

Appendix

1 A Mathematical model of GTSP

2 This appendix provides the detailed integer linear programming formulation of the Generalized
3 Traveling Salesman Problem (GTSP) referenced in the main body of this paper.

4 **Definitions and notation.** The GTSP is defined on a graph where nodes are partitioned into a
5 number of predefined, mutually exclusive clusters. The objective is to find a minimum-cost tour that
6 visits exactly one node from each cluster.

7 Let $V = v_1, v_2, \dots, v_n$ be the set of n nodes. The set of nodes V is partitioned into m mutually ex-
8 clusive and collectively exhaustive clusters V_1, V_2, \dots, V_m , such that $V = \bigcup_{i=1}^m V_i$ and $V_i \cap V_j =$
9 $\emptyset, \forall i \neq j$. Let c_{ij} be the non-negative cost (e.g., distance or travel time) associated with traversing
10 the arc from node $i \in V$ to node $j \in V$.

11 The following decision variables are used in the model:

12 x_{ij} : A binary variable. $x_{ij} = 1$ if the tour travels directly from node i to node j , and 0 otherwise.

13 y_i : A binary variable. $y_i = 1$ if node i is included in the tour, and 0 otherwise.

14 u_i : An auxiliary continuous variable used for Miller-Tucker-Zemlin (MTZ) subtour elimination. If
15 node i is visited, it represents the position of node i in the sequence of the tour.

16 **Objective Function** The objective is to minimize the total cost of the tour, which is the sum of the
17 costs of all selected arcs:

$$\min Z = \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} x_{ij} \quad (1)$$

18 **Constraints** The minimization of the objective function is subject to the following constraints.

$$\sum_{i \in V_p} y_i = 1, \forall p \in \{1, 2, \dots, m\} \quad (2)$$

$$\sum_{j \in V, j \neq i} x_{ji} = y_i, \forall i \in V \quad (3)$$

$$\sum_{j \in V, j \neq i} x_{ij} = y_i, \forall i \in V \quad (4)$$

$$y_i \leq u_i \leq m \cdot y_i, \forall i \in V \quad (5)$$

$$u_i - u_j + m \cdot x_{ij} \leq m - 1, \forall i, j \in V, i \neq j \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V, i \neq j \quad (7)$$

$$y_i \in \{0, 1\}, \forall i \in V \quad (8)$$

$$u_i \geq 0, \forall i \in V \quad (9)$$

19 Constraint(2) ensures that exactly one node is selected and included in the final tour from each of
20 the m clusters. Constraint(3) and (4) ensure that if a node i is selected for the tour (i.e., $y_i =$
21 1), then exactly one arc must enter that node, and exactly one arc must leave it. If a node is not
22 selected ($y_i = 0$), no arcs can be incident to it, this guarantees proper connectivity at the node
23 level. Constraints(5) and (6) are the MTZ subtour elimination constraints, designed to ensure a
24 single, continuous tour. Constraint(5) represents that if node i is visited ($y_i = 1$), u_i takes a value
25 between 1 and m (representing its position in the m -node tour), and $u_i = 0$ if node i is not visited.
26 Constraint(6) then imposes a sequential order: if the tour travels from node i to node j ($x_{ij} = 1$), this

implies $u_j \geq u_i + 1$, meaning node j must appear after node i in the sequence, thereby preventing the formation of premature cycles or subtours before all m clusters have been visited. Constraints (7 - 9) are the variable constraints.

The GTSP is an NP-hard combinatorial optimization problem. This implies that finding a provably optimal solution using exact algorithms based on the above ILP formulation can be computationally intractable for large-scale instances. This intractability underscores the necessity for advanced approaches, such as Deep Reinforcement Learning, to effectively tackle such complex problems by learning high-quality solution policies.

B Training Procedure

The training algorithm for our MMFL framework is summarized in Algorithm 1. We employ the REINFORCE algorithm to train our model. To reduce the variance of the policy gradient and stabilize the training process, we incorporate a shared baseline. This baseline is computed for each problem instance by averaging the rewards obtained from multiple rollouts (i.e., multiple generated solution tours). For each GTSP instance λ_i , we sample k different routes $\pi^{i,j}$ using the SAMPLEROLLOUT function (line 4), which utilizes the multi-start decoder to generate feasible solutions from different starting nodes based on the k -nearest neighbors principle. We then compute the shared baseline $b(\lambda_i)$ for each instance, which is the average reward across the k solutions (line 5).

In line 7, we calculate the policy gradient $\nabla_{\theta} \mathcal{L}(\theta|\lambda_i)$ using the core REINFORCE formula. Here, $\mathcal{R}(\pi^{i,j}|\lambda_i) - b(\lambda_i)$ serves as an advantage function, reducing the variance of the gradient estimates. The $\nabla_{\theta} \mathcal{L}(\theta|\lambda_i)$ term indicates increasing the probability of producing solutions with high rewards while decreasing the probability of solutions with low rewards. Finally, line 8 updates the model parameters θ using the Adam optimizer, adjusting the weights through the policy gradient and the learning rate α .

Algorithm 1: Training Algorithm for MMFL

Input: Initialize policy network p_{θ} with random weights θ , number of training epochs E , number of rollouts k , number of batches B per epoch, batch size N , learning rate α

```

1 for  $epoch = 1, \dots, E$  do
2   for  $b = 1, \dots, B$  do
3     for  $i = 1, \dots, N$  do
4        $\pi^{i,j} \leftarrow \text{SAMPLEROLLOUT}(\lambda_i) \quad \forall j \in \{1, 2, \dots, k\};$ 
5        $b(\lambda_i) \leftarrow \frac{1}{k} \sum_{j=1}^k \mathcal{R}(\pi^{i,j});$ 
6     end
7      $\nabla_{\theta} \mathcal{L}(\theta|\lambda_i) \simeq \frac{1}{kN} \sum_{i=1}^N \sum_{j=1}^k (\mathcal{R}(\pi^{i,j}|\lambda_i) - b(\lambda_i)) \nabla_{\theta} \log p_{\theta}(\pi^{i,j}|\lambda_i);$ 
8      $\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathcal{L}(\theta|\lambda_i);$ 
9   end
10 end
```

C Ablation Study

Our ablation study demonstrates the critical contribution of each component in the MMFL framework for solving GTSP. The results are shown in Table 1. For smaller problem instances ($n = 20, c = 4$ and $n = 50, c = 10$), all model variants perform similarly, but as problem complexity increases, the advantages of our complete architecture become evident. Without the Image Encoder, performance gaps range from 1.59% to 10.00%, while removing the Multimodal Fusion module causes even larger degradation (up to 35.83% for Large group distributions).

The most significant performance differences appear in complex spatial configurations and heterogeneous structures (Hybrid, Mixed, and Large groups), confirming that MMFL’s strength lies in effectively integrating both topological and geometric information. Importantly, all model variants maintain comparable inference times, demonstrating that our approach achieves superior solution

62 quality without sacrificing the computational efficiency required for real-time robotic task planning.

Table 1: Ablation study.

Instances	MMFL		w/o Multimodal Fusion			w/o Image Encoder		
	Obj.	Time	Obj.	Gap	Time	Obj.	Gap	Time
$n = 20, c = 4$	1.13	0.12	1.13	0.00	0.11	1.13	0.00	0.12
$n = 50, c = 10$	1.71	0.11	1.71	0.00	0.12	1.71	0.00	0.09
$n = 100, c = 20$	2.25	0.13	2.31	2.67	0.11	2.69	19.56	0.11
$n = 150, c = 30$	3.20	0.13	3.32	3.75	0.13	3.53	10.31	0.12
$n = 200, c = 40$	3.63	0.14	3.74	3.03	0.13	3.78	4.13	0.12
Random	2.37	0.13	2.41	1.69	0.12	2.47	4.22	0.12
Proximity	3.15	0.15	3.20	1.59	0.13	3.22	2.22	0.12
Density	1.47	0.09	1.50	2.04	0.10	1.56	6.12	0.08
Hybrid	2.16	0.12	2.27	5.09	0.11	2.42	12.04	0.10
Uniform	2.10	0.12	2.23	6.19	0.12	2.31	10.00	0.10
Small	3.92	0.16	3.98	1.53	0.14	4.12	5.10	0.14
Large	1.20	0.12	1.32	10.00	0.11	1.63	35.83	0.10
Mixed	2.01	0.13	2.21	9.95	0.13	2.36	17.41	0.11

63