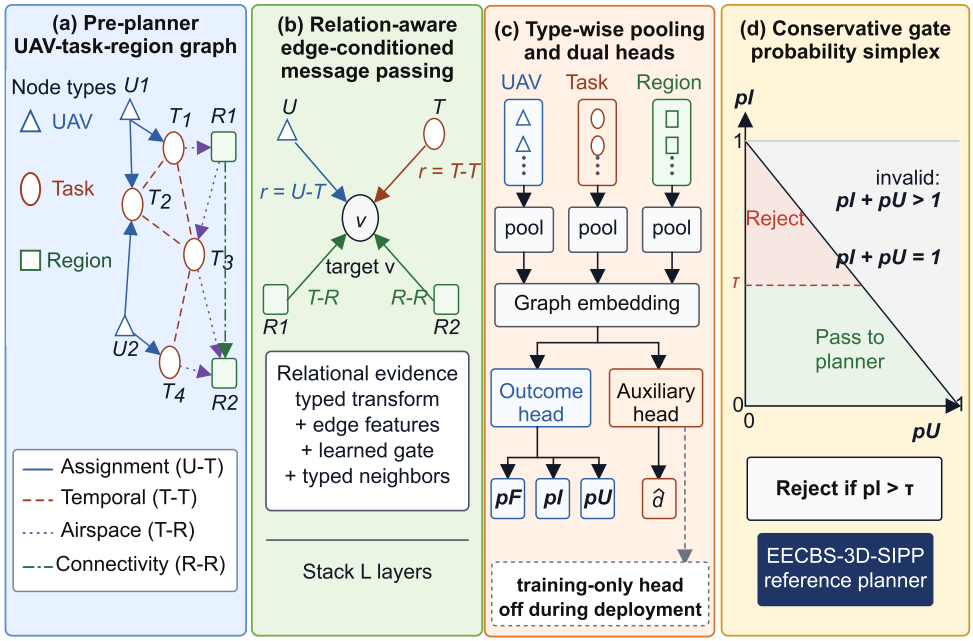



Figure 2: Relation-aware mechanism of CONS-GNN. (a) Typed UAV–task–region relations. (b) Relation-aware, edge-conditioned message passing. (c) Type-wise pooling feeds the outcome head and a training-only difficulty head. (d) The gate rejects candidates with $p_I > \tau$ and forwards all others to EECBS–3D-SIPP.

# 4 METHOD

**Overview.** CONS-GNN constructs a heterogeneous graph, predicts the three planner outcomes, and uses the infeasibility score to decide whether to invoke EECBS–3D-SIPP (Figure 1).

## 4.1 RELATION-AWARE HETEROGENEOUS UAV–TASK–REGION GRAPH

CONS-GNN represents each task instance as a graph $G = (V, E, \tau, \rho)$, where $\tau$ and $\rho$ denote the node- and edge-type functions. The graph contains UAV, task, and region nodes. UAV nodes encode position, speed, energy, range, available time, and accessible altitudes. Task nodes encode the start, goal, release time, deadline, priority, static shortest-path estimates, slack, detour, and static pressure features. Region nodes summarize local airspace properties, including spatial extent, free-space ratio, blocked-boundary ratio, capacity, and pressure.

The graph contains four relation families. UAV–task edges represent service compatibility through the distance to the task start, total lower-bound distance, energy margin, time margin, and altitude compatibility. Task–task edges represent temporal and spatial interactions, including time-window overlap, start and goal proximity, corridor overlap, and deadline gaps. Task–region edges encode lower-bound airspace usage and capacity pressure, whereas region–region edges encode connectivity and local bottlenecks. A UAV may have sufficient resources for an individual task even when several tasks compete for the same vehicle or corridor. UAV–task edges encode the individual assignment margins; task–task and task–region edges capture the competing demands. Region–region edges provide their spatial context. The ablations test how these relations contribute to screening.

All input features are computed before planner invocation. Root-conflict features describe conflicts among independently computed lower-bound paths, before EECBS high-level search. Planner outcomes and search statistics are used only as offline labels and auxiliary targets. We denote the pre-planner feature-construction procedure by BUILDGRAPH.

### 4.2 Relation-Aware Feasibility Prediction

As shown in Figure 2(a)–(c), CONS-GNN first projects UAV, task, and region nodes, together with their typed edge attributes, into a shared hidden space. Each relation-aware layer then performs edge-conditioned message passing. Relation-specific transformations and gates aggregate assignment, temporal, corridor, airspace, and regional information. Node $v$ aggregates edge-conditioned messages from its neighbors across relations $r$:

$$h_v^{(\ell+1)} = \sigma\Big(W_0^{(\ell)} h_v^{(\ell)} + \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{N}_r(v)} \alpha_{uv}^{r,\ell} \cdot W_r^{(\ell)}[h_u^{(\ell)}; e_{uv}]\Big), \tag{4}$$

where $h_v^{(\ell)}$ denotes the embedding of node $v$, $e_{uv}$ is the edge feature vector, and $\alpha_{uv}^{r,\ell}$ is a learned gate computed from the source, destination, and edge attributes. Cross-type relations also transmit context in the reverse direction, allowing UAV, task, and region nodes to exchange information without duplicating relation definitions.

After message passing, mean and max pooling are applied separately to UAV, task, and region nodes. The pooled representations are concatenated to form a graph embedding that feeds a three-class outcome head and an auxiliary difficulty head. We optimize the following loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(y, \hat{y}) + \lambda_d \mathcal{L}_{\mathrm{Huber}}(d, \hat{d}), \tag{5}$$

where $\mathcal{L}_{\mathrm{CE}}$ is class-balanced cross-entropy and $d$ contains the normalized runtime and counts of conflicts, expansions, and low-level calls. The auxiliary objective encourages the shared representation to encode planning difficulty in addition to the three-way outcome. The difficulty head is used only during training; Algorithm 1 gives the complete training and deployment procedure.

### 4.3 Conservative Deployment Gate

As shown in Figure 2(d), the deployment rule rejects only candidates assigned high predicted probabilities of infeasibility. Let the outcome head produce $(p_F, p_I, p_U)$ for FEASIBLE, INFEASIBLE, and UNRESOLVED. We choose $\tau$ using the validation set and reject a candidate only when

$$a(\mathcal{I}) = \textsc{Reject} \quad \text{if} \quad p_I > \tau. \tag{6}$$

We jointly select a checkpoint and threshold on validation data to maximize planner-call savings subject to $\mathrm{FRR}_{\mathrm{val}} \leq 0.5\%$. Ties are broken by lower validation FRR; if no candidate satisfies the constraint, the best validation checkpoint uses $\tau = 1$ to pass all instances. The selected model and threshold are frozen before test evaluation. The deployment action is distinct from the most likely class: an instance can be assigned the largest probability to INFEASIBLE and still be forwarded if that probability does not exceed $\tau$. Rejecting an instance thus requires more confidence than assigning it to the most likely class. Algorithm 1 details the procedure, and Supplementary Section B gives the threshold grid and training configuration.

## 5 Experiments

### 5.1 Experimental Setup

**Data and labels.** We generate each task instance synthetically and label it using the same EECBS–3D-SIPP reference planning pipeline evaluated at deployment. Table 1 summarizes the maps, task-instance distribution, data split, and feature provenance. The planner uses a default budget of 0.03 s, $w = 1.1$, and fixed search limits. We examine the sensitivity of the labels to this restrictive budget in Section 5.4.

**Baselines and protocol.** For seeds 37, 41, and 43, we generate disjoint training, validation, and test sets containing 35,000, 5,000, and 10,000 task instances, respectively. The baselines are Rule Filter; XGBoost (Chen & Guestrin, 2016) and a global-feature multilayer perceptron (MLP-Global); Graph Attention Network (GAT) (Veličković et al., 2018), message-passing neural network

**Algorithm 1** CONS-GNN training, validation-based threshold selection, and online screening

**procedure** TRAINANDSELECT($\mathcal{G}, \mathcal{P}_B, \mathcal{T}, \epsilon_{\mathrm{FRR}}, \lambda_d$)
  Generate disjoint sets $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{val}}$
  **for** each $\mathcal{I} \in \mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{val}}$ **do**
    $G(\mathcal{I}) \leftarrow$ BUILDGRAPH($\mathcal{I}$)
    $(r(\mathcal{I}), d(\mathcal{I})) \leftarrow$ RUNPLANNER($\mathcal{P}_B, \mathcal{I}$)
    $y(\mathcal{I}) \leftarrow$ OUTCOMELABEL($r(\mathcal{I})$)
                                  ▷ $d$ is used only for offline supervision
  Train $f_\theta$ on $\mathcal{D}_{\mathrm{tr}}$ only with
  $\mathcal{L}_{\mathrm{CE}} + \lambda_d \mathcal{L}_{\mathrm{Huber}}$
  Retain checkpoints; let $\theta_{\mathrm{best}}$ have the best metric on $\mathcal{D}_{\mathrm{val}}$
  **for** each checkpoint $\theta_c$ and $\tau \in \mathcal{T}$ **do**
    Obtain outcome-head probabilities $p_I^{(c)}$ on $\mathcal{D}_{\mathrm{val}}$
    Compute $\mathrm{Savings}_{\mathrm{val}}(\theta_c, \tau)$ and $\mathrm{FRR}_{\mathrm{val}}(\theta_c, \tau)$
  $\mathcal{C} \leftarrow \{(\theta_c, \tau) : \mathrm{FRR}_{\mathrm{val}}(\theta_c, \tau) \le \epsilon_{\mathrm{FRR}}\}$
  **if** $\mathcal{C} = \emptyset$ **then**
    $(\theta^\star, \tau^\star) \leftarrow (\theta_{\mathrm{best}}, 1)$  ▷ pass-all fallback
  **else**
    Select $(\theta^\star, \tau^\star) \in \mathcal{C}$ maximizing $\mathrm{Savings}_{\mathrm{val}}(\theta_c, \tau)$
                                  ▷ break ties by lower $\mathrm{FRR}_{\mathrm{val}}(\theta_c, \tau)$
  **return** $\theta^\star, \tau^\star$

**procedure** SCREENANDPLAN($\mathcal{I}, f_{\theta^\star}, \tau^\star, \mathcal{P}_B$)
  $G(\mathcal{I}) \leftarrow$ BUILDGRAPH($\mathcal{I}$)
  $(p_F, p_I, p_U) \leftarrow \mathrm{softmax}(f_{\theta^\star}^{\mathrm{out}}(G(\mathcal{I})))$
  **if** $p_I > \tau^\star$ **then**
    **return** REJECT without invoking $\mathcal{P}_B$
  **else**
    **return** $\mathcal{P}_B(\mathcal{I})$

(MPNN) (Gilmer et al., 2017), relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018), Heterogeneous Graph Transformer (HGT) (Hu et al., 2020); and Adapted GRN (Ait Bouhsain et al., 2025). Adapted GRN replaces the original action and grasp outputs with the three-way outcome head used by the other models. It uses the same data splits and pre-planner inputs. Every evaluated method uses the same single-threshold rejection rule, $p_I > \tau$. Each $\tau$ is selected only on the corresponding validation set under the same FRR constraint.

**Model and optimization.** CONS-GNN uses three relation-aware layers of width 128 with dropout 0.10. We train with AdamW (Loshchilov & Hutter, 2019), learning rate $10^{-3}$, weight decay $10^{-4}$, batch size 64, and difficulty-loss weight $\lambda_d = 0.15$. Training runs for at most 55 epochs, with

Table 1: Benchmark, data splits, and labeling protocol.

| Item | Setting |
|---|---|
| Labels | Feasible / Infeasible / Unresolved |
| Labeler | EECBS–3D-SIPP reference planning pipeline; 0.03 s planning budget, $w = 1.1$, fixed search limits |
| Split | 35k / 5k / 10k training / validation / test task instances per seeded run |
| Test distribution | Feasible task instances across seeds 37 / 41 / 43: 4,000 / 4,149 / 4,300 (40.00% / 41.49% / 43.00%); Infeasible or Unresolved task instances: 6,000 / 5,851 / 5,700 |
| Map and tasks | $10 \times 10 \times 3$ and $12 \times 12 \times 3$ grids; 3–4 UAVs; 6%–16% obstacles; 0%–3% no-fly density |
| Feature provenance | Only pre-planner task, map, static lower-bound, assignment, and root-conflict features |

Table 2: Main-test results over three independent 10,000-instance test sets. Each cell shows the mean with sample standard deviation beneath it. All gates use the same validation FRR constraint. Latency is the reported time per instance in milliseconds; FRR and savings are percentages.

| Method | Acc ↑ | F1 ↑ | PR-AUC ↑ | Savings ↑ | FRR ↓ | Latency ↓ |
|---|---|---|---|---|---|---|
| Rule Filter | 0.4912 | 0.3068 | 0.3841 | 6.40 | 0.0241 | 17.61 |
| | ±0.0008 | ±0.0010 | ±0.0010 | ±0.02 | ±0.0009 | ±0.02 |
| XGBoost | 0.7782 | 0.7416 | 0.8246 | 27.85 | 0.4258 | 17.45 |
| | ±0.0006 | ±0.0007 | ±0.0005 | ±0.06 | ±0.0077 | ±0.03 |
| MLP-Global | 0.7792 | 0.7337 | 0.8236 | 27.64 | 0.4499 | 17.50 |
| | ±0.0004 | ±0.0005 | ±0.0004 | ±0.04 | ±0.0080 | ±0.02 |
| GAT | 0.9459 | 0.9335 | 0.9811 | 38.26 | 0.0482 | 16.83 |
| | ±0.0003 | ±0.0003 | ±0.0002 | ±0.09 | ±0.0017 | ±0.02 |
| MPNN | 0.9454 | 0.9331 | 0.9815 | 39.11 | 0.1766 | 16.76 |
| | ±0.0004 | ±0.0004 | ±0.0002 | ±0.08 | ±0.0088 | ±0.02 |
| R-GCN | 0.9437 | 0.9308 | 0.9808 | 37.42 | 0.0319 | 16.92 |
| | ±0.0003 | ±0.0003 | ±0.0002 | ±0.07 | ±0.0127 | ±0.02 |
| HGT | 0.9417 | 0.9288 | 0.9802 | 37.03 | 0.0801 | 16.97 |
| | ±0.0003 | ±0.0003 | ±0.0002 | ±0.04 | ±0.0113 | ±0.02 |
| Adapted GRN | 0.9462 | 0.9341 | 0.9810 | 37.65 | 0.0325 | 16.73 |
| | ±0.0004 | ±0.0004 | ±0.0002 | ±0.08 | ±0.0152 | ±0.02 |
| CONS-GNN | **0.9517** | **0.9388** | **0.9831** | **41.36** | **0.0241** | **15.31** |
| | ±0.0003 | ±0.0003 | ±0.0002 | ±0.07 | ±0.0009 | ±0.03 |

early stopping after 15 epochs without validation macro-F1 improvement. Retained checkpoints are then evaluated with the validation-only gate-selection rule. Supplementary Section B records the hardware, software, and threshold grid.

**Metrics and operating points.** Accuracy and macro-F1 measure three-way classification performance. For each class, we compute the area under its one-vs-rest precision–recall curve by trapezoidal numerical integration and report the macro-average across the three classes as PR-AUC. Planner-call savings are the fraction of candidates rejected before planning. Deployment FRR is the fraction of planner-feasible task instances rejected by the validation-selected rule; their retention is its complement. We report latency per task instance and throughput in task instances per second. Table 2 reports three-run means and sample standard deviations; Supplementary Table S6 also reports planner-feasible retention. For every evaluated method, the validation threshold $\tau$ maximizes planner-call savings subject to $\epsilon_{\mathrm{FRR}} = 0.005$. Supplementary Table S1 lists the selected thresholds and runtime statistics.

### 5.2 Main Results

CONS-GNN achieves the highest accuracy (0.9517), macro-F1 (0.9388), and PR-AUC (0.9831) among the evaluated methods (Table 2). Its gate avoids 41.36% of planner calls and retains 99.9759% of planner-feasible instances. There is one false rejection in each run: three among 12,449 planner-feasible instances in total, with a mean FRR of 0.0241%.

The savings exceed Adapted GRN's by 3.71 percentage points at comparable observed FRR. Rule Filter matches CONS-GNN's observed FRR but saves only 6.40% of calls. GAT also achieves similar predictive scores, yet saves 38.26% of calls. Similar classification scores can therefore yield different planner-call savings when the gate must retain planner-feasible instances. Supplementary Figure S7 plots the operating points, and Figure S6 gives decision counts and stress-subset results.

Mean reported latency is 15.31 ms per instance, compared with 16.73 ms for Adapted GRN, an 8.5% reduction. Supplementary Table S1 reports runtime variability and throughput. The measurements do not separate graph construction, inference, and planning costs.

### 5.3 Ablation and Diagnostic Evidence

Table 3 reports savings after removing each model component, while Supplementary Figure S5 reports the corresponding macro-F1 changes. Every ablation reduces both macro-F1 and planner-call savings. Removing task–task edges causes the largest decrease in macro-F1, whereas removing edge features causes the largest savings loss on both the standard test set and hard subset. On the hard subset left after deterministic pipeline pre-checks, the full model still avoids 33.48% of planner calls. Its decisions therefore use information beyond deterministic reachability and resource filters. Relative to the full model, the ablations reduce overall savings by 5.23–6.00 percentage points and hard-subset savings by 1.71–2.53 points. Variants with similar macro-F1 scores can still yield different savings at their selected thresholds.

Supplementary Figure S3 illustrates representative decisions: congested planner-feasible instances and timed-out cases with $p_I \leq \tau$ are forwarded, whereas clear assignment bottlenecks are rejected.

### 5.4 Outcome Modeling and Budget Sensitivity

We test the role of UNRESOLVED by merging it with INFEASIBLE during training while retaining the graph inputs, architecture, and validation-only selection protocol. Table 4 shows that this binary formulation rejects more candidates but also rejects more planner-feasible instances. The three-way model has a tenfold lower observed FRR at its selected operating point. Both observed test FRRs are below 0.5%. These operating points show different tradeoffs but do not establish dominance at matched test risk.

To assess label sensitivity to the restrictive 0.03 s budget, we relabel 3,000 instances at $2\times$ and $5\times$ that budget. Only 0.80% and 1.57% of labels change, respectively. Most labels remain unchanged, although this experiment does not evaluate the gate under the new budgets.

### 5.5 Scalability

Table 5 evaluates teams of 4–10 UAVs. Neither graph model falsely rejects an instance in these finite stress sets. CONS-GNN avoids more planner calls at every team size and reaches 70.50% savings with 10 UAVs. Its advantage over Adapted GRN is nonmonotonic and ranges from 0.67 to 1.42 percentage points across the tested team sizes. Supplementary Figure S4 visualizes the corresponding savings, runtime, throughput, and retention trends.

Table 3: Planner-call savings on the standard and hard subsets (mean $\pm$ sample standard deviation over three seeds). The hard subset excludes instances rejected by deterministic pre-checks; W/O denotes without.

| Variant | Standard-test savings (%) ↑ | Hard-subset savings (%) ↑ |
|---|---|---|
| CONS-GNN | $\mathbf{41.36 \pm 0.07}$ | $\mathbf{33.48 \pm 0.09}$ |
| W/O Region Nodes | $35.51 \pm 0.08$ | $31.11 \pm 0.10$ |
| W/O Task-Task Edges | $35.69 \pm 0.09$ | $31.30 \pm 0.11$ |
| W/O Task-Region Edges | $35.59 \pm 0.07$ | $31.19 \pm 0.09$ |
| W/O Difficulty Head | $36.13 \pm 0.08$ | $31.77 \pm 0.10$ |
| W/O Edge Features | $35.36 \pm 0.08$ | $30.95 \pm 0.12$ |
| W/O Heterogeneity | $35.98 \pm 0.07$ | $31.61 \pm 0.09$ |

Table 4: Three-way versus binary outcome modeling at validation-selected operating points. Savings and FRR are percentages.

| Outcome model | Planner-call savings ↑ | FRR ↓ |
|---|---|---|
| CONS-GNN (three-way) | 41.36 | **0.0241** |
| Binary collapse | **52.96** | 0.2414 |

Table 5: Scalability on 1,200 instances per team size. Both methods retain all planner-feasible instances. Latency is seconds per instance and throughput is instances per second; bold marks higher savings and throughput at each team size.

| UAVs | Method | Planner-call savings (%) | FRR (%) | Lat. | Thr. |
|---|---|---|---|---|---|
| 4 | Adapted GRN | 39.83 | 0.000 | 0.0498 | 20.08 |
| 4 | CONS-GNN | **41.25** | 0.000 | 0.0496 | **20.17** |
| 6 | Adapted GRN | 51.67 | 0.000 | 0.1019 | 9.82 |
| 6 | CONS-GNN | **52.34** | 0.000 | 0.0978 | **10.22** |
| 8 | Adapted GRN | 60.17 | 0.000 | 0.0701 | 14.27 |
| 8 | CONS-GNN | **61.17** | 0.000 | 0.0698 | **14.33** |
| 10 | Adapted GRN | 69.17 | 0.000 | 0.0734 | 13.63 |
| 10 | CONS-GNN | **70.50** | 0.000 | 0.0646 | **15.48** |

**Additional validation.** Six ROS 2/Gazebo scenarios, with 40 instances each, cover urban obstacles, no-fly restrictions, and congested corridors. The gate saves 32.5%–50.0% of calls with no observed false rejection; forwarded instances can still time out or deadlock. Supplementary Tables S3–S4 report the complete closed-loop results. A separate PR2 action/grasp analysis shows modest improvements over the GRN reference (Supplementary Table S2).

## 6 Discussion and Limitations

The hard-subset results help distinguish learned screening from deterministic filtering. Even after pre-checks remove obvious failures, CONS-GNN avoids 33.48% of planner calls. The ablations further indicate that both task interactions and edge attributes matter, although their largest effects occur on different metrics. Separating UNRESOLVED gives the model a way to represent budget-limited outcomes without labeling them infeasible. At the selected operating points, this choice reduces false rejections at the cost of lower savings than binary collapse.

Our experiments use one EECBS–3D-SIPP pipeline and synthetic task distributions. The three seeded runs support descriptive comparisons; we have not established statistical significance. The validation FRR constraint is empirical; it provides no bound on unseen or shifted inputs. A change in planner, search limits, or candidate distribution requires renewed threshold validation. The PR2 comparison and tests with up to ten UAVs cover only a limited range of transfer settings. ROS 2/Gazebo simulations also omit real-flight sensing, communication, and timing effects, and the runtime results lack a breakdown of graph construction, inference, and planning costs.

In deployment, the reference planner remains responsible for trajectories and their constraints. Unfamiliar inputs should be forwarded until the gate has been validated for them. Outcome audits and suspension of rejection when validation no longer applies are possible precautions. We have not evaluated either as an automated capability.

## 7 Conclusion

CONS-GNN combines a relation-aware UAV–task–region graph with three-way planner-outcome prediction and a validation-selected rejection gate. Across three test sets, it avoids 41.36% of planner calls while rejecting only 3 of 12,449 planner-feasible instances. Component ablations and the hard subset support the use of structured pre-planner information. The binary counterfactual illustrates the tradeoff from merging unresolved outcomes with negative ones. Future work should test risk-matched operating curves, transfer across planners and task distributions, and real-flight deployment with complete computational cost accounting.

### AI USE STATEMENT

Generative AI tools assisted with language polishing, manuscript organization. The authors are responsible for the final manuscript.

### REPRODUCIBILITY STATEMENT

Implementation, training, threshold-selection, and evaluation details are provided in the supplementary material.

## REFERENCES

Smail Ait Bouhsain, Rachid Alami, and Thierry Siméon. Learning geometric reasoning networks for robot task and motion planning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/561dd71d04695c85e1c0f6a0fbe0bd85-Abstract-Conference.html.

Anton Andreychuk, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. MAPF-GPT: Imitation learning for multi-agent pathfinding at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23126–23134, 2025. doi: 10.1609/aaai.v39i22.34477.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421, 2021. doi: 10.1016/j.ejor.2020.07.063.

Christopher Bradley and Nicholas Roy. Learning feasibility and cost to guide TAMP. In *Experimental Robotics*, Springer Proceedings in Advanced Robotics, pp. 203–216. Springer, 2024. doi: 10.1007/978-3-031-63596-0_18.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pp. 440–448, 2020. doi: 10.1609/icaps.v30i1.6739.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. doi: 10.1146/annurev-control-091420-084139.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4878–4887, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

André Hottung and Kevin Tierney. Neural large neighborhood search for the capacitated vehicle routing problem. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pp. 443–450, 2020. doi: 10.3233/FAIA200124.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference*, pp. 2704–2710, 2020. doi: 10.1145/3366423.3380027.

Taoan Huang, Jiaoyang Li, Sven Koenig, and Bistra Dilkina. Anytime multi-agent path finding via machine learning-guided large neighborhood search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9368–9376, 2022. doi: 10.1609/aaai.v36i9.21168.

Elias B. Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6348–6358, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d9896106ca98d3d05b8cbdf4fd8b13a1-Abstract.html.

Mohamed Khodeir, Ben Agro, and Florian Shkurti. Learning to search in task and motion planning with streams. *IEEE Robotics and Automation Letters*, 8(4):1983–1990, 2023. doi: 10.1109/LRA.2023.3242201.

Beomjoon Kim, Luke Shimanuki, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Representation, learning, and planning algorithms for geometric task and motion planning. *The International Journal of Robotics Research*, 41(2):210–231, 2022. doi: 10.1177/02783649211038280.

Jiaoyang Li, Wheeler Ruml, and Sven Koenig. EECBS: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12353–12362, 2021a. doi: 10.1609/aaai.v35i14.17466.

Jiaoyang Li, Zhe Chen, Daniel Harabor, Peter J. Stuckey, and Sven Koenig. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10256–10265, 2022. doi: 10.1609/aaai.v36i9.21266.

Sirui Li, Zhongxia Yan, and Cathy Wu. Learning to delegate for large-scale vehicle routing. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26198–26211, 2021b. URL https://proceedings.neurips.cc/paper/2021/hash/dc9fa5f217a1e57b8a6adeb065560b38-Abstract.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Keisuke Okumura. LaCAM: Search-based algorithm for quick multi-agent pathfinding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11655–11662, 2023. doi: 10.1609/aaai.v37i10.26377.

Keisuke Okumura, Manao Machida, Xavier Défago, and Yasumasa Tamura. Priority inheritance with backtracking for iterative multi-agent path finding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 535–542, 2019. doi: 10.24963/ijcai.2019/76.

Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018. doi: 10.1002/net.21818.

Thomy Phan, Taoan Huang, Bistra Dilkina, and Sven Koenig. Adaptive anytime multi-agent path finding using bandit-based large neighborhood search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17514–17522, 2024. doi: 10.1609/aaai.v38i16.29701.

Mike Phillips and Maxim Likhachev. SIPP: Safe interval path planning for dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5628–5635, 2011. doi: 10.1109/ICRA.2011.5980306.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607, 2018. doi: 10.1007/978-3-319-93417-4_38.

Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015. doi: 10.1016/j.artint.2014.11.006.

Tom Silver, Rohan Chitnis, Aidan Curtis, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Planning with learned object importance in large problem instances using graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11962–11971, 2021. doi: 10.1609/aaai.v35i13.17421.

Jia Song, Kai Zhao, and Yang Liu. Survey on mission planning of multiple unmanned aerial vehicles. *Aerospace*, 10(3):208, 2023. doi: 10.3390/aerospace10030208.

Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, pp. 151–158, 2019. doi: 10.1609/socs.v10i1.18510. URL https://ojs.aaai.org/index.php/SOCS/article/view/18510.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Yutong Wang, Tanishq Duhan, Jiaoyang Li, and Guillaume Sartoretti. LNS2+RL: Combining multi-agent reinforcement learning with large neighborhood search in multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23343–23350, 2025. doi: 10.1609/aaai.v39i22.34501.

Andrew M. Wells, Neil T. Dantam, Anshumali Shrivastava, and Lydia E. Kavraki. Learning feasibility for task and motion planning in tabletop environments. *IEEE Robotics and Automation Letters*, 4(2):1255–1262, 2019. doi: 10.1109/LRA.2019.2894861.

Weiran Yao, Naiming Qi, Neng Wan, and Yongbei Liu. An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. *Aerospace Science and Technology*, 86:455–464, 2019. doi: 10.1016/j.ast.2019.01.061.