ON FAIR TEAM ORIENTEERING PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Team Orienteering Problem (TOP) has a wide range of applications, ranging from logistics and human/robot patrol to door-to-door marketing. In TOP, each agent must jointly select an appropriate node subset and plan a feasible route connecting all its selected nodes, constrained by its route budget. Existing TOP solutions mainly focus on the efficiency tailored to maximize the selected node reward of all agents. Yet, studies in the social psychology literature have shown that fair task assignment is crucial for various multi-agent applications. However, existing literature fall short on modeling and solving the TOP with fairness consideration. In this work, we leverage the max-min reward among the agents as the fairness objective in TOP and formalize the new challenge as Fair TOP (fTOP). We propose FLOG (Fairness-aware Latent Orienteering based on Graph), a novel fTOP solution framework integrating spatial reward fields and policy learning to enable fairness-aware node selection and route planning. Across synthetic and real-world datasets, FLOG substantially improves the reward fairness over the state-of-the-art efficiency-oriented (SOTA) TOP solver, while maintaining competitive total reward. Specifically, averaged over all evaluated scales, FLOG improves the min-to-max reward ratio from 0.235 to 0.743 compared with the SOTA solver, while reducing the total reward by only 1.3%. Our code and datasets are available at: `https://github.com/route-research/FLOG`.

1 INTRODUCTION

Team Orienteering Problem (TOP) (Gao et al., 2026; Xu et al., 2020; Mansfield et al., 2021) is a fundamental multi-agent routing problem with applications in logistics, human/robot patrol, and door-to-door marketing. In TOP, multiple agents need to select subsets of nodes and construct budget-constrained routes, aiming to maximize the total collected reward of all selected nodes. TOP and its variants have been widely studied in recent years. Related problems such as Subset Selection Problem (SSP) (Natarajan, 1995; Davis et al., 1997) and Orienteering Problem (OP) (Zhang & Vorobeychik, 2016; Qian et al., 2017; Bian et al., 2020) investigate budget-constrained selection and routing under reward maximization objectives. Existing studies have explored exact, approximation, and heuristic approaches for solving these problems. For TOP, early works explored exact optimization methods, such as branch-and-bound algorithms (Boussier et al., 2007), while recent studies mainly focused on approximation and greedy strategies for improving total reward (Xu et al., 2020; 2021). However, these studies mainly focus on maximizing aggregate reward, while overlooking the imbalance of reward allocation among individual agents in multi-agent routing scenarios under constraints.

Although maximizing total reward improves overall efficiency, it may lead to imbalanced reward allocation among agents, where some agents receive substantially fewer rewards than others. Fairness has been widely recognized as an important factor affecting cooperation, satisfaction, and performance in multi-human scenarios (Brawley & Pury, 2016; Boiney, 1995). This consideration also applies to multi-robot systems, where fair scheduling can balance energy consumption and improve system longevity (Beyazit et al., 2024), while affecting human-robot collaboration efficiency (Brandao et al., 2020; Londoño et al., 2024). Several studies have investigated fairness-aware TOP variants; however, existing approaches still suffer from scalability and optimization limitations. Related fairness-aware logistics studies mainly focus on task allocation rather than joint routing optimization (Basık et al., 2018; Hosseini et al., 2025; Sobhanan et al., 2026), leaving fairness-aware TOP largely unexplored. Specifically, Gross et al. (Gross et al., 2025) studied a bi-objective multi-constrained TOP variant and adopted NSGA-II to search for Pareto solutions balancing reward and fairness. However, such

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.

evolutionary search suffers from long execution time and may improve fairness at the sacrifice of total reward. Sanchez et al. (Sánchez-Yepez et al., 2023) proposed a greedy algorithm for fTOP, which fell into local optima, like general heuristics algorithms. A recent work (Boyer et al., 2026) proposed a branch-and-cut based algorithm for fTOP, which however suffers prohibitive long execution time. For example, for a small instance with 102 nodes and 58 agents, its execution time reaches $228s$ on average and up to $3803s$ in extreme cases.

Motivated by the limitations of fairness-aware routing methods, we study the Fair Team Orienteering Problem (fTOP) from a reinforcement learning perspective. Specifically, we adopt max-min reward as the fairness objective, seeking to improve the reward of the worst-served agent while satisfying travel-budget and node-exclusivity constraints. To solve this problem, we propose FLOG (**F**airness-aware **L**atent **O**rienteering based on **G**raph), a graph-based reinforcement learning framework that learns latent node and agent representations to guide fairness-aware orienteering route construction. FLOG learns latent representations of nodes and agents to capture spatial, reward, and agent-level interactions for fairness-aware route construction. Using these representations and dynamic routing states, FLOG evaluates candidate actions and prioritizes under-rewarded agents during route construction. Finally, a fairness-preserving residual reward recovery mechanism exploits the remaining travel budgets to recover additional reward without excessively degrading the attained fairness.

The contribution of the paper is summarized as follows:

- We formulate fTOP as a single-objective max–min optimization problem that maximizes the worst-served agent's reward under travel-budget and node-exclusivity constraints, providing a direct measure of agent-level reward fairness beyond the conventional TOP objective.
- We propose **FLOG**, a graph-based reinforcement learning framework integrating multi-directional latent-field encoding, fairness-guided route construction, and fairness-preserving reward recovery. FLOG combines latent node–agent interactions with dynamic routing states to prioritize under-rewarded agents and recover reward while maintaining fairness.
- Experiments on synthetic and real-world datasets show that FLOG improves reward fairness while preserving reward efficiency. Averaged over all evaluated scales, FLOG improves the min-to-max reward ratio from 0.235 to 0.743 compared with the SOTA solver GRIP (Hu et al., 2026), while reducing the total reward by only 1.3% . Moreover, at the dataset-scale level, GRIP's min-to-max reward ratio falls to 0.05 or below in 12.5% of settings, whereas FLOG maintains a ratio of at least 0.659 across all settings, achieves the highest minimum agent reward in 91.1% of them, and ranks the second-highest in all remaining settings.

The remaining of the work is organized as follows: Sec.2 reviews the work closely related to TOP and fTOP; Sec.3 states the formal definition of fTOP; Sec.4 presents the detailed design of FLOG, our solution to fTOP; Sec.5 demonstrate the comprehensive evaluation of FLOG, followed by the conclusion of the work in Sec.6.

2 Literature Review

2.1 Research on TOP

TOP lies at the intersection of subset selection and routing optimization. Classical subset selection problems select a subset of elements to maximize utility under relatively simple constraints, such as cardinality or additive budgets (Natarajan, 1995; Davis et al., 1997), with representative applications including influence maximization (Lei et al., 2015; Chen et al., 2009; 2016; Aral & Dhillon, 2018) and coverage maximization (Gupta & Levin, 2020; Karimi et al., 2017). Once routing constraints are introduced, however, the cost of selecting a subset becomes non-additive, as it depends jointly on the spatial configuration of the selected nodes and their visiting order. Some researchers (Zhang & Vorobeychik, 2016) studied subset selection under routing-dependent costs and proposed a generalized cost-benefit greedy algorithm, while some subsequent works (Qian et al., 2017; Bian et al., 2020) further explored evolutionary approaches to search a broader solution space, albeit at potentially high computational cost on large-scale instances (Fan et al., 2024). The Orienteering Problem (OP) represents a canonical routing-constrained subset selection problem, in which a single agent selects and visits a subset of reward-bearing nodes within a travel budget to maximize the collected reward.

TOP extends OP to multi-agent scenarios. In TOP, each agent constructs a budget-feasible route, and each candidate node can be visited by at most one agent. This extension couples three decisions: which nodes to select, how to allocate the selected nodes among agents, and how each agent should construct its route. These coupled decisions substantially enlarge the combinatorial search space. Early studies mainly focused on exact approaches. For example, the work (Boussier et al., 2007) developed a branch-and-bound-based exact algorithm, whose computational cost limits its applicability to relatively small instances. Subsequent studies therefore investigated heuristic and metaheuristic methods to improve practical scalability (Keller, 1989; Gunawan et al., 2016), while some researchers (Xu et al., 2020; 2021) developed approximation algorithms with provable approximation guarantees for TOP and its generalized variant. More recently, learning-based approaches have also been explored for closely related multi-agent budget-constrained routing problems. GRIP (Hu et al., 2026), for example, learns reward-oriented node selection and route planning to efficiently construct high-quality multi-agent routes. Despite their different solution paradigms, existing TOP and related multi-agent routing methods mainly focus on maximizing aggregate team reward. Such an objective does not explicitly control how rewards are distributed among individual agents: a solution with a high total reward may still assign substantially different rewards to different agents. Agent-level reward fairness therefore remains insufficiently addressed in existing TOP research.

2.2 Research on Fair Task Assignment

Fairness has been widely studied in multi-agent task assignment across crowdsourced delivery, mobility, and sensing applications. Existing studies consider fairness from different agent-level perspectives, including completed tasks, payoff, revenue, and operational cost. In early stage, the study (Basık et al., 2018) incorporated courier fairness while maximizing the number of completed deliveries, while a recent work (Ignatius & Bahsoon, 2023) studied fair task assignment among heterogeneous delivery agents such as human couriers and robots. From an economic-reward perspective, some researchers (Zhao et al., 2021) proposed heuristic algorithms that reduce payoff differences among workers while maintaining a high average payoff, and a recent work (Li et al., 2024) considered platform revenue maximization subject to individual courier revenue-fairness constraints. A subsequent work (Hosseini et al., 2025) instead considered fairness in courier costs and developed a cost-oriented heuristic assignment strategy. The research (Wang et al., 2025) explored a reinforcement learning approach involving fairness-aware task allocation, while the recent study (Sobhanan et al., 2026) evaluated fairness-aware assignment on real-world data and reported improved fairness with a limited reduction in cost efficiency.

2.3 Literature Summary

Taken together, existing TOP studies mainly focus on aggregate team reward, while fairness-aware task-assignment studies generally do not account for routing-dependent decision coupling. This gap motivates the study of Fair Team Orienteering Problem (fTOP), where agent-level reward fairness is considered alongside routing efficiency under travel-budget constraints.

3 Notations and Problem Formulation

Let $\mathcal{A} = \{a_1, \ldots, a_M\}$ be the set of M agents and $\mathcal{V} = \{v_1, \ldots, v_N\}$ the set of N candidate nodes, where v_i has a non-negative reward r_i. Each agent a_j has a start location s_j and an end location e_j. Since this paper focuses on fairness, all agents are set to have an identical travel budget, denoted by B, ensuring that fairness is not confounded by unequal budgets. Let P_j denote the route of agent a_j from s_j to e_j. When no ambiguity arises, we also use P_j to denote its node set. The route length is denoted by L_j, whose collected reward is $R_j = \sum_{v_i \in P_j} r_i$. This work uses Euclidean distance between two locations v_i and $v_{i'}$ as the travel cost of the , which is denoted by $d(v_i, v_{i'})$. The classical TOP formulation maximizes the total collected reward (Hu et al., 2026):

$$\max \quad \sum_{j=1}^{M} R_j, \tag{1}$$

$$\text{s.t.} \quad L_j \le B, \qquad \forall j = 1, \ldots, M, \tag{2}$$

$$P_j \cap P_{j'} = \emptyset, \qquad \forall j \ne j'. \tag{3}$$

where (1) is the reward function, (2) guarantees that the cost of each agent does not exceed its budget, and (3) guarantees that each node is selected by one agent at most. The solution of TOP may result in

imbalanced reward allocation across agents, which will be further verified in the evaluation section. To explicitly incorporate max-min fairness, we formulate fTOP as:

$$\begin{aligned} \max \quad & \min_{a_j \in \mathcal{A}} R_j, \\ \text{s.t.} \quad & (2), (3). \end{aligned} \tag{4}$$

fTOP maximizes the reward of the worst-served agent, thereby promoting fairness among agents under a common travel budget. We develop FLOG (**F**airness-aware **L**atent **O**rienteering based on **G**raph), a graph-based reinforcement learning framework that learns a latent spatial–reward field over nodes and agents to guide fairness-aware and budget-feasible orienteering route construction.

4 Design of FLOG

FLOG exploits the learned latent representations to optimize the max–min reward objective through three components: (1) **Multi-Directional Latent-Field Encoding**, which captures node–node, node-to-agent, and agent–agent interactions; (2) **Fairness-Guided Sequential Route Construction**, which dynamically evaluates budget-feasible agent–node insertion actions according to the current route, reward, and travel-cost states and prioritizes under-rewarded agents; and (3) **Fairness-Preserving Residual Reward Recovery**, which exploits the remaining travel budgets to recover additional reward without excessively degrading the attained fairness.

4.1 Multi-Directional Latent Field Encoding

We represent each routing instance as a heterogeneous graph over candidate nodes and agents. Each node v_i is described by its coordinate and reward, with raw feature $\mathbf{v}_i = [\mathbf{p}_i; r_i]$, while each agent a_j is described by its start location, end location, and budget, with raw feature $\mathbf{a}_j = [s_j; e_j; B]$. Type-specific linear projections map them into a common d-dimensional latent space:

$$\mathbf{h}_{i,0} = W_v \mathbf{v}_i + b_v, \qquad \mathbf{g}_{j,0} = W_a \mathbf{a}_j + b_a. \tag{5}$$

where W_v, W_a and b_v, b_a are learnable projection matrices and bias vectors for node and agent features, respectively, and $\mathbf{h}_{i,0}, \mathbf{g}_{j,0}$ are the corresponding initial embeddings. An L-layer multi-directional attention encoder then models node–node, node-to-agent, and agent–agent interactions. After L layers, FLOG obtains latent node representations $\mathbf{H}_L$ and agent representations $\mathbf{G}_L$, which provide the context for route construction. Detailed encoder architecture is provided in Appendix A.

4.2 Fairness-Guided Sequential Route Construction

Given $\mathbf{H}_L$ and $\mathbf{G}_L$, FLOG constructs routes by sequentially selecting budget-feasible agent–node insertion actions over steps $t = 0, 1, \ldots\ldots$. At step t, $P_{j,t}$, $L_{j,t}$, and $R_{j,t} = \sum_{v_i \in P_{j,t}} r_i$ denote the route, length, and accumulated reward of agent a_j, respectively. Let $v_{i'}$ and $v_{i'+1}$ denote two consecutive nodes of $P_{j,t}$. For an unvisited node v_i, the minimum insertion cost into $P_{j,t}$, which measures the additional travel cost caused by inserting v_i, is:

$$\Delta_{ji,t} = \min \left[d(v_{i'}, v_i) + d(v_i, v_{i'+1}) - d(v_{i'}, v_{i'+1}) \right]. \tag{6}$$

Let (a_j, v_i) denote the action of assigning node v_i to agent a_j with the minimum insertion cost $\Delta_{ji,t}$. The budget-feasible actions are:

$$\mathcal{F}_t = \left\{ (a_j, v_i) \mid v_i \notin \bigcup_{m=1}^{M} P_{m,t},\ r_i > 0,\ L_{j,t} + \Delta_{ji,t} \le B \right\}. \tag{7}$$

For each $(a_j, v_i) \in \mathcal{F}_t$, FLOG follows the scaled query–key compatibility in attention-based neural routing (Kool et al., 2019) and the tanh-based logit clipping in neural combinatorial optimization (Bello et al., 2017) to combine the dynamic agent representation, the latent node representation, and route-aware insertion features into an action score:

$$u_{ji,t} = C \tanh \left(\frac{(W_g \mathbf{g}_{j,t})^\top (W_h \mathbf{h}_{i,L})}{\sqrt{d}} \right) + \phi_c(\mathbf{c}_{ji,t}), \tag{8}$$

where $W_g, W_h \in \mathbb{R}^{d\times d}$ are learnable projection matrices for the dynamic agent and node representations, respectively; $C > 0$ is the tanh-clipping coefficient that restricts the latent compatibility term to $(-C, C)$, limiting excessive score separation and promoting exploration; $\phi_c : \mathbb{R}^3 \to \mathbb{R}$ is a learnable multilayer transformation that maps the action-level feature to a scalar route-aware adjustment. Here, $\mathbf{h}_{i,L}$ is the latent representation of node v_i, while $\mathbf{g}_{j,t}$ is computed from the static agent representation $\mathbf{g}_{j,L}$ and the dynamic routing state $\mathbf{z}_{j,t}$, as defined in Eq.21. The action feature $\mathbf{c}_{ji,t}$ encodes the reward and insertion-cost information of action (a_j, v_i), as defined in Eq.22. Detailed definitions of $\mathbf{z}_{j,t}$, $\mathbf{c}_{ji,t}$, and $\mathbf{g}_{j,L}$ are provided in Appendix A.

To incorporate fairness into sequential construction, we measure the fairness level at step t by

$$R_{\max,t} = \max_{a_j\in\mathcal{A}} R_{j,t}, \quad \text{and} \quad \mathrm{RFR}_t = \frac{\min_{a_j\in\mathcal{A}} R_{j,t}}{R_{\max,t} + \epsilon}, \tag{9}$$

where $R_{j,t}$ are defined above and $R_{\max,t}$ denotes the largest accumulated reward among the agents, and RFR_t denotes the reward fairness metric. To guide fairness-aware route construction and subsequent reward recovery, we introduce a fairness threshold $\rho \in (0, 1]$, which specifies the target reward-fairness level maintained throughout the route construction and recovery process.

Based on Eq.(9), FLOG first prioritizes zero-reward agents that still have feasible actions, and protects the sole feasible positive-reward node of a zero-reward agent from being allocated to another agent. When no such agent exists, FLOG prioritizes agents whose rewards are below $\rho R_{\max,t}$, where ρ is the target fairness threshold. The retained actions form the fairness-filtered action set $\widetilde{\mathcal{F}}_t$. The policy over retained actions is then computed as

$$\pi_{\theta,t}(a_j, v_i) = \frac{\exp(u_{ji,t})}{\sum_{(a_m,v_n)\in\widetilde{\mathcal{F}}_t} \exp(u_{mn,t})}, \qquad (a_j, v_i) \in \widetilde{\mathcal{F}}_t. \tag{10}$$

where $\pi_{\theta,t}(a_j, v_i)$ is the probability of selecting action (a_j, v_i) at step t, and θ denotes all learnable parameters of FLOG. The complete procedure of fairness-guided sequential route construction is summarized in Alg. 1 in Appendix B.

4.3 Fairness-Preserving Residual Reward Recovery

After the route construction, FLOG further exploits the remaining budget to recover reward from unvisited candidate nodes while preserving fairness among agents. In the recovery process, we index the recovery process by $t' = 0, 1, \ldots$, where $t' = 0$ corresponds to the routes, accumulated rewards, and unvisited nodes returned by the construction module.

To prevent residual reward recovery from degrading the fairness attained during route construction, the recovery process reuses the fairness threshold ρ defined in the construction module. We define the corresponding fairness deficit as:

$$G_{t'} = \sum_{j=1}^{M} \max\left\{\rho R_{\max,t'} - R_{j,t'},\, 0\right\}, \tag{11}$$

where the definition of $R_{\max,t'}$ is similar in Eq.(9). For each candidate action $(a_j, v_i) \in \mathcal{F}_{t'}$, let $\mathrm{RFR}^+_{ji,t'}$ and $G^+_{ji,t'}$ denote the fairness ratio and fairness deficit obtained after tentatively inserting v_i into the current route of a_j. We retain only actions that preserve the attained fairness level or reduce the current fairness deficit:

$$\mathcal{A}_{t'} = \left\{(a_j, v_i) \in \mathcal{F}_{t'} \mid \mathrm{RFR}^+_{ji,t'} \geq \rho \text{ if } \mathrm{RFR}_{t'} \geq \rho;\ G^+_{ji,t'} < G_{t'} \text{ if } \mathrm{RFR}_{t'} < \rho\right\}. \tag{12}$$

FLOG selects the action from $\mathcal{A}_{t'}$ with the highest reward efficiency according to:

$$\arg\max_{(a_j,v_i)\in\mathcal{A}_{t'}} \frac{r_i}{\Delta_{ji,t'} + \epsilon}. \tag{13}$$

The procedure of fairness-preserving residual reward recovery is summarized in Alg. 2 in Appendix B.

4.4 FLOG Training and Inference

FLOG is trained with multi-sample policy optimization. For each instance, it generates S candidate rollouts and recovers each deterministically. Each recovered rollout $s \in \{1, \dots, S\}$ is evaluated as:

$$U^{(s)} = \min\left\{\frac{\mathrm{RFR}^{(s)}}{\rho}, 1\right\} + \frac{R_{\mathrm{tot}}^{(s)}}{R_{\mathrm{tot}}^{\max} + \epsilon}, \tag{14}$$

where $R_{\mathrm{tot}}^{\max}$ is the maximum total reward among the S rollouts. The policy is optimized with an advantage-weighted log-likelihood loss using an instance-specific rollout baseline.

During inference, FLOG generates K candidates, including one greedy rollout and $K-1$ sampled rollouts. Each candidate is followed by deterministic recovery. FLOG then selects the candidate with the highest RFR among those whose total reward is at least ρ_R times the best candidate total reward; ties are broken by total reward. Full training and inference details are given in Appendix A.

5 Evaluation

5.1 Experimental Setup

Training and Testing Scales. FLOG is trained on synthetic instances with the number of agents $M \in [5, 10]$ and candidate nodes $N \in [5M, 100]$. Evaluation includes two settings: Normal Testing (NT), which uses problem scales within the training range, and Generalization Testing (GT), which extends M and N beyond the training range to evaluate scalability.

Synthetic Dataset. Each synthetic instance is generated by uniformly sampling $2M + N$ locations in the unit square $[0, 1]^2$. Among them, $2M$ locations are randomly paired to form $\langle s_j, e_j \rangle$ for the M agents, while the remaining N locations serve as candidate nodes. Node rewards are independently sampled from $[0, 1]$. The common budget B is the larger of the mean original budget and $\max_j d(s_j, e_j) + \epsilon$; with original budgets sampled from $[0, 2]$, B is typically close to 1 and at most 2, allowing each agent to travel directly from s_j to e_j.

Real-World Dataset. The real-world-data-based evaluation corresponds to an incident-oriented patrol scenario, in which police officers or UAVs (agents) patrol regions (namely nodes) with high incident risk. We evaluate FLOG using the urban incident datasets in Seattle (Seattle Police Department, 2024) and Los Angeles (Los Angeles Police Department, 2024) and Denver (Mooney, 2014). Detailed dataset statistics and an example incident record are reported in Tables 5 and 6 in Appendix C, respectively. The coordinates of each city are normalized to $[0, 1]^2$ while preserving their spatial distributions. Incidents are clustered into $2M + N$ high-risk locations using K-Means. Among the resulting cluster centers, $2M$ are randomly paired to form $\langle s_j, e_j \rangle$ for the M agents, while the remaining N serve as candidate nodes. Candidate-node rewards are determined by cluster sizes, and the common budget B follows the synthetic setting, i.e., the larger of the mean original budget and the maximum start–end distance plus ϵ.

Baselines. We compare FLOG with six representative baselines spanning learning-based, greedy, classical optimization, and metaheuristic. GRIP Hu et al. (2026): GRIP is a SOTA DRL solution for TOP. GCB (Pan et al., 2024): the researchers extends the greedy cost-benefit OP algorithm (Zhang & Vorobeychik, 2016) to solve TOP. OR-Tools (Furnon & Perron, 2023) and LKH-3 (Helsgaun, 2017) are also implemented as representative classical routing solvers. We implement two forms of GCB, OR-Tools and LKH-3, respectively: GCB(R), OR-Tools(R) and LKH-3(R) maximize the total reward; and GCB(F), OR-Tools(F) and LKH-3(F) maximize the fairness, namely the minimum agent reward. NSGA-II (Gross et al., 2025): NSGA-II is a genetic algorithm to solve the Bi-objective TOP problem of maximizing both the total reward and min reward. For each scale $\langle M, N \rangle$ in both the synthetic data and real-world data of each city, NSGA-II is evaluated on only 10 instances, because evaluating 100 instances would require well over 30,000 seconds at each scale. The other baselines, in contrast, are evaluated on 100 instances.

Evaluation Metrics. 4 metrics are adopted to evaluate the performance: 1) Minimum Reward (min R) is the reward of the least-rewarded agent, directly reflecting the fairness objective of fTOP. 2) Reward Balance Ratio (RFR) is the ratio between the minimum and maximum agent rewards, as defined in Eq.(9), measuring the relative fairness of the collected rewards across agents. 3) Total

Table 1: Execution Time on 10 Synthetic Validation Instances.

Method	$\langle 5, 50\rangle$	$\langle 5, 100\rangle$	$\langle 10, 100\rangle$	$\langle 10, 500\rangle$	$\langle 20, 500\rangle$	$\langle 20, 1000\rangle$
GCB	1.70 s	5.49 s	11.23 s	534.65 s	1,136.77 s	6,933.19 s
OR-Tools	100.00 s	100.00 s	100.00 s	100.01 s	100.00 s	100.02 s
LKH3	6.40 s	13.70 s	26.84 s	226.08 s	426.01 s	1,065.88 s
NSGA-II	3,669.80 s	4,001.23 s	4,992.22 s	8,165.35 s	12,328.40 s	15,647.37 s
GRIP	**0.58 s**	**1.12 s**	**1.43 s**	**21.08 s**	**25.80 s**	**90.83 s**
FLOG	2.18 s	3.74 s	5.39 s	25.92 s	51.12 s	93.52 s

Table 2: Performance on Representative Synthetic Validation Instances.

Type	Method	$\langle 5, 50\rangle$			$\langle 5, 100\rangle$			$\langle 7, 70\rangle$			$\langle 10, 100\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
NT	GCB (R)	20.838	2.038	0.334	34.921	4.441	0.474	31.642	1.597	0.216	48.403	0.903	0.101
	GCB (F)	20.315	2.428	0.412	34.171	4.892	0.541	30.860	2.136	0.300	47.465	1.515	0.173
	OR-Tools (R)	**22.122**	2.702	0.445	**36.702**	5.388	0.575	**33.194**	2.353	0.348	**49.510**	1.030	0.137
	OR-Tools (F)	21.776	3.274	0.572	36.205	5.628	0.613	32.604	3.271	0.502	48.354	2.603	0.351
	LKH (R)	20.751	2.040	0.332	32.312	4.064	0.464	30.905	1.801	0.254	46.953	1.420	0.170
	LKH (F)	19.492	2.678	0.493	30.656	4.680	0.592	29.080	2.576	0.411	44.435	2.328	0.310
	NSGA-II	14.488	2.887	**0.992**	18.983	3.786	**0.994**	20.350	2.849	**0.914**	35.444	3.502	**0.978**
	GRIP	21.082	2.491	0.423	35.578	4.834	0.520	31.706	2.135	0.318	48.114	1.238	0.152
	FLOG	19.914	**3.341**	0.727	34.511	**5.916**	0.745	30.000	**3.543**	0.707	45.535	**3.653**	0.688
		$\langle 5, 1000\rangle$			$\langle 10, 500\rangle$			$\langle 10, 1000\rangle$			$\langle 20, 1000\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
GT	GCB (R)	**136.881**	22.807	0.720	**167.356**	10.425	0.464	**254.840**	18.096	0.566	**419.970**	8.690	0.275
	GCB (F)	135.540	23.432	0.757	164.716	11.640	0.529	251.730	19.565	0.623	413.708	11.057	0.352
	OR-Tools (R)	110.744	16.441	0.599	151.264	10.089	0.504	204.304	13.140	0.483	352.030	9.712	0.381
	OR-Tools (F)	110.389	16.469	0.601	149.361	10.207	0.514	202.090	13.521	0.501	352.030	10.686	0.418
	LKH (R)	72.136	10.397	0.568	111.321	6.666	0.423	136.610	8.278	0.434	247.982	6.428	0.334
	LKH (F)	69.218	11.224	0.665	106.403	7.613	0.520	129.791	9.458	0.539	236.959	7.608	0.423
	NSGA-II	40.728	8.131	**0.994**	59.675	5.921	**0.984**	73.888	7.336	**0.987**	134.367	6.530	**0.953**
	GRIP	116.950	17.223	0.592	162.899	10.144	0.467	233.229	15.555	0.514	404.919	9.122	0.299
	FLOG	136.343	**25.156**	0.848	165.280	**14.016**	0.732	254.398	**22.160**	0.767	412.483	**16.289**	0.671

Reward (R) is the sum rewards collected by all agents, measuring the overall reward retained. 4) Training and Inference/Execution Time is the wall-clock time required for training, where applicable, and for solution generation at test time, measuring the computational efficiency of each method.

Implementation Details. FLOG is trained for 3,000 steps using Adam. We set the fairness threshold to $\rho = 0.65$ and the reward-retention threshold to $\rho_R = 0.90$ in the main experiments. Full model hyperparameters, training settings, and hardware/software details are provided in Appendix D.

5.2 Training and Inference/Execution Time Comparison

Before comparing solution quality, we first evaluate training and inference/execution efficiency. FLOG requires 3.836 hours of training, compared with 5.625 hours for GRIP, with detailed training-time statistics in Appendix E. Table 1 reports execution time on the same 10 instances across six representative synthetic validation scales. GRIP achieves the lowest execution time across all evaluated scales. FLOG remains efficient across scales, with runtime increasing from 2.18 s at $\langle 5, 50\rangle$ to 93.52 s at $\langle 20, 1000\rangle$. On the largest scale, its runtime is close to GRIP and much faster than LKH3, GCB, and NSGA-II. The computational cost of search-based baselines grows more rapidly with problem size. At $\langle 20, 1000\rangle$, LKH3 requires $1{,}065.88$ s, GCB requires $6{,}933.19$ s, and NSGA-II requires $15{,}647.37$ s. OR-Tools uses a fixed 10-s budget per instance, so its approximately 100-s runtime reflects the prescribed search budget rather than natural termination. Overall, FLOG maintains favorable computational scalability while incorporating fairness-aware route construction.

5.3 Performance on Synthetic Dataset

Table 2 reports the representative results on synthetic validation instances under Normal Testing (NT) and Generalization Testing (GT). The complete results over all 14 synthetic validation scales are provided in Appendix F. FLOG achieves the highest minimum agent reward on all 14 scales and substantially improves RFR over all baselines as well as retains competitive total reward.

Normal Testing (NT). FLOG consistently improves the max–min objective, with the advantage becoming more pronounced as the number of agents increases. At $\langle 10, 100\rangle$, FLOG raises the $\min R$ from 1.238 to 3.653 and RFR from 0.152 to 0.688 relative to GRIP, while reducing the total reward by only 5.4%. Although NSGA-II attains a near-perfect RFR of 0.978, its R and $\min R$ are only 35.444 and 3.502, compared with 45.535 and 3.653 for FLOG, indicating that its higher RFR is

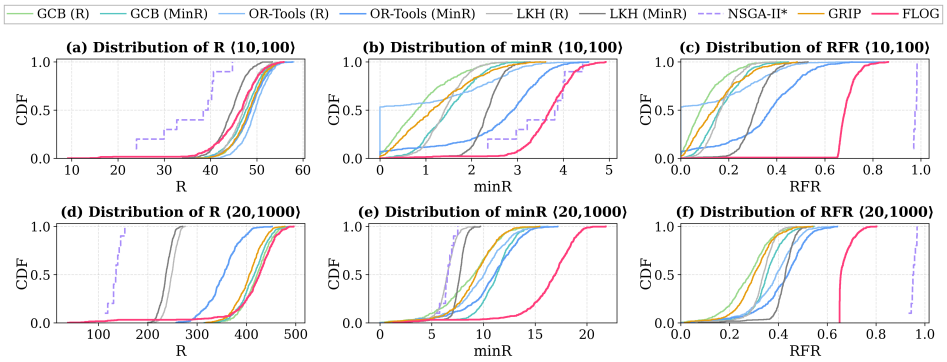

Figure 1: Empirical Distribution of R, $\min R$ and RFR on Synthetic Setting.

accompanied by substantially lower R. For the largest problem scale ($\langle 20, 1000\rangle$), the reward (R) of NSGA-II is only $32.5\% = 134.367/412.483$ of FLOG. These results demonstrate that NSGA-II achieves perfect fairness at the cost of significantly reduced reward.

Generalization Testing (GT). FLOG maintains its advantage beyond the training range, particularly at larger scales. At $\langle 10, 1000\rangle$, FLOG improves the minimum reward from 15.555 to 22.160 and RFR from 0.514 to 0.767 relative to GRIP, while its R of 254.398 is only 0.2% below that of the best baseline GCB(R). At the largest scale, $\langle 20, 1000\rangle$, FLOG further increases the $\min R$ from 9.122 to 16.289 and RFR from 0.299 to 0.671, while also improve R from 404.919 to 412.483.

More Insights. To reveal more insights of these algorithms, Fig. 1 (a)-(f) compare the per-instance distributions of the metrics at $\langle 10, 100\rangle$ and $\langle 20, 1000\rangle$. For total reward R, FLOG remains competitive at $\langle 10, 100\rangle$ and shows the rightmost CDF at $\langle 20, 1000\rangle$. Its advantage is more pronounced for $\min R$, where the FLOG curves in panels (b) and (e) are consistently shifted to the right, indicating stronger worst-agent rewards than the baselines. For RFR, nearly all FLOG instances at $\langle 10, 100\rangle$ and all instances at $\langle 20, 1000\rangle$ remain above approximately 0.65, while the baselines extend further into the low-fairness region. A few unusually low-RFR cases at $\langle 10, 100\rangle$ account for the short CDF segment near zero. NSGA-II achieves high RFR but with markedly lower total reward R. Overall, FLOG consistently improves fairness while maintains strong reward performance.

5.4 Performance on Real-World Dataset

Table 3 reports representative results on Denver, Los Angeles (LA), and Seattle using four selected scales. The complete real-world results over all evaluated scales are provided in Appendix G. Averaged over all evaluated real-world scales, FLOG increases RFR from 0.172 for GRIP to 0.744, while incurring only about a 2.2% loss in total reward.

Normal Testing (NT). FLOG shows a consistent fairness advantage across the three cities. At $\langle 10, 100\rangle$, averaged over Denver, LA, and Seattle, FLOG increases RFR from 0.062 to 0.737 and the $\min R$ from 0.373 to 1.918 relative to GRIP, while reducing total reward R by only about 1.2%. The similar behavior across different city distributions indicates robustness of FLOG across diverse spatial structures of the nodes.

Generalization Testing (GT). FLOG continues to improve both the $\min R$ and RFR on larger real-world instances. At the largest scale, $\langle 20, 1000\rangle$, FLOG increases the average RFR across the three cities from 0.055 to 0.680 and the average $\min R$ from 1.213 to 5.766, while reducing total reward R by only about 2.5% relative to GRIP. As in the synthetic experiments, NSGA-II attains near-perfect RFR ($0.966 - 0.998$) but with substantially lower total reward R: For example, on Seattle at $\langle 20, 1000\rangle$, its R and min Rs are 52.176 and 2.587, compared with 111.691 and 4.492 for FLOG.

Although NSGA-II achieves near-perfect RFR values ($0.966 - -0.998$) in both NT and GT, its total and maximum agent rewards are typically only 30%–50% of FLOG's across all evaluated settings. For example, at $\langle 20, 1000\rangle$, NSGA-II's total reward R is 32.6% of FLOG's on synthetic data (134.367 vs. 412.483) and 40.7%–46.7% on real-world datasets, with maximum agent rewards similarly lower. This indicates that its high RFR mainly reflects a compressed reward scale rather than a better fairness–efficiency trade-off. Across all datasets, FLOG increases the average RFR from 0.235 for GRIP to 0.743, while reducing R by only 1.3% on average. Moreover, while GRIP's RFR falls to 0.05 or below in 12.5% of the evaluated settings, FLOG maintains an RFR of at least 0.659 across all settings and achieves the highest $\min R$ in 91.1% of them. We further report the per-instance

Table 3: Performance Comparison on Representative Real-World Datasets.

City	Method	$\langle 10, 100\rangle$			$\langle 10, 500\rangle$			$\langle 10, 1000\rangle$			$\langle 20, 1000\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
Denver	GCB (R)	25.847	0.057	0.006	91.926	2.699	0.106	118.476	4.828	0.165	**138.309**	0.264	0.010
	GCB (F)	25.735	0.088	0.010	90.152	3.618	0.184	116.733	5.592	0.195	137.709	0.572	0.020
	OR-Tools (R)	**25.865**	0.009	0.002	89.372	4.233	0.257	104.454	4.740	0.229	136.605	1.287	0.065
	OR-Tools (F)	25.843	0.103	0.019	88.146	4.563	0.276	104.441	4.969	0.237	136.605	1.325	0.066
	LKH (R)	25.775	0.151	0.016	67.747	2.660	0.128	73.300	2.895	0.140	102.172	1.665	0.082
	LKH (F)	25.451	0.404	0.045	65.560	3.179	0.159	69.983	3.611	0.196	98.994	2.126	0.112
	NSGA-II	22.359	**2.217**	**0.986**	37.671	3.749	**0.991**	30.017	2.988	**0.992**	54.277	2.673	**0.972**
	GRIP	25.707	0.585	0.095	**92.071**	3.727	0.194	**119.955**	5.315	0.205	137.691	1.588	0.090
	FLOG	25.030	2.097	0.734	85.473	**6.896**	0.709	106.242	**8.263**	0.682	129.036	**4.817**	0.659
LA	GCB (R)	29.627	0.063	0.007	**102.460**	3.266	0.141	**188.333**	8.671	0.226	**208.443**	0.391	0.012
	GCB (F)	29.574	0.084	0.010	100.301	4.438	0.218	185.111	9.980	0.277	207.844	0.811	0.027
	OR-Tools (R)	**29.637**	0.004	0.001	98.128	4.959	0.283	163.010	8.267	0.284	206.044	1.875	0.078
	OR-Tools (F)	29.628	0.062	0.011	96.683	5.288	0.304	163.010	8.780	0.301	206.044	1.936	0.079
	LKH (R)	29.591	0.198	0.020	71.189	3.129	0.171	105.570	4.800	0.187	144.617	2.849	0.122
	LKH (F)	29.239	0.556	0.059	67.963	3.795	0.222	101.013	5.753	0.246	139.486	3.565	0.166
	NSGA-II	25.182	**2.493**	**0.981**	38.885	3.868	**0.991**	46.936	4.673	**0.992**	83.348	4.107	**0.974**
	GRIP	29.528	0.353	0.042	101.649	3.911	0.164	185.354	8.942	0.220	207.128	1.424	0.039
	FLOG	29.385	2.443	0.732	99.171	**8.102**	0.697	181.639	**14.606**	0.686	204.606	**7.989**	0.673
Seattle	GCB (R)	14.480	0.007	0.001	**67.603**	0.911	0.040	81.026	2.356	0.088	**111.828**	0.033	0.001
	GCB (F)	14.475	0.009	0.002	66.619	1.618	0.091	79.315	3.078	0.146	111.786	0.069	0.003
	OR-Tools (R)	**14.481**	0.000	0.000	65.769	2.546	0.210	72.990	2.747	0.185	111.744	0.186	0.010
	OR-Tools (F)	14.480	0.011	0.003	65.414	2.817	0.228	72.990	2.910	0.192	111.744	0.189	0.011
	LKH (R)	14.477	0.021	0.003	54.952	1.777	0.084	55.358	1.900	0.093	93.354	1.235	0.043
	LKH (F)	14.415	0.093	0.015	52.946	2.221	0.109	53.255	2.330	0.119	90.892	1.595	0.059
	NSGA-II	12.727	**1.266**	**0.991**	26.618	2.653	**0.993**	22.371	2.232	**0.996**	52.176	2.587	**0.985**
	GRIP	14.478	0.181	0.050	67.562	2.063	0.124	**81.123**	3.081	0.173	111.816	0.627	0.036
	FLOG	14.463	1.215	0.744	67.048	**5.755**	0.759	79.024	**6.597**	0.728	111.691	**4.492**	0.707

Table 4: Ablation Study on FLOG Variants.

Method	$\langle 5, 50\rangle$			$\langle 5, 100\rangle$			$\langle 10, 100\rangle$			$\langle 10, 500\rangle$		
	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
FLOG-NoFGC	20.188	3.313	0.695	32.114	5.215	0.679	46.004	3.411	0.577	124.470	8.650	0.530
FLOG-NoRR	8.690	1.479	**0.776**	9.177	1.547	**0.766**	18.231	1.547	**0.745**	19.288	1.651	**0.738**
FLOG-ROR	**20.826**	3.092	0.592	**35.479**	5.656	0.660	**47.859**	2.990	0.435	**169.313**	13.464	0.659
FLOG	19.914	**3.341**	0.727	34.511	**5.916**	0.745	45.535	**3.653**	0.688	165.280	**14.016**	0.732

empirical distributions of R, $\min R$, and RFR on real-world settings in Appendix G.1, which further corroborates that FLOG could optimize fairness and maintain significant reward performance.

5.5 Ablation Study And Sensitivity Analysis

Table 4 reports the ablation results on four representative synthetic scales. We consider three variants: FLOG-NoFGC removes the Fairness-Guided Construction mechanism described in Sec. 4.2, FLOG-NoRR disables the Residual Reward Recovery described in Sec. 4.3, and FLOG-ROR replaces the Fairness-Preserving Recovery in Sec. 4.3 with reward-only recovery. Removing fairness-guided construction increasingly degrades fairness as the problem scale grows; at $\langle 10, 500\rangle$, FLOG-NoFGC reduces the $\min R$ from 14.016 to 8.650 and RFR from 0.732 to 0.530. FLOG-NoRR retains high RFR but collects substantially less reward, with total and min Rs respectively dropping to 19.288 and 1.651. FLOG-ROR Replaces the fairness-preserving recovery with reward-only recovery, which increases R from 165.280 to 169.313 but reduces the $\min R$ to 13.464 and RFR to 0.659. These results show that fairness-guided construction, residual reward recovery, and fairness preservation play distinct but complementary roles in FLOG.

Additional sensitivity analysis on the fairness threshold ρ shows that ρ provides an effective control over the fairness–reward trade-off, which is provided in Appendix H.

6 Conclusion

In this work, we study the Fair Team Orienteering Problem (fTOP) with a max–min reward objective and propose FLOG, a graph-based reinforcement learning framework for fTOP. FLOG exploits the learned latent representations to optimize the max–min reward objective through three components: Multi-Directional Latent-Field Encoding, Fairness-Guided Sequential Route Construction and Fairness-Preserving Residual Reward Recovery. Extensive experiments on synthetic and real-world datasets demonstrate that FLOG consistently improves reward fairness while largely preserving overall reward. Compared with the state-of-the-art TOP solver GRIP, FLOG increases the average RFR from 0.235 to 0.743 with only a 1.3% average reduction in R; Moreover, it maintains an RFR

of at least 0.659 across all evaluated settings and achieves the highest $\min R$ in 91.1% of them. To summarize, FLOG could achieve significant fairness as well as maintain remarkable reward.

REFERENCES

Sinan Aral and Paramveer S Dhillon. Social influence maximization under empirical influence models. Nature human behaviour, 2(6):375–382, 2018.

Fuat Basık, Buğra Gedik, Hakan Ferhatosmanoğlu, and Kun-Lung Wu. Fair task allocation in crowdsourced delivery. IEEE Transactions on Services Computing, 14(4):1040–1053, 2018.

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In International Conference on Learning Representations, 2017.

MA Beyazit, AK Erenoglu, and A Tascikaraoglu. Fairness and equity in electric vehicle charging with mobile charging stations. IEEE Transactions on Industry Applications, 61(1):1280–1291, 2024.

Chao Bian, Chao Feng, Chao Qian, and Yang Yu. An efficient evolutionary algorithm for subset selection with general cost constraints. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 3267–3274, 2020.

Lindsley G Boiney. When efficient is insufficient: Fairness in decisions affecting a group. Management science, 41(9):1523–1537, 1995.

Sylvain Boussier, Dominique Feillet, and Michel Gendreau. An exact algorithm for team orienteering problems. 4or, 5(3):211–230, 2007.

Vincent Boyer, Gabriela Sánchez-Yepez, and M Angélica Salazar-Aguilar. Branch and cut for the team orienteering problem with nonidentical agents and balanced score. Expert Systems with Applications, 307:131007, 2026.

Martim Brandao, Marina Jirotka, Helena Webb, and Paul Luff. Fair navigation planning: A resource for characterizing and designing fairness in mobile robots. Artificial Intelligence, 282:103259, 2020.

Alice M. Brawley and Cynthia L.S. Pury. Work experiences on mturk: Job satisfaction, turnover, and information sharing. Computers in Human Behavior, 54:531–546, 2016. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2015.08.031. URL https://www.sciencedirect.com/science/article/pii/S0747563215301072.

Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 199–208, 2009.

Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. Robust influence maximization. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 795–804, 2016.

Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. Constructive approximation, 13:57–98, 1997.

Haitao Fan, Mengzhe Hei, Zuyu Chen, Yijia Xing, Yujiao Hu, Deke Guo, Xin Zhang, Xiang Zhao, and Yan Pan. E-bus-based standby energy sharing for evs in the event of large-scale grid outage. IEEE Internet of Things Journal, 11(23):37885–37898, 2024.

Vincent Furnon and Laurent Perron. Or-tools. Google Optimization Tools, 2023.

Songqun Gao, Zanxi Ruan, Patrick Floor, Marco Roveri, Luigi Palopoli, and Daniele Fontanelli. Learning to solve orienteering problem with time windows and variable profits. 2026.

Raimundo Gross, Elizabeth Montero, Nicolás Gálvez Ramírez, and Adriana Menchaca-Méndez. Generalising the team orienteering problem through multi-constraining and multi-objectives. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '25 Companion, pp. 219–222. ACM, 2025. doi: 10.1145/3712255.3726719.

Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. European journal of operational research, 255(2):315–332, 2016.

Anupam Gupta and Roie Levin. The online submodular cover problem. In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1525–1537. SIAM, 2020.

Keld Helsgaun. An extension of the lin–kernighan–helsgaun tsp solver for constrained traveling salesman and vehicle routing problems. Technical report, Roskilde University, Roskilde, Denmark, 2017.

Hadi Hosseini, Shivika Narang, and Tomasz Was. Fair distribution of delivery orders. Artificial Intelligence, 347:104389, 2025.

Yujiao Hu, Zuyu Chen, MengJie Lee, Jinchao Chen, Meng Shen, Hailun Zhang, Wei Li, and Yan Pan. Grip: Latent field-guided graph policy for budget-constrained multi-agent routing. In aaai (ed.), Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2026), pp. 29495–29503, 2026.

Hargyo TN Ignatius and Rami Bahsoon. Equity, equality, and need: digital twin approach for fairness-aware task assignment of heterogeneous crowdsourced logistics. IEEE Transactions on Computational Social Systems, 11(3):3420–3431, 2023.

Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. Advances in Neural Information Processing Systems, 30, 2017.

C Peter Keller. Algorithms to solve the orienteering problem: A comparison. European Journal of Operational Research, 41(2):224–231, 1989.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In International Conference on Learning Representations, 2019.

Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. Online influence maximization. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 645–654, 2015.

Yafei Li, Huiling Li, Baolong Mei, Xin Huang, Jianliang Xu, and Mingliang Xu. Fairness-guaranteed task assignment for crowdsourced mobility services. IEEE Transactions on Mobile Computing, 23(5):5385–5400, 2024. doi: 10.1109/TMC.2023.3310591.

Laura Londoño, Juana Valeria Hurtado, Nora Hertz, Philipp Kellmeyer, Silja Voeneky, and Abhinav Valada. Fairness and bias in robot learning. Proceedings of the IEEE, 112(4):305–330, 2024.

Los Angeles Police Department. Crime in los angeles. https://www.kaggle.com/datasets/cityofLA/crime-in-los-angeles, 2024. Accessed: 2024-10-01.

Ariella Mansfield, Sandeep Manjanna, Douglas G Macharet, and M Ani Hsieh. Multi-robot scheduling for environmental monitoring as a team orienteering problem. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6398–6404. IEEE, 2021.

Patrick Mooney. Denver crime data. https://www.kaggle.com/datasets/paultimothymooney/denver-crime-data, 2014. Accessed: 2024-10-01.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. SIAM journal on computing, 24(2):227–234, 1995.

Yan Pan, Junhui Gao, Jiashun Duan, Juan Shi, Bin Guo, Yunji Liang, and Yujiao Hu. Pioneering cooperative air-ground instant delivery using uavs and crowdsourced couriers. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 8(4):1–26, 2024.

Chao Qian, Jing-Cheng Shi, Yang Yu, and Ke Tang. On subset selection with general cost constraints. In IJCAI, volume 17, pp. 2613–2619, 2017.

Gabriela Sánchez-Yepez, M Angélica Salazar-Aguilar, and Pamela J Palomo-Martínez. Team orienteering problem with nonidentical agents and balanced score. International Journal of Production Research, 61(23):7957–7971, 2023.

Seattle Police Department. Seattle police department 911 incident response. https://www.kaggle.com/datasets/sohier/seattle-police-department-911-incident-response, 2024. Accessed: 2024-10-02.

Abhay Sobhanan, Hadi Charkhgard, and Iman Dayarian. Express: Equity-driven workload allocation for crowdsourced last-mile delivery. Production and Operations Management, pp. 10591478261425875, 2026.

Yingjie Wang, Dihong Luo, Haojun Teng, Peiyong Duan, Yang Gao, Haijing Zhang, and Zhipeng Cai. Multi-space crowd sensing task allocation: A dynamic co-optimization framework with fairness-aware reinforcement learning. IEEE Transactions on Mobile Computing, 2025.

Wenzheng Xu, Zichuan Xu, Jian Peng, Weifa Liang, Tang Liu, Xiaohua Jia, and Sajal K Das. Approximation algorithms for the team orienteering problem. In IEEE INFOCOM 2020-IEEE conference on computer communications, pp. 1389–1398. IEEE, 2020.

Wenzheng Xu, Weifa Liang, Zichuan Xu, Jian Peng, Dezhong Peng, Tang Liu, Xiaohua Jia, and Sajal K. Das. Approximation algorithms for the generalized team orienteering problem and its applications. IEEE/ACM Transactions on Networking, 29(1):176–189, 2021. doi: 10.1109/TNET.2020.3027434.

Haifeng Zhang and Yevgeniy Vorobeychik. Submodular optimization with routing constraints. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 819–826, 2016.

Yan Zhao, Kai Zheng, Jiannan Guo, Bin Yang, Torben Bach Pedersen, and Christian S Jensen. Fairness-aware task assignment in spatial crowdsourcing: Game-theoretic approaches. In 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 265–276. IEEE, 2021.

Appendix

A Additional Method Details

A.1 Details of Multi-Directional Latent Field Encoding

We construct a heterogeneous interaction graph consisting of candidate nodes and agents. Each candidate node v_i is characterized by its spatial coordinate $\mathbf{p}_i \in \mathbb{R}^2$ and reward r_i, with raw feature

$$\mathbf{v}_i = [\mathbf{p}_i; r_i] \in \mathbb{R}^3. \tag{15}$$

Each agent a_j is characterized by its start location $s_j \in \mathbb{R}^2$, end location $e_j \in \mathbb{R}^2$, and common budget B, with raw feature

$$\mathbf{a}_j = [s_j; e_j; B] \in \mathbb{R}^5. \tag{16}$$

The node and agent features are mapped into a common d-dimensional latent space by type-specific linear projections:

$$\mathbf{h}_{i,0} = W_v \mathbf{v}_i + b_v, \qquad \mathbf{g}_{j,0} = W_a \mathbf{a}_j + b_a. \tag{17}$$

Here, $W_v \in \mathbb{R}^{d\times 3}$ and $W_a \in \mathbb{R}^{d\times 5}$ are learnable projection matrices for node and agent features, respectively; $b_v, b_a \in \mathbb{R}^d$ are the corresponding learnable bias vectors. The resulting $\mathbf{h}_{i,0}$ and $\mathbf{g}_{j,0}$ are the initial node and agent embeddings.

Based on these initial embeddings, FLOG applies an L-layer multi-directional attention encoder to model three types of interactions. First, node-to-node attention captures spatial and reward dependencies among candidate nodes. Second, node-to-agent attention allows each agent to aggregate node-level spatial and reward information. Third, agent-to-agent attention captures inter-agent dependencies under the shared travel budget. After each layer, node and agent representations are updated with residual connections, normalization, and feed-forward transformations.

After L layers, the encoder outputs latent node representations

$$\mathbf{H}_L = \{\mathbf{h}_{1,L}, \ldots, \mathbf{h}_{N,L}\} \tag{18}$$

and latent agent representations

$$\mathbf{G}_L = \{\mathbf{g}_{1,L}, \ldots, \mathbf{g}_{M,L}\}. \tag{19}$$

These representations integrate spatial, reward, and inter-entity interaction information, and are used as the static context for the subsequent fairness-guided route construction module.

A.2 Details of Fairness-Guided Sequential Route Construction

This section provides the detailed definitions omitted from Sec. 4.2.

Dynamic agent state. For each feasible action $(a_j, v_i) \in \mathcal{F}_t$, FLOG represents the current reward and routing status of agent a_j by a dynamic state:

$$\mathbf{z}_{j,t} = \left[\frac{R_{j,t}}{\max\left\{\sum_{m=1}^{M} R_{m,t}, 1\right\}}, \frac{B - L_{j,t}}{B}, \frac{L_{j,t}}{B}, \mathbb{I}[R_{j,t} = 0], \frac{|P_{j,t}|}{N} \right]^\top. \tag{20}$$

where $\mathbb{I}[\cdot]$ equals 1 if the condition holds and 0 otherwise. The five components represent the relative accumulated reward, remaining budget ratio, consumed budget ratio, zero-reward indicator, and route-progress ratio of agent a_j, respectively. The dynamic state is mapped into the latent space and added to the static agent representation:

$$\mathbf{g}_{j,t} = \mathbf{g}_{j,L} + \phi_z(\mathbf{z}_{j,t}), \tag{21}$$

where $\phi_z : \mathbb{R}^5 \to \mathbb{R}^d$ is a learnable transformation.

Action-level feature. For each feasible action $(a_j, v_i) \in \mathcal{F}_t$, FLOG encodes its immediate reward and routing consequence into an action-level feature:

$$\mathbf{c}_{ji,t} = \left[\frac{r_i}{r_{\max} + \epsilon}, \frac{\Delta_{ji,t}}{B}, \frac{B - L_{j,t} - \Delta_{ji,t}}{B} \right]^\top. \tag{22}$$

where $r_{\max} = \max_{v_i \in \mathcal{V}} r_i$, $\epsilon > 0$ is a small constant for numerical stability, $\Delta_{ji,t}/B$ is the normalized insertion cost, and $(B - L_{j,t} - \Delta_{ji,t})/B$ is the remaining budget ratio after insertion.

Fairness filtering. The zero-reward agent set is defined as

$$\mathcal{A}_{z,t} = \{a_j \in \mathcal{A} \mid R_{j,t} = 0,\ \exists\, v_i : (a_j, v_i) \in \mathcal{F}_t\}. \tag{23}$$

where $\mathcal{A}_{z,t}$ contains zero-reward agents that still have at least one feasible action. When $\mathcal{A}_{z,t} \neq \emptyset$, FLOG retains feasible actions associated with agents in $\mathcal{A}_{z,t}$ and protects the sole feasible positive-reward node of a zero-reward agent from being allocated to another agent.

When $\mathcal{A}_{z,t} = \emptyset$, FLOG defines the deficit-agent set as

$$\mathcal{A}_{d,t} = \{a_j \in \mathcal{A} \mid R_{j,t} < \rho R_{\max,t}\}. \tag{24}$$

where $\mathcal{A}_{d,t}$ contains agents whose accumulated rewards remain below the target fairness level. The actions retained after zero-reward prioritization, unique-node protection, or deficit-agent filtering form the fairness-filtered action set $\widetilde{\mathcal{F}}_t \subseteq \mathcal{F}_t$.

Algorithm 1: Fairness-Guided Sequential Route Construction

Input: Latent representations $\mathbf{H}_L, \mathbf{G}_L$; rewards $\{r_i\}$; start and end locations $\{s_j, e_j\}$; budget B; fairness threshold ρ
Output: Constructed routes $\{P_j\}$

1 Initialize $P_{j,0} \leftarrow \langle s_j, e_j \rangle$, $L_{j,0} \leftarrow d(s_j, e_j)$, and $R_{j,0} \leftarrow 0$ for all $a_j \in \mathcal{A}$; set $\mathcal{V}_0^- \leftarrow \mathcal{V}$ and $t \leftarrow 0$;
2 Construct $\mathcal{F}_t$ via Eqs.(6)–(7) and compute RFR_t via Eq.(9);
3 **while** $\mathcal{F}_t \neq \emptyset$ **and** $\mathrm{RFR}_t < \rho$ **do**
4 Compute $\mathbf{g}_{j,t}$ and $u_{ji,t}$ via Eqs.(20)–(8);
5 Construct $\widetilde{\mathcal{F}}_t$ using Eqs.(23)–(24) and the unique-node protection rule;
6 Compute $\pi_{\theta,t}$ via Eq.(10) and select (a_j, v_i);
7 Insert v_i into $P_{j,t}$ at the position attaining $\Delta_{ji,t}$;
8 Update $P_{j,t+1}$, $L_{j,t+1}$, $R_{j,t+1}$, and $\mathcal{V}_{t+1}^-$;
9 $t \leftarrow t+1$ and reconstruct $\mathcal{F}_t$ and RFR_t;
10 **return** $\{P_j\}_{j=1}^M$;

Fairness filtering. The zero-reward agent set is defined as

$$\mathcal{A}_{z,t} = \{a_j \in \mathcal{A} \mid R_{j,t} = 0,\ \exists\, v_i : (a_j, v_i) \in \mathcal{F}_t\}. \tag{25}$$

where $\mathcal{A}_{z,t}$ contains zero-reward agents that still have at least one feasible action. When $\mathcal{A}_{z,t} \neq \emptyset$, FLOG retains feasible actions associated with agents in $\mathcal{A}_{z,t}$ and protects the sole feasible positive-reward node of a zero-reward agent from being allocated to another agent.

When $\mathcal{A}_{z,t} = \emptyset$, FLOG defines the deficit-agent set as

$$\mathcal{A}_{d,t} = \{a_j \in \mathcal{A} \mid R_{j,t} < \rho R_{\max,t}\}. \tag{26}$$

where $\mathcal{A}_{d,t}$ contains agents whose accumulated rewards remain below the target fairness level. The actions retained after zero-reward prioritization, unique-node protection, or deficit-agent filtering form the fairness-filtered action set $\widetilde{\mathcal{F}}_t \subseteq \mathcal{F}_t$.

B Algorithm Details

B.1 Fairness-Guided Sequential Route Construction

To summarize, the sequential route construction procedure is presented in Algorithm 1.The algorithm takes the latent node and agent representations, node rewards, agent start and end locations, travel budget, and fairness threshold as input, and outputs the constructed routes for all agents. Lines 1–2 initialize the routes and construct the initial feasible actions. At each decision step, Lines 4–6 update the route-aware action scores, apply fairness-guided filtering, and select an agent–node action according to the learned policy. Lines 7–8 perform the minimum-cost insertion and update the selected agent's route, length, accumulated reward, and the unvisited-node set. Line 9 then reconstructs the feasible actions and fairness level for the next step, and the procedure terminates when no feasible action remains or $\mathrm{RFR}_t \geq \rho$.

B.2 Fairness-Preserving Residual Reward Recovery

To summarize, the recovery procedure in this recovery process is presented in Algorithm 2. The algorithm takes the routes constructed in the construction module, node rewards, travel budget, and fairness threshold as input, and outputs the recovered routes for all agents. Lines 1–2 initialize the recovery process from the route construction solution and construct the budget-feasible and fairness-admissible recovery actions. At each recovery step, Lines 4–6 select the most reward-efficient admissible insertion, apply it at the position attaining the minimum insertion cost, and update the corresponding route, reward, and unvisited-node states. Line 7 then reconstructs the recovery action sets from the updated solution, and the procedure terminates when no admissible insertion remains.

B.3 Training and Inference Details

For rollout s, the advantage is computed as

$$A^{(s)} = U^{(s)} - \frac{1}{S}\sum_{s'=1}^{S} U^{(s')}. \tag{27}$$

Algorithm 2: Fairness-Preserving Residual Reward Recovery

Input: Routes $\{P_j\}$ from the construction module; rewards $\{r_i\}$; budget B; fairness threshold ρ
Output: Recovered routes $\{P_j\}$

Initialize $t' \leftarrow 0$ and construct $\mathcal{V}^-_{t'}$;
Construct $\mathcal{F}_{t'}$ as in Eqs.(6)–(7) and $\mathcal{A}_{t'}$ via Eq.12;
while $\mathcal{A}_{t'} \neq \emptyset$ **do**
  Select (a_j, v_i) via Eq.13;
  Insert v_i into $P_{j,t'}$ at the position attaining $\Delta_{ji,t'}$;
  Update $P_{j,t'+1}$, $L_{j,t'+1}$, $R_{j,t'+1}$, and $\mathcal{V}^-_{t'+1}$;
  $t' \leftarrow t' + 1$ and reconstruct $\mathcal{F}_{t'}$ and $\mathcal{A}_{t'}$;
return $\{P_j\}_{j=1}^{M}$;

Table 5: Basic Information of Crime Datasets.

City	Dataset	# Records	Period	Area (km^2)
Seattle	Crime Dataset	308,687	2009–2017	217.12
Denver	Crime Dataset	509,646	2016–2021	396.48
Los Angeles	Crime Dataset	220,722	2020–2021	1215.97

The policy is optimized by

$$\mathcal{L}(\theta) = -\frac{1}{S}\sum_{s=1}^{S} A^{(s)} \sum_{t} \log \pi_{\theta,t}(a_j, v_i). \tag{28}$$

At inference time, candidate k is retained only if

$$R^{(k)}_{\text{tot}} \geq \rho_R R^{\max}_{\text{cand}}, \tag{29}$$

where $R^{\max}_{\text{cand}}$ is the highest R among the K candidates. Among the retained candidates, FLOG selects the one with the highest RFR and uses R to break ties.

C Real-world Dataset Details

Table 5 summarizes the real-world incident datasets used in our evaluation, and Table 6 provides a representative incident record. The three datasets differ substantially in spatial scale, temporal span, and incident density. Denver contains the largest number of records ($509{,}646$), Seattle covers a longer period from 2009 to 2017 with $308{,}687$ records, while Los Angeles has the largest geographic area (1215.97 km^2) but a shorter period from 2020 to 2021. These differences lead to heterogeneous spatial and reward distributions after clustering.

In our evaluation, incident locations are converted into candidate patrol nodes through K-Means clustering, and the cluster sizes are used as node rewards. Therefore, unlike the synthetic instances with uniformly sampled locations and rewards, the real-world datasets introduce non-uniform spatial concentrations and city-specific reward patterns. This setting complements the synthetic validation and tests whether FLOG can preserve fairness under more realistic incident-oriented patrol distributions.

D Implementation Details

FLOG uses a 128-dimensional embedding with two attention layers, eight attention heads, and a feed-forward hidden dimension of 256. The model is trained for 3,000 optimization steps using the Adam optimizer with a learning rate of 5×10^{-5} and a batch size of 8. For multi-sample training, we use $S = 12$ sampled solutions per training instance. The fairness threshold for fairness-guided route construction is set to $\rho = 0.65$, while the reward-retention threshold for candidate selection is set to $\rho_R = 0.90$.

All experiments are conducted on a server equipped with an NVIDIA RTX 3090 GPU with 24 GB memory, 14 vCPUs of an Intel Xeon Gold 6330 CPU at 2.00 GHz, and 60 GB memory. The software environment is Ubuntu 22.04, Python 3.12, PyTorch 2.5.1, and CUDA 12.4.

Table 6: Example of a Crime Record.

Incident ID	Offense Type	Category	Occurrence Time	Coordinates	Is Crime	Is Traffic
20216009452	theft-other	larceny	2021-03-22 12:51:00	(-104.9692994, 39.7648619)	Yes	No

Table 7: Training Efficiency Comparison between GRIP and FLOG.

Method	Training Time (h) ↓	Time/Step (s) ↓
GRIP	5.625	6.750
FLOG	**3.836**	**4.603**

E Training Efficiency

As shown in Table 7, FLOG requires 3.836 hours of training, compared with 5.625 hours for GRIP, corresponding to a 31.8% reduction in training time and approximately a $1.47\times$ speedup. This gain mainly comes from FLOG's direct budget-feasible route construction mechanism (Section 4.2), which avoids the pruning and reinsertion procedures required by GRIP, reducing the per-step training time from 6.750 s to 4.603 s.

F Complete Synthetic Validation Results

Table 8 reports the complete synthetic validation results over all 14 evaluated scales. Compared with the representative results in the main text, this table further shows that the advantage of FLOG is not limited to a few selected scales. Across both normal-testing and generalization-testing settings, FLOG consistently improves the worst-agent reward and maintains high RFR, while keeping R competitive with reward-oriented baselines.

The complete results also reveal how the fairness–efficiency trade-off changes with problem scale. For example, at $\langle 5, 300\rangle$, FLOG achieves the highest R (70.198), the highest $\min R$ (12.583), and a substantially higher RFR (0.804) than GRIP (0.605). At $\langle 20, 500\rangle$, where reward-oriented baselines suffer severe imbalance, FLOG increases $\min R$ from 2.668 for GRIP to 8.915 and improves RFR from 0.126 to 0.661. These results indicate that FLOG remains effective when the number of agents and candidate nodes increase beyond the training range.

The table also clarifies the behavior of NSGA-II. Although NSGA-II often obtains the highest RFR, its total reward is much lower than that of FLOG and other practical baselines. For instance, at $\langle 5, 1000\rangle$, NSGA-II obtains an RFR of 0.994 but only 40.728 total reward, whereas FLOG obtains 136.343 total reward and a higher $\min R$ of 25.156. This confirms that a high RFR alone does not necessarily indicate a better solution, since it may be achieved by compressing the overall reward scale. Overall, the complete synthetic results support the main-text conclusion that FLOG improves fairness mainly by increasing the worst-agent reward rather than by simply reducing all agents' rewards.

G Complete Real-World Results

Tables 9–11 report the complete real-world results on Denver, LA, and Seattle, respectively. These results provide city-level evidence for the robustness observed in the main text. Across the three cities, FLOG consistently improves $\min R$ and RFR over GRIP and other reward-oriented baselines, even though the cities have different incident distributions and reward concentrations.

The improvement is especially clear in large-scale generalization settings. On Denver at $\langle 20, 1000\rangle$, FLOG improves $\min R$ from 1.588 for GRIP to 4.817 and RFR from 0.090 to 0.659. On LA at the same scale, FLOG improves $\min R$ from 1.424 to 7.989 and RFR from 0.039 to 0.673. On Seattle, FLOG improves $\min R$ from 0.627 to 4.492 and RFR from 0.036 to 0.707. These city-level results show that FLOG's fairness gain is consistent rather than driven by a single dataset.

Meanwhile, reward-oriented baselines often retain high total reward but leave some agents with extremely low rewards, especially on larger scales. NSGA-II again obtains very high RFR values, but its total reward is substantially lower than FLOG. For example, on Seattle at $\langle 20, 1000\rangle$, NSGA-II obtains $R = 52.176$, whereas FLOG obtains $R = 111.691$ while also achieving a higher $\min R$.

Table 8: Performance on Synthetic Validation Instances.

Type	Method	$\langle 5, 50\rangle$			$\langle 5, 100\rangle$			$\langle 7, 70\rangle$			$\langle 10, 100\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
NT	GCB (R)	20.838	2.038	0.334	34.921	4.441	0.474	31.642	1.597	0.216	48.403	0.903	0.101
	GCB (F)	20.315	2.428	0.412	34.171	4.892	0.541	30.860	2.136	0.300	47.465	1.515	0.173
	OR-Tools (R)	**22.122**	2.702	0.445	**36.702**	5.388	0.575	**33.194**	2.353	0.348	**49.510**	1.030	0.137
	OR-Tools (F)	21.776	3.274	0.572	36.205	5.628	0.613	32.604	3.271	0.502	48.354	2.603	0.351
	LKH (R)	20.751	2.040	0.332	32.312	4.064	0.464	30.905	1.801	0.254	46.953	1.420	0.170
	LKH (F)	19.492	2.678	0.493	30.656	4.680	0.592	29.080	2.576	0.411	44.435	2.328	0.310
	NSGA-II	14.488	2.887	**0.992**	18.983	3.786	**0.994**	20.350	2.849	**0.914**	35.444	3.502	**0.978**
	GRIP	21.082	2.491	0.423	35.578	4.834	0.520	31.706	2.135	0.318	48.114	1.238	0.152
	FLOG	19.914	**3.341**	0.727	34.511	**5.916**	0.745	30.000	**3.543**	0.707	45.535	**3.653**	0.688

Type	Method	$\langle 5, 200\rangle$			$\langle 5, 300\rangle$			$\langle 5, 500\rangle$			$\langle 5, 1000\rangle$			$\langle 10, 200\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
GT	GCB (R)	54.403	7.899	0.575	69.537	10.647	0.628	94.170	15.058	0.674	**136.881**	22.807	0.720	86.846	3.493	0.260
	GCB (F)	53.560	8.429	0.633	68.679	11.236	0.680	93.084	15.767	0.723	135.540	23.432	0.757	84.842	4.526	0.341
	OR-Tools (R)	54.082	8.354	0.628	67.031	10.515	0.646	85.401	13.443	0.652	110.744	16.441	0.599	**91.132**	5.766	0.467
	OR-Tools (F)	53.651	8.530	0.649	66.698	10.619	0.657	85.092	13.506	0.658	110.389	16.469	0.601	89.120	6.424	0.532
	LKH (R)	43.799	6.100	0.539	51.175	7.201	0.543	60.204	8.626	0.557	72.136	10.397	0.568	74.225	3.809	0.338
	LKH (F)	41.976	6.644	0.635	48.950	7.856	0.650	57.751	9.296	0.651	69.218	11.224	0.665	70.138	4.673	0.454
	NSGA-II	30.158	6.025	**0.997**	29.721	5.936	**0.995**	40.352	8.064	**0.998**	40.728	8.131	**0.994**	46.357	4.598	**0.985**
	GRIP	53.750	7.936	0.589	66.767	10.009	0.605	86.364	13.206	0.625	116.950	17.223	0.592	87.143	3.878	0.300
	FLOG	**54.614**	**9.638**	0.780	**70.198**	**12.583**	0.804	**95.006**	**17.254**	0.825	136.343	**25.156**	0.848	83.722	**6.770**	0.685

Type	Method	$\langle 10, 300\rangle$			$\langle 10, 500\rangle$			$\langle 10, 1000\rangle$			$\langle 20, 500\rangle$			$\langle 20, 1000\rangle$		
		R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR	R	$\min R$	RFR
GT	GCB (R)	**117.899**	6.119	0.363	**167.356**	10.425	0.464	**254.840**	18.096	0.566	**241.840**	2.165	0.101	**419.970**	8.690	0.275
	GCB (F)	115.742	7.300	0.441	164.716	11.640	0.529	251.730	19.565	0.623	238.078	3.654	0.169	413.708	11.057	0.352
	OR-Tools (R)	117.721	7.843	0.502	151.264	10.089	0.504	204.304	13.140	0.483	227.294	4.306	0.247	352.030	9.712	0.381
	OR-Tools (F)	115.226	8.218	0.535	149.361	10.207	0.514	202.090	13.521	0.501	222.494	6.369	0.370	352.030	10.686	0.418
	LKH (R)	90.963	5.132	0.388	111.321	6.666	0.423	136.610	8.278	0.434	186.504	4.261	0.277	247.982	6.428	0.334
	LKH (F)	86.271	6.042	0.498	106.403	7.613	0.520	129.791	9.458	0.539	178.044	5.314	0.371	236.959	7.608	0.423
	NSGA-II	53.376	5.284	**0.970**	59.675	5.921	**0.984**	73.888	7.336	**0.987**	106.254	5.161	**0.947**	134.367	6.530	**0.953**
	GRIP	117.714	6.192	0.375	162.899	10.144	0.467	233.229	15.555	0.514	236.496	2.668	0.126	404.919	9.122	0.299
	FLOG	115.757	**9.552**	0.703	165.280	**14.016**	0.732	254.398	**22.160**	0.767	231.981	**8.915**	0.661	412.483	**16.289**	0.671

Therefore, the complete real-world results further confirm that FLOG achieves a more practical fairness–efficiency trade-off under heterogeneous urban incident distributions.

G.1 Per-Instance Distributions on Real-World Datasets

To further examine the per-instance behavior of different methods on real-world datasets, we report the empirical cumulative distribution functions (ECDFs) of total reward R, minimum agent reward $\min R$, and reward fairness ratio (RFR) on two representative scales: $\langle 10, 100\rangle$ and $\langle 20, 1000\rangle$. A curve shifted to the right indicates that the corresponding method achieves larger metric values on more instances.

As shown in Figures 2 and 3, FLOG consistently shifts the distributions of $\min R$ and RFR to the right across Denver, LA, and Seattle, indicating stronger worst-agent reward and more balanced reward allocation on most instances. At the normal-testing scale $\langle 10, 100\rangle$, FLOG maintains the distributions of R close to the reward-oriented baselines, while clearly improving the distributions of $\min R$ and RFR. This shows that the fairness improvement is not obtained by substantially sacrificing total reward.

The advantage becomes more evident at the larger generalization scale $\langle 20, 1000\rangle$. FLOG remains competitive in R and achieves the strongest $\min R$ distributions among most practical baselines. In contrast, GRIP and classical reward-oriented solvers often obtain high R but exhibit much lower RFR distributions, indicating that their rewards are unevenly allocated among agents. NSGA-II achieves near-perfect RFR on many instances, but its total-reward distributions are substantially shifted to the left. This suggests that its high RFR is mainly obtained by compressing the reward scale rather than achieving a better fairness–efficiency trade-off. Overall, the ECDF results further support that FLOG improves agent-level fairness while preserving strong reward efficiency on real-world datasets.

Table 9: Performance Comparison In Denver.

Type	Method	⟨5, 50⟩			⟨5, 100⟩			⟨7, 70⟩			⟨10, 100⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
NT	GCB (R)	18.566	0.659	0.076	25.082	1.795	0.174	20.550	0.274	0.033	25.847	0.057	0.006
	GCB (F)	18.362	0.892	0.113	24.443	2.327	0.269	20.427	0.431	0.057	25.735	0.088	0.010
	OR-Tools (R)	**18.790**	0.955	0.172	**26.068**	2.754	0.342	**20.650**	0.271	0.052	**25.865**	0.009	0.002
	OR-Tools (F)	18.678	2.090	0.391	25.562	3.325	0.432	20.565	0.857	0.170	25.843	0.103	0.019
	LKH (R)	18.418	0.791	0.092	23.762	1.813	0.178	20.358	0.419	0.049	25.775	0.151	0.016
	LKH (F)	18.037	1.186	0.139	22.870	2.272	0.239	19.919	0.779	0.095	25.451	0.404	0.045
	NSGA-II	15.013	2.998	**0.997**	16.385	3.273	**0.997**	17.862	**2.540**	**0.992**	22.359	**2.217**	**0.986**
	GRIP	18.541	1.631	0.263	25.351	2.392	0.283	20.447	0.882	0.153	25.707	0.585	0.095
	FLOG	18.160	**3.210**	0.795	24.180	**4.236**	0.784	19.859	2.408	0.748	25.030	2.097	0.734

Type	Method	⟨5, 200⟩			⟨5, 300⟩			⟨5, 500⟩			⟨5, 1000⟩			⟨10, 200⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	40.607	3.917	0.253	49.787	5.161	0.270	55.984	5.944	0.288	85.081	9.788	0.320	53.679	0.546	0.035
	GCB (F)	39.336	4.673	0.369	48.384	5.871	0.355	53.919	6.728	0.389	84.224	10.293	0.340	53.342	0.839	0.059
	OR-Tools (R)	**41.957**	4.826	0.386	50.240	6.047	0.386	50.743	5.370	0.322	67.192	6.862	0.309	**54.067**	0.526	0.050
	OR-Tools (F)	40.987	5.153	0.424	49.231	6.241	0.408	50.242	5.436	0.333	66.769	7.018	0.318	53.565	1.870	0.178
	LKH (R)	34.429	3.076	0.206	39.311	3.644	0.227	40.819	3.745	0.206	54.991	4.819	0.216	50.493	1.412	0.084
	LKH (F)	33.274	3.632	0.262	37.661	4.306	0.293	39.118	4.383	0.269	51.766	5.931	0.311	48.583	1.950	0.125
	NSGA-II	18.542	3.704	**0.998**	18.240	3.644	**0.997**	19.498	3.896	**0.998**	19.100	3.817	**0.999**	37.058	3.683	**0.988**
	GRIP	41.246	4.318	0.288	**51.069**	5.565	0.319	**57.908**	6.226	0.304	**88.248**	9.137	0.296	53.371	1.466	0.110
	FLOG	39.744	**6.948**	0.780	48.434	**8.321**	0.760	54.260	**9.199**	0.752	83.846	**14.230**	0.756	51.662	**4.216**	0.710

Type	Method	⟨10, 300⟩			⟨10, 500⟩			⟨10, 1000⟩			⟨20, 500⟩			⟨20, 1000⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	66.392	1.040	0.053	91.926	2.699	0.106	118.476	4.828	0.165	**88.588**	0.011	0.001	**138.309**	0.264	0.010
	GCB (F)	65.447	1.699	0.112	90.152	3.618	0.184	116.733	5.592	0.195	88.438	0.035	0.002	137.709	0.572	0.020
	OR-Tools (R)	**66.671**	2.096	0.174	89.372	4.233	0.257	104.454	4.740	0.229	88.299	0.000	0.000	136.605	1.287	0.065
	OR-Tools (F)	66.009	3.026	0.249	88.146	4.563	0.276	104.441	4.969	0.237	88.299	0.003	0.000	136.605	1.325	0.066
	LKH (R)	56.824	1.965	0.107	67.747	2.660	0.128	73.300	2.895	0.140	81.949	1.041	0.054	102.172	1.665	0.082
	LKH (F)	54.882	2.441	0.140	65.560	3.179	0.159	69.983	3.611	0.196	79.643	1.398	0.076	98.994	2.126	0.112
	NSGA-II	33.917	3.373	**0.990**	37.671	3.749	**0.991**	30.017	2.988	**0.992**	55.726	2.734	**0.966**	54.277	2.673	**0.972**
	GRIP	66.105	2.089	0.141	**92.071**	3.727	0.194	**119.955**	5.315	0.205	88.351	0.613	0.048	137.691	1.588	0.090
	FLOG	62.866	**5.085**	0.708	85.473	**6.896**	0.709	106.242	**8.263**	0.682	85.439	**3.289**	0.674	129.036	**4.817**	0.659

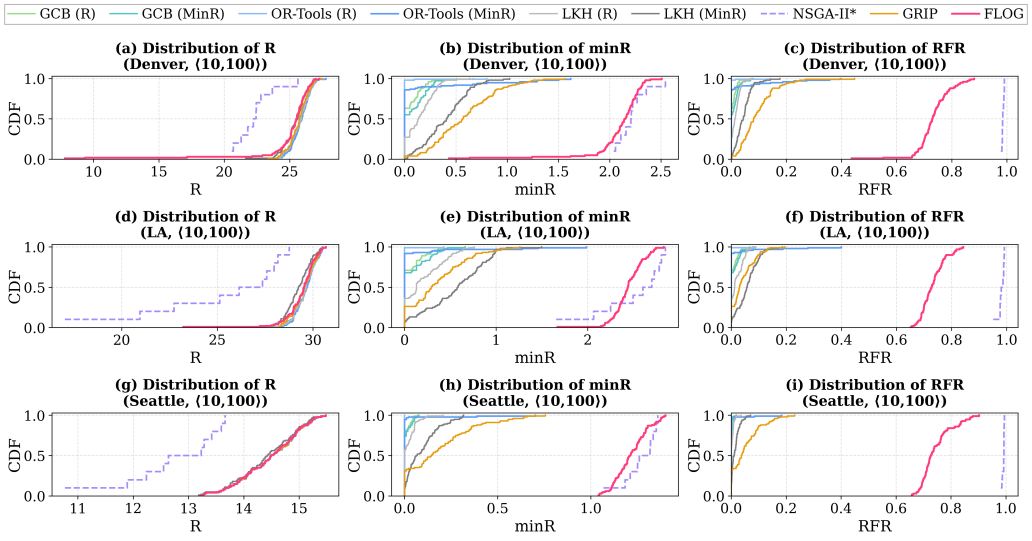

Figure 2: Empirical Distributions of R, min R, and RFR on Real-world Datasets at Scale $\langle 10, 100 \rangle$.

H Sensitivity to the Fairness Threshold

As shown in Table 12, increasing ρ consistently improves both the minimum agent reward and RFR, while gradually reducing R. In particular, when ρ increases from 0.50 to 0.75, min R increases from 3.490 to 3.743 and RFR from 0.589 to 0.762, whereas R decreases from 46.978 to 44.091. This demonstrates that ρ provides an effective control over the fairness–reward trade-off.

Table 10: Performance Comparison In LA.

Type	Method	⟨5, 50⟩			⟨5, 100⟩			⟨7, 70⟩			⟨10, 100⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
NT	GCB (R)	15.407	0.609	0.088	29.249	2.034	0.175	26.251	0.475	0.049	29.627	0.063	0.007
	GCB (F)	15.230	0.841	0.143	28.709	2.548	0.247	26.066	0.604	0.071	29.574	0.084	0.010
	OR-Tools (R)	**15.600**	0.928	0.184	**30.132**	3.116	0.356	**26.385**	0.333	0.052	**29.637**	0.004	0.001
	OR-Tools (F)	15.524	1.848	0.406	29.699	4.119	0.491	26.206	0.981	0.156	29.628	0.062	0.011
	LKH (R)	15.317	0.655	0.092	27.490	2.173	0.175	26.055	0.560	0.053	29.591	0.198	0.020
	LKH (F)	14.907	1.116	0.170	26.556	2.688	0.235	25.418	1.088	0.110	29.239	0.556	0.059
	NSGA-II	11.571	2.309	**0.995**	17.204	3.436	**0.998**	20.806	2.960	**0.992**	25.182	**2.493**	**0.981**
	GRIP	15.286	0.882	0.150	29.282	2.399	0.224	26.056	0.779	0.093	29.528	0.353	0.042
	FLOG	15.082	**2.674**	0.796	28.430	**5.054**	0.800	25.603	**3.128**	0.754	29.385	2.443	0.732

Type	Method	⟨5, 200⟩			⟨5, 300⟩			⟨5, 500⟩			⟨5, 1000⟩			⟨10, 200⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	48.357	4.898	0.298	57.731	6.344	0.332	**77.667**	9.146	0.366	**87.587**	10.931	0.407	51.271	0.548	0.039
	GCB (F)	47.708	5.469	0.342	55.295	7.473	0.477	75.420	9.914	0.421	85.270	11.639	0.447	50.980	0.761	0.059
	OR-Tools (R)	**49.913**	6.404	0.446	**58.442**	7.297	0.416	70.751	8.863	0.425	70.336	8.327	0.377	**51.416**	0.466	0.049
	OR-Tools (F)	48.764	6.633	0.473	57.643	7.422	0.430	69.926	8.949	0.434	69.954	8.461	0.386	50.976	1.601	0.173
	LKH (R)	38.370	4.062	0.279	42.996	4.197	0.233	52.652	5.443	0.279	49.212	5.145	0.299	47.306	1.465	0.111
	LKH (F)	36.762	4.677	0.359	41.332	4.858	0.292	49.813	6.473	0.384	46.494	6.143	0.403	45.476	2.077	0.169
	NSGA-II	22.301	4.455	**0.998**	22.212	4.437	**0.976**	20.278	4.051	**0.998**	19.221	3.842	**0.999**	30.479	3.028	**0.988**
	GRIP	48.183	4.866	0.268	57.868	6.591	0.315	76.758	8.944	0.322	84.296	10.097	0.345	50.846	0.980	0.074
	FLOG	47.765	**8.380**	0.785	56.730	**9.895**	0.770	76.634	**13.150**	0.747	86.278	**14.231**	0.707	49.889	**4.156**	0.722

Type	Method	⟨10, 300⟩			⟨10, 500⟩			⟨10, 1000⟩			⟨20, 500⟩			⟨20, 1000⟩		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	83.273	1.378	0.063	**102.460**	3.266	0.141	**188.333**	8.671	0.226	**111.662**	0.009	0.000	**208.443**	0.391	0.012
	GCB (F)	82.542	1.988	0.102	100.301	4.438	0.218	185.111	9.980	0.277	111.602	0.019	0.001	207.844	0.811	0.027
	OR-Tools (R)	**83.466**	2.759	0.196	98.128	4.959	0.283	163.010	8.267	0.284	111.539	0.026	0.002	206.044	1.875	0.078
	OR-Tools (F)	82.720	3.994	0.277	96.683	5.288	0.304	163.010	8.780	0.301	111.539	0.027	0.002	206.044	1.936	0.079
	LKH (R)	70.809	2.563	0.132	71.189	3.129	0.171	105.570	4.800	0.187	103.077	1.424	0.084	144.617	2.849	0.122
	LKH (F)	67.428	3.368	0.191	67.963	3.795	0.222	101.013	5.753	0.246	99.547	1.979	0.125	139.486	3.565	0.166
	NSGA-II	43.119	4.292	**0.991**	38.885	3.868	**0.991**	46.936	4.673	**0.992**	71.694	3.529	**0.972**	83.348	4.107	**0.974**
	GRIP	82.515	1.956	0.089	101.649	3.911	0.164	185.354	8.942	0.220	111.460	0.420	0.019	207.128	1.424	0.039
	FLOG	80.994	**6.723**	0.718	99.171	**8.102**	0.697	181.639	**14.606**	0.686	111.006	**4.395**	0.688	204.606	**7.989**	0.673

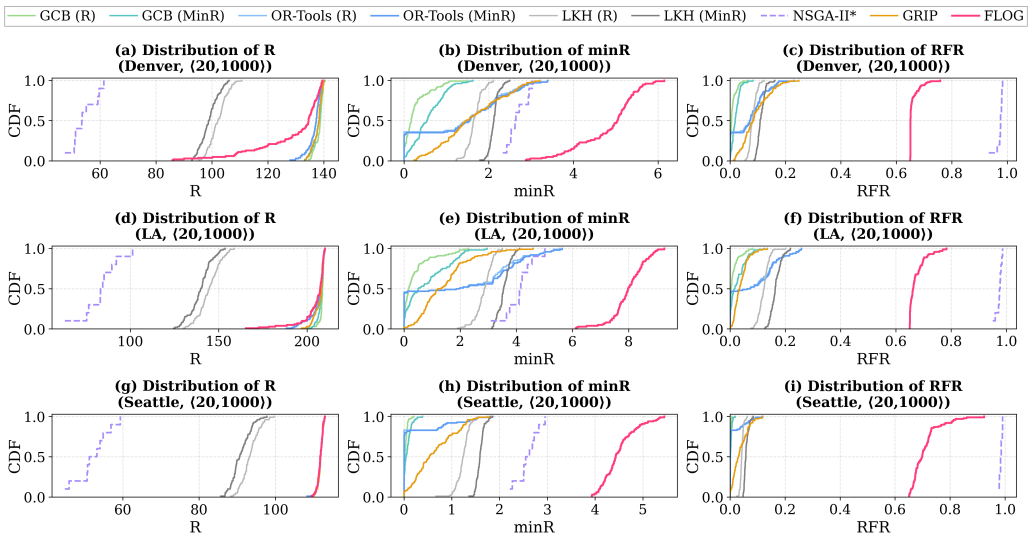

Figure 3: Empirical Distributions of R, min R, And RFR on Real-world Datasets at Scale $\langle 20, 1000 \rangle$.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, grammar checking, and improving the clarity of technical writing. We also used generative AI tools for assisting with code debugging and implementation-related discussions. We have not used generative AI tools for generating experimental results, designing the core methodology, conducting scientific analysis, or making research claims. All AI-assisted content has been carefully reviewed and verified by the authors. We

Table 11: Performance Comparison In Seattle.

Type	Method	$\langle 5, 50\rangle$			$\langle 5, 100\rangle$			$\langle 7, 70\rangle$			$\langle 10, 100\rangle$		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
NT	GCB (R)	9.331	0.109	0.021	20.477	0.813	0.080	13.999	0.046	0.007	14.480	0.007	0.001
	GCB (F)	9.296	0.212	0.042	20.154	1.228	0.136	13.976	0.075	0.012	14.475	0.009	0.002
	OR-Tools (R)	**9.356**	0.178	0.052	**20.803**	1.571	0.231	**14.014**	0.030	0.009	**14.481**	0.000	0.000
	OR-Tools (F)	9.332	0.621	0.206	20.616	2.554	0.398	14.010	0.172	0.045	14.480	0.011	0.003
	LKH (R)	9.325	0.136	0.025	20.004	0.973	0.094	13.982	0.070	0.010	14.477	0.021	0.003
	LKH (F)	9.200	0.336	0.062	19.354	1.423	0.145	13.878	0.234	0.035	14.415	0.093	0.015
	NSGA-II	8.114	1.620	**0.996**	13.272	2.652	**0.998**	12.348	**1.758**	**0.993**	12.727	**1.266**	**0.991**
	GRIP	9.321	0.572	0.169	20.462	1.638	0.219	13.986	0.408	0.098	14.478	0.181	0.050
	FLOG	9.275	**1.638**	0.794	20.350	**3.575**	0.788	13.963	1.731	0.773	14.463	1.215	0.744

Type	Method	$\langle 5, 200\rangle$			$\langle 5, 300\rangle$			$\langle 5, 500\rangle$			$\langle 5, 1000\rangle$			$\langle 10, 200\rangle$		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	32.121	2.236	0.146	36.058	3.031	0.176	55.367	4.896	0.193	77.629	7.393	0.217	29.932	0.054	0.004
	GCB (F)	31.327	2.880	0.223	34.963	3.503	0.253	53.427	5.776	0.291	74.200	8.652	0.327	29.874	0.114	0.011
	OR-Tools (R)	**32.963**	3.645	0.348	**36.233**	4.109	0.355	50.280	5.315	0.288	57.706	5.627	0.289	**29.954**	0.042	0.007
	OR-Tools (F)	32.262	4.154	0.418	35.303	4.323	0.385	49.489	5.398	0.301	57.486	5.800	0.297	29.915	0.403	0.069
	LKH (R)	29.063	2.112	0.140	30.504	2.428	0.155	43.351	3.494	0.161	53.907	4.223	0.158	29.489	0.396	0.032
	LKH (F)	28.084	2.593	0.180	29.317	2.865	0.197	41.769	4.126	0.200	51.628	5.051	0.211	28.778	0.768	0.064
	NSGA-II	17.060	3.411	**0.999**	14.210	2.841	**0.999**	15.366	3.072	**0.999**	17.359	3.471	**0.999**	22.801	2.272	**0.993**
	GRIP	32.318	2.895	0.237	36.190	3.547	0.273	**56.298**	5.731	0.260	79.171	7.892	0.249	29.916	0.533	0.072
	FLOG	32.121	**5.712**	0.802	35.710	**6.397**	0.813	55.891	**10.060**	0.819	**79.632**	**14.328**	0.819	29.842	**2.518**	0.746

Type	Method	$\langle 10, 300\rangle$			$\langle 10, 500\rangle$			$\langle 10, 1000\rangle$			$\langle 20, 500\rangle$			$\langle 20, 1000\rangle$		
		R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR	R	min R	RFR
GT	GCB (R)	41.975	0.169	0.011	**67.603**	0.911	0.040	81.026	2.356	0.088	**61.672**	0.001	0.000	**111.828**	0.033	0.001
	GCB (F)	41.801	0.409	0.030	66.619	1.618	0.091	79.315	3.078	0.146	61.668	0.002	0.000	111.786	0.069	0.003
	OR-Tools (R)	**41.982**	0.819	0.104	65.769	2.546	0.210	72.990	2.747	0.185	61.670	0.000	0.000	111.744	0.186	0.010
	OR-Tools (F)	41.799	1.479	0.186	65.414	2.817	0.228	72.990	2.910	0.192	61.670	0.000	0.000	111.744	0.189	0.011
	LKH (R)	38.924	1.017	0.062	54.952	1.777	0.084	55.358	1.900	0.093	60.155	0.489	0.025	93.354	1.235	0.043
	LKH (F)	37.805	1.355	0.086	52.946	2.221	0.109	53.255	2.330	0.119	58.898	0.770	0.041	90.892	1.595	0.059
	NSGA-II	22.144	2.206	**0.992**	26.618	2.653	**0.993**	22.371	2.232	**0.996**	45.288	2.242	**0.983**	52.176	2.587	**0.985**
	GRIP	41.960	0.939	0.092	67.562	2.063	0.124	**81.123**	3.081	0.173	61.668	0.270	0.028	111.816	0.627	0.036
	FLOG	41.816	**3.490**	0.742	67.048	**5.755**	0.759	79.024	**6.597**	0.728	61.640	**2.470**	0.706	111.691	**4.492**	0.707

Table 12: Sensitivity to the Fairness Threshold ρ on $\langle 10, 100\rangle$ Validation Instances.

ρ	R	min R	RFR
0.50	**46.978**	3.490	0.589
0.55	46.667	3.572	0.619
0.60	46.184	3.628	0.655
0.65	45.646	3.676	0.687
0.70	44.510	3.684	0.725
0.75	44.091	**3.743**	**0.762**

take full responsibility for the final content of this work, including all text, claims, and experimental results.

Ethics Statement

This work studies fairness-aware multi-agent routing optimization and does not involve human subjects or personal decision-making. The real-world datasets used in our experiments are aggregated spatial incident datasets and are processed without personally identifiable information. The proposed method optimizes agent-level reward allocation under routing constraints and is not intended for predicting individual behavior or making law-enforcement decisions. We acknowledge the importance of responsible deployment of fairness-aware optimization systems and encourage further evaluation in specific application contexts before practical use.

Reproducibility Statement

To facilitate reproducibility, we provide detailed descriptions of the Fair Team Orienteering Problem formulation, the FLOG architecture, training procedure, and experimental settings in the main paper and supplementary materials. The implementation details, hyperparameters, dataset generation procedures, and additional experimental results are provided in the supplementary materials. We will release the source code, trained models, and processed datasets upon publication.