HLMTrans: A Sim-to-Real Transfer Framework for Spatial Crowdsourcing with Human-Guided Language Models

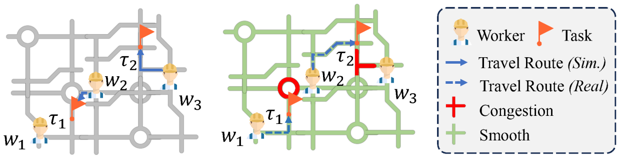
Qingshun Wu, **Yafei Li***, **Lulu Li**, **Yuanyuan Jin**, **Shuo He**, **Mingliang Xu**
School of Computer Science and Artifcial Intelligence, Zhengzhou University
wqszzu@gs.zzu.edu.cn, {ieyfli, lilulu1997, jinyuanyuan, heshuo, iexumingliang}@zzu.edu.cn

Abstract

Reinforcement Learning (RL), trained via trial and error in simulators, has been proven to be an effective approach for addressing task assignment problems in spatial crowdsourcing. However, a performance gap still exists when transferring the simulator-trained RL Models (RLMs) to real-world settings due to the misalignment of travel time. Existing works mostly focus on using data-driven and learning-based methods to predict travel time; unfortunately, these approaches are limited in achieving accurate predictions by requiring a large amount of real-world data covering the entire state distribution. In this paper, we propose a Sim-to-Real Transfer with Human-guided Language Models framework called HLMTrans, which comprises three core modules: RLMs decision for task assignment, sim-to-real transfer with large language models (LLMs), and preference learning from human feedback. HLMTrans first leverages the zero-shot chain-of-thought reasoning capability of LLMs to estimate travel time by capturing the real-world dynamics. This estimation is then input as domain knowledge into the forward model of Grounded Action Transformation (GAT) to enhance the action transformation of RLMs. Further, we design a human preference learning mechanism to fine-tune LLMs, improving their generation quality and enabling RLMs learn a more realistic policy. We evaluate the proposed HLMTrans on two real-world datasets, and the experimental results demonstrate that HLMTrans outperforms the SOTA methods.

1 Introduction

Task assignment [Li *et al.*, 2024; Li *et al.*, 2025] is a critical problem aimed at maintaining the operation of Spatial Crowdsourcing (SC) platforms (*e.g.,* Uber, Gigwalk, and Waze). Recently, Reinforcement Learning (RL) methods have demonstrated remarkable success in addressing task assignment challenges through trial and error in simulators [Li *et al.*, 2023; Wang *et al.*, 2023; Wu *et al.*, 2024], which are valuable tools for training RL models (RLMs) by providing simplified and controlled environments compared to the complexities of real-world settings. However, due to discrepancies in system dynamics between training simulators and actual road conditions, a notable performance gap (*i.e.,* sim-to-real gap [Da *et al.*, 2023]) emerges when transferring simulator-trained RLMs to real-world environments.



(a) Simulator Assignment (b) Real-World Assignment

Figure 1: A toy example with sim-to-real gap in travel time.

The misalignment of travel time is a key factor contributing to the sim-to-real gap, as travel time calculations in simulated environments usually rely on simplified distance-time relationships or fixed traffic parameters, failing to capture the system dynamics of real-world road conditions [Da *et al.*, 2024]. As shown in Fig. 1, we assume that the RLM's optimal assignment policy in the simulator is $\{(w_2, \tau_1), (w_3, \tau_2)\}$, yielding a total travel time of 3 (=1+2). However, in the real world, traffic congestion (*i.e.,* red roads) leads to an increase in total travel time to 9 (=3+6), even though the travel distance remains unchanged. In this case, the optimal assignment policy becomes $\{(w_1, \tau_1), (w_2, \tau_2)\}$, with a total travel time of 5 (=2+3).

Existing studies on travel time estimation can be categorized into two types: data-driven and learning-based approaches. Data-driven methods typically divide a route into discrete segments and predict travel time for each using rule-based [Wang *et al.*, 2014], statistical [Wang *et al.*, 2023], or spatio-temporal pattern analysis techniques [Tang *et al.*, 2018]. These methods overlook the dynamics and uncertainty of traffic systems, such as intersection and traffic light delays, reducing segment-level accuracy and causing cumulative errors and biases. Moreover, the impact of personalized preferences on travel time is often neglected when routes are treated as simple sequences of road segments, such as workers may prioritize time-efficient routes during peak hours but opt for distance-efficient routes during off-peak hours. In contrast,

[*] Corresponding author.

learning-based approaches can directly predict travel time for entire routes by leveraging deep learning models like CNNs [Wang *et al.*, 2018], RNNs [Ye *et al.*, 2022], and GNNs [Liu *et al.*, 2024] to implicitly capture the interactive relationships between intersections and road segments. However, their accurate predictions usually require large amounts of real-world data covering the entire state distribution.

Therefore, mitigating the sim-to-real gap by aligning travel time still faces the following challenges: 1) the real-world data with system dynamics is both costly and sparse, making it challenging to accurately estimate travel times in real-world settings with limited data; 2) the workers' preferences for travel routes vary across different contexts, and these preference significantly impact travel time estimation, how to effectively capture these preferences is another challenge.

To tackle the above challenges, we propose HLMTrans, a Sim-to-Real Transfer framework with Human-Guided Language Models, aimed at bridging the performance gap between simulator-trained RLMs and real-world deployment caused by travel time misalignment. In this framework, we first propose a Large Language Models (LLMs)-based method for operationalizing Grounded Action Transformation (GAT) [Hanna and Stone, 2017], which integrates prompt generation and dynamics modeling modules to capture changes in system dynamics during sim-to-real transfer. By leveraging LLMs with prompts and zero-shot chain-of-thought reasoning, we can accurately estimate travel time with limited data and enhance RLMs' understanding of real-world dynamics. Furthermore, we design a human preference learning mechanism that fine-tunes LLMs using human feedback, enabling them to align with human preferences and improve the relevance and quality of their generations. Our main contributions can be summarized as follows:

- We investigate the RLMs-based Task Assignment (RTA) problem in spatial crowdsourcing, focusing on bridging the performance gap between simulator-trained RLMs and real-world settings due to travel time misalignment. To this end, we propose HLMTrans, the first sim-to-real transfer framework that leverages human-guided LLMs to estimate travel times.
- We present a Monte Carlo Graph Search-based method to identify minimal congestion subgraphs affecting worker-task travel times, serving as contextual prompts to accelerate LLMs' inference. Additionally, we introduce a human preference learning mechanism to fine-tune LLMs with human feedback, enhancing its alignment with human preferences and generation quality.
- We extensively evaluate HLMTrans against three widely used benchmarks on two real-world datasets. The results demonstrate that HLMTrans outperforms state-of-the-art methods, validating the effectiveness of HLM-Trans.

2 Related Work

Sim-to-real Transfer. The sim-to-real transfer methods can generally be divided into three main categories: 1) *domain randomization* [Wu *et al.*, 2023], aims to train policies capable of adapting to diverse environmental variations. This approach leverages simulated data and is particularly beneficial when handling uncertain or evolving target domains; 2) *domain adaptation* [Bousmalis *et al.*, 2018], aims to achieve domain distribution shifts by aligning features between the source and target domains. Many domain adaptation methods emphasize bridging perception gaps in robotics [Fang *et al.*, 2018; James *et al.*, 2019]. However, in travel time estimation, the primary domain gap lies in dynamics rather than perception, since it typically represents observations in vectorized forms, such as road-level worker number and average speed; 3) *grounding methods*, enhance simulator accuracy by correcting biases relative to the real world. Recently, Grounded Action Transformation (GAT) has shown significant promise in sim-to-real transfer of traffic signal control [Da *et al.*, 2024; Da *et al.*, 2023]. Different from system identification techniques [Cutler *et al.*, 2014; Cully *et al.*, 2015] that estimate precise physical parameters, GAT [Hanna and Stone, 2017; Desai *et al.*, 2020] adjusts the simulator's dynamics to align with real-world actions without requiring a parameterized simulator for modification. Building on GAT, our HLMTrans utilizes human-guided LLMs to estimate travel time and inputs it as domain knowledge into the forward model of GAT, enhancing the action transformation of RLMs.

Large Language Models with Human Guidance. Large Language Models (LLMs) have demonstrated exceptional linguistic capabilities; however, their alignment with human intentions remains challenging due to pre-training on large and noisy datasets. Human guidance plays a pivotal role in refining LLMs to generate coherent, ethical, and human-aligned outputs [Vafa *et al.*, 2024; Jagadish *et al.*, 2024]. The traditional approach for fine-tuning LLMs based on human preferences involves learning a reward signal using the Bradley-Terry model [Bradley and Terry, 1952], followed by applying RL to optimize against the learned reward signal [Griffith *et al.*, 2013; Christiano *et al.*, 2017]. High-quality human feedback datasets, including VisAlign [Lee *et al.*, 2023] and PKU-SafeRLHF [Ji *et al.*, 2023], have further enhanced LLMs' ability to interpret and respond to diverse instructions. However, inconsistencies among annotators and misalignments with human intentions can compromise the quality of feedback, posing challenges to effective model fine-tuning. Recent advancements, such as Direct Preference Optimization (DPO)[Lee *et al.*, 2023], provide a model-free alternative that eliminates the need for explicit reward learning. Inspired by DPO, we propose a human preference learning mechanism that enables fine-tuning of LLMs using human feedback without an explicit reward model.

3 Preliminaries

3.1 Concepts of RTA and RL Solutions

On the SC platform, a road network, is denoted as a weighted graph $G =< V, E, U >$, where $v_i \in V$ is an intersection, $e_{ij} \in E$ is a road that connects v_i and v_j, and $u_{ij} \in U$ is the weight of e_{ij}, determined by factors such as road length and road type. Users can dynamically publish spatial tasks at any time, each of which can be denoted as $\tau =< l_\tau, t_\tau, d_\tau, \rho_\tau >$, where $l_\tau \in V$ is the task's location, t_τ is the publish time, d_τ is the task's deadline, and ρ_τ is the payoff for serving

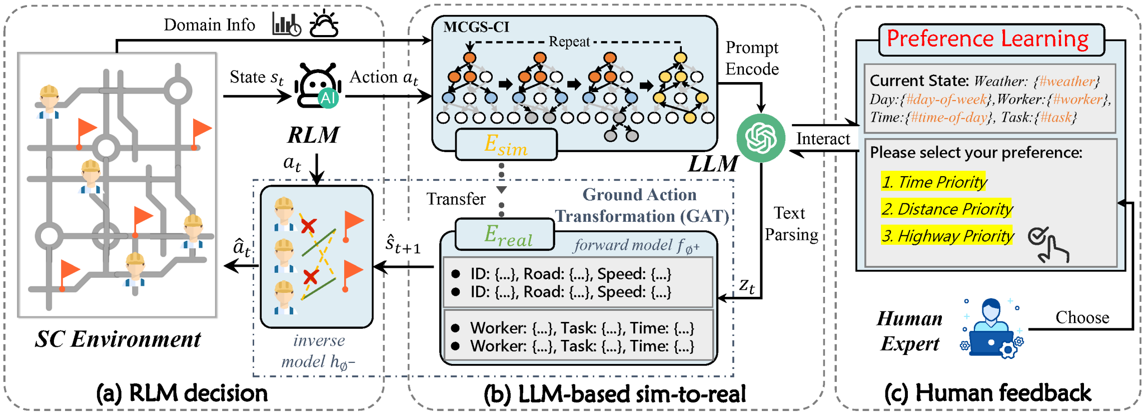

Domain Info
State s_t
Action a_t
RLM
a_t
$\hat{a}_t$
$\hat{s}_{t+1}$
inverse model h_{ϕ^-}
SC Environment
(a) RLM decision
MCGS-CI
Repeat
E_{sim}
LLM
Transfer
Prompt Encode
Text Parsing
z_t
Ground Action Transformation (GAT)
E_{real}
forward model f_{ϕ^+}
ID: {...}, Road: {...}, Speed: {...}
ID: {...}, Road: {...}, Speed: {...}
Worker: {...}, Task: {...}, Time: {...}
Worker: {...}, Task: {...}, Time: {...}
(b) LLM-based sim-to-real
Interact
Preference Learning
Current State: Weather: {#weather} Day:{#day-of-week}, Worker:{#worker}, Time:{#time-of-day}, Task:{#task}
Please select your preference:
1. Time Priority
2. Distance Priority
3. Highway Priority
Human Expert
Choose
(c) Human feedback

Figure 2: An overview of HLMTrans.

the task. Once tasks are published, the platform assigns mobile workers to physically travel to the task locations and complete them. Here, a worker can be denoted as a triplet $w =< l_w, c_w, s_w >$, where $l_w \in V$ is the worker's location, c_w is the worker's status (*i.e.,* idle or busy), and s_w is the worker's service range. The RTA problem can be formulated as: Given a set of mobile workers W and a stream of spatial tasks $\varGamma$, the problem aims to apply RLMs-based approaches to find an assignment plan $\mathcal{M} = \{(w, \tau) | w \in W, \tau \in \varGamma\}$ that maximizes the platform's total revenue $\mathbb{E}_{\mathcal{M}}$:

$$\mathbb{E}_{\mathcal{M}} = \sum_{(w,\tau)\in\mathcal{M}} \mathcal{R}(w, \tau), \tag{1}$$

where (w, τ) is a valid worker-task pair that satisfies the spatiotemporal constraints (*e.g.,* reach the specified location before the deadline), and $\mathcal{R}(w, \tau) = \rho_\tau - \kappa \cdot \mathcal{T}_{w\tau}$ is the revenue function, in which $\mathcal{T}_{w\tau}$ is the travel time of w finishing τ, and κ is the travel cost per unit time.

The RTA problem is typically modeled as a Markov Decision Process (MDP), defined as $M =< S, A, R, P, \gamma >$, where S represents the state space, A represents the action space, $r_t \in R$ is the reward obtained by executing action $a_t \in A$ in state $s_t \in S$ at time slice t, P is the probability of transitioning from state s_t to state s_{t+1} after executing a_t, and $\gamma \in [0, 1]$ represents the discount factor describing the importance of future rewards. An RL approach solves this problem by learning an optimal policy π that maximizes the the long-term expectation of discounted accumulation reward $\mathbb{E}_{(s,a)\sim\pi}[\sum_{t=0}^{T} \gamma^{T-t} r(s, a)]$, where $r(s, a)$ is the immediate reward obtained by agent after executing action a in state s.

3.2 Grounded Action Transformation

The Grounded Action Transformation (GAT) is a framework proposed to address the challenges of transferring policies learned in simulated environments to real-world applications [Hanna and Stone, 2017]. This framework focuses on bridging the discrepancies between simulated dynamics and real-world dynamics. Within the GAT framework, the simulated environment E_{sim} is modeled as an MDP, where the transition dynamics, parameterized as $P_\phi(\cdot|s, a)$, is imperfect but can be adjustable. By leveraging real-world trajectories, GAT iteratively refines the simulated dynamics to better reflect real-world behavior. Given a real-world trajectory dataset $D_{real} = \{o_1, o_2, \ldots, o_I\}$, where each trajectory $o_i =< s_0^i, a_0^i, s_1^i, a_1^i, \ldots, s_{T-1}^i, a_{T-1}^i, s_T^i >$ represents a sequence of states and actions collected by executing a policy π within the real-world environment E_{real}. The goal of GAT is to optimize ϕ^* such that the differences between the transition dynamics of the simulated environment and the real-world environment is minimized:

$$\phi^* = \arg\min_{\phi} \sum_{o_i \in D_{real}} \sum_{t=0}^{T-1} d(P^*(\cdot), P_\phi(\cdot), \tag{2}$$

where $P^*(s_{t+1}^i | s_t^i, a_t^i)$ is the transition dynamics of E_{real}, $P_\phi(s_{t+1}^i | s_t^i, a_t^i)$ is the transition dynamics of E_{sim}, and $d(\cdot)$ quantifies the discrepancy between P^* and P_ϕ.

Given the current state s_t and the action a_t predicted by the policy π, the grounded action $\hat{a}_t$, executed in E_{sim}, adjusts the resulting next state s_{t+1} in E_{sim} to approximate the predicted next state $\hat{s}_{t+1}$ in E_{real}. This process ensures that the simulated dynamics $P_\phi(s_{t+1}|s_t, \hat{a}_t)$ align closely with the real-world dynamics $P^*(\hat{s}_{t+1}|s_t, a_t)$. Hence, the policy π, trained in E_{sim} using P_ϕ, achieves dynamics more similar to P^*, reducing the performance gap when transferred to E_{real}.

4 Methodology

4.1 Overview

The overview of HLMTrans, as depicted in Fig. 2, consists of three key components: (a) RLMs Decision for Task Assignment, which employs the RLMs to dynamically allocate tasks to workers based on real-time observed states, with the objective of maximizing the platform's total revenue, (b) LLMs-based Sim-to-Real Transfer, which utilizes the LLMs to bridge the sim-to-real gap by adapting simulated dynamics to align with real-world dynamics, ensuring more reliable policies, and (c) Preference Learning from Human Feedback, which enables the LLMs to iteratively learn human preferences through interaction, using this knowledge to fine-tune their generates and enhance the alignment between worker travel times and real-world settings, further improving the RLMs' performance.

4.2 RLMs Decision for Task Assignment

In this paper, we formulate the RTA problem as an MDP, where the SC platform is treated as an agent that learns an optimize policy through interactions with the environment. Specifically, the agent observes real-time states, and makes decisions for worker-task matching in a trial-and-error manner. Next, we detail the modeling of state space S, action space A, and reward function R, as these are the critical components in reinforcement learning.

State Space S. At any time slice t, the state $s_t \in S$ can be represented as a six-tuple: $s_t =< D_{\mathit{\Gamma}}, L_{\mathit{\Gamma}}, L_W, l_\tau, l_w, t >$, where $D_{\mathit{\Gamma}}$ is the deadline distribution of unassigned tasks $\mathit{\Gamma}$, $L_{\mathit{\Gamma}}$ is the location distribution of unassigned tasks $\mathit{\Gamma}$, L_W is the location distribution of idle workers W, l_τ is the location of task $\tau \in \mathit{\Gamma}$, and l_w is the location of worker $w \in W$.

Action Space A. During time interval Δt, the SC platform accumulates a batch of unassigned tasks $\mathit{\Gamma}$ and a set of idle workers W. For each task $\tau \in \mathit{\Gamma}$, the agent observes the current state s_t and selects an action a_t based on the learned policy π. For each worker $w \in W$, the action a_t is defined as: $a_t = \{0, 1\}$, where 1 indicates assigning the worker w to τ, and 0 indicates not assigning.

Reward Function R. After the agent executes action a_t in state s_t, it obtains a reward r_t, which is designed to reflect the difference between task payoff and the travel cost incurred by assigned worker. The reward can be calculated as: $r = R(w, \tau) = \rho_\tau - \kappa \cdot \mathcal{T}_{w\tau}$.

Following the previous works [Li *et al.*, 2023; Wu *et al.*, 2024], we use Deep Q-Network (DQN) algorithm to optimize the RL policy of RTA problem.

4.3 Sim-to-Real Transfer with LLMs

Congestion Influence Subgraph. Identifying the Congestion Influence Subgraph (CIS), which represents the minimal traffic subgraph influencing the travel time from a worker's location to a task's location, is a challenging issue due to the complexity and dynamics of real-world road network. To address this issue, we propose a *Monte Carlo Graph Search-based Congestion Influence Subgraph Identify (MCGS-CI)* algorithm for CIS identification. Compared with traditional Monte Carlo Tree Search methods [Shi *et al.*, 2021], which rely on a tree structure for sequential decision-making, and face limitations in such scenarios due to redundant node expansions and insufficient information sharing between nodes, our approach explores a graph structure rather than a tree, enabling effective information sharing across nodes and reducing the number of required simulations [Xie *et al.*, 2024]. This design improves computational efficiency while maintaining high accuracy in identifying the CIS.

Our MCGS-CI algorithm starts with an initial graph $G =< V, E, U >$, where V represents all traffic intersections and E denotes road segments with weights U. The algorithm dynamically builds and updates the graph structure during the search process, using shared information from previously explored nodes to refine subsequent simulations. Specifically, it balances exploration and exploitation by leveraging the Upper Confidence Bound (UCB) mechanism to guide the selection of the most promising paths within the graph. The key steps of MCGS-CI algorithm are described as follow:

Initialization: The graph G is initialized with the worker's location l_w as the starting node and the task's location l_τ as the goal node. The weight $u_{ij} \in U$ of edge $e_{ij} \in E$ in G is initialized to:

$$u_{ij} = \log\left(1 + \frac{\ell_{ij}^{\eta} \cdot q_{ij}^{1-\eta}}{\nu_{ij}^{\eta} \cdot q_{\max}^{1-\eta}}\right), \tag{3}$$

where ℓ_{ij} is the length of road segment e_{ij}, and ν_{ij} denotes the historical average travel speed on e_{ij}. The term q_{ij} indicates the historical average traffic volume on e_{ij}, while $q_{\max}$ is the maximum capacity of e_{ij}. $\eta \in [0, 1]$ is a hyper-parameter that balances the travel time and congestion level.

1) Selection: Starting from l_w, the algorithm recursively selects the neighboring nodes with weight not less than the threshold σ (*i.e.,* $u_{ij} \geq \sigma$) based on the UCB formula $\mathcal{F}_{ucb}$, and the score of child node v can be calculated as:

$$\mathcal{F}_{ucb}(v) = \bar{S}_v + \alpha\sqrt{2\ln n/n_v}, \tag{4}$$

where $\bar{S}_v$ is the average score of child node v, n is the visit count of the v's parent node, n_v is the visit count of child node v, and α is a constant to balance exploration and exploitation.

2) Expansion: When the algorithm encounters an unexpanded node, it incorporates the neighboring nodes into the graph G.

3) Simulation: During the simulation phase, the algorithm starts from the current node and performs simulation actions to evaluate the feasibility of adding expanded nodes to the graph G. It first checks whether the expanded node is valid by determining if there exists a path through this node that can reach the goal node l_τ. If such a path exists, the algorithm quantifies the node's contribution to congestion by assigning a weight, and its corresponding edge and node are incorporated into the graph G.

4) Backpropagation: After the simulation phase, the results are propagated back through the explored graph, updating the weights and visit counts of all nodes and edges along the search path. Specifically, the weight of each edge on the path is incremented by 1 to reflect its contribution to the current simulation, and the visit count of each node is also increased by 1. This process refines future decision-making by reinforcing the importance of nodes that contribute to successful simulations, enabling the algorithm to prioritize promising paths in subsequent explorations.

Terminal Condition: The search process continues until a predefined number of iterations or l_τ is explored. The algorithm then outputs the resulting subgraph $G^c_{w\tau} \subseteq G$, consisting of the nodes and edges with the higher cumulative impact on travel time, as the identified CIS.

Prompt-based LLMs. Large Language Models (LLMs) have demonstrated superior zero-shot and few-shot reasoning abilities by adapting to unseen tasks through in-context learning [Alayrac *et al.*, 2022]. Building on these capabilities, we propose an LLMs-based approach to estimate travel time from worker's location to task's location in a real-world road conditions. The LLMs (*e.g.,* LLaMA) utilize the CIS, denoted as $G^c_{w\tau} = \langle V^c, E^c, U^c \rangle$, and structured contextual knowledge to achieve accurate travel time estimations. The LLMs' prompt template is defined as:

$$\langle \textbf{Question} \rangle \langle [\mathit{Context}] \rangle \langle \textbf{Response} \rangle, \tag{5}$$

where $\langle Question \rangle$ is the task description that LLMs require reasoning, such as estimating the travel time from worker's location to task's location, $\langle [Context] \rangle$ is structured knowledge about worker-task pairs, traffic conditions, and domain info (*e.g.,* weather, time of day, day of week), and $\langle Response \rangle$ specifies the generate format of LLMs. For each worker-task pair $(w, \tau) \in \mathcal{M}$, the context includes detailed information as follows:

$$\langle \textbf{CIS} \rangle \langle \textbf{Locations} \rangle \langle \textbf{Traffic} \rangle \langle \textbf{Weather} \rangle \langle \textbf{Day} \rangle \langle \textbf{Time} \rangle. \tag{6}$$

The LLMs utilize the prompt structured in Eq. (5) to infer the real-world travel time $\mathcal{T}_{w\tau}$:

$$\mathcal{T}_{w\tau} = LLM(Prompt(G^c_{w\tau}, \mathcal{D}_t, l_w, l_\tau)), \tag{7}$$

where $\mathcal{D}_t$ encodes contextual knowledge such as weather conditions, traffic density, and time of day. Then, we fuse $\mathcal{T}_{w\tau}$ and $\mathcal{D}_t$ with the state feature:

$$z_t = Concate(\mathcal{T}_{w\tau}, \mathcal{D}_t, s_t), \tag{8}$$

$$\mathbf{z}_t = ReLU(Linear(z_t)), \tag{9}$$

where $Concate$ denotes the concatenation operation, and z_t represents the temporary feature embedding.

Next, we use a forward model f_{ϕ^+} to predict the grounded state $\hat{s}_{t+1}$ in E_{real} as follow:

$$\hat{s}_{t+1} = f_{\phi^+}(s_t, a_t, \mathbf{z}_t), \tag{10}$$

where f_{ϕ^+} is implemented using a neural network and ϕ^+ can be optimized by minimizing the Mean Squared Error (MSE) loss:

$$\mathcal{L}(\phi^+) = MSE(\hat{s}^i_{t+1}, s^i_{t+1}), \tag{11}$$

where s^i_t, a^i_t, s^i_{t+1} are the ground trajectories sampled from real-world environment E_{real}.

Ultimately, we can predict the grounded action $\hat{a}_t$ that can enable the transition from state s_t to state $\hat{s}_{t+1}$ in E_{sim} by an inverse model h_{ϕ^-}:

$$\hat{a}_t = h_{\phi^-}(\hat{s}_{t+1}, s_t). \tag{12}$$

Compared to the forward model f_{ϕ^+}, which incorporates detailed dynamics knowledge for precise state transitions, $h_{\phi^-}(\hat{s}_{t+1}, s_t)$ can operate with lower computational complexity by relying solely on the observed states within the simulators. Thus, we can implement the inverse model using a neural network, and optimize it by minimizing the Categorical Cross-Entropy (CE) loss:

$$\mathcal{L}(\phi^-) = CE(\hat{a}^i_t, a^i_t), \tag{13}$$

where s^i_t, a^i_t, s^i_{t+1} are the predicted trajectories sampled from simulated environment E_{sim}.

4.4 Preference Learning from Human Feedback

To enable LLMs to estimate travel times in alignment with human preferences, *i.e.,* dynamically adjusting priorities such as time and distance, we propose a *Human Preference Learning Mechanism (HPLM)*. This mechanism introduces a preference vector $\mathbf{p} =< p_1, p_2, \ldots, p_n >$, where each $p \in [0, 1]$ represents the importance of a specific factor (*e.g.,* time, distance, and road type) derived from human feedback. Assuming a dominant preference is specified, such as p_k (*e.g.,* prioritizing distance), p_k can be set to 1, while other components of $\mathbf{p}$ are set to 0. Accordingly, the input prompt is refined to embed $\mathbf{p}$, ensuring that the LLMs' generations align with human preferences:

$$\widetilde{\mathcal{T}}_{w\tau} = LLM(Prompt(G^c_{w\tau}, \mathcal{D}_t, l_w, l_\tau, \mathbf{p})). \tag{14}$$

Given a preference dataset $D_p = \{(x_i, \mathcal{T}_{w_i\tau_i}, \widetilde{\mathcal{T}}_{w_i\tau_i})\}^M_{i=1}$, where x_i is the LLMs' input prompt, $\mathcal{T}_{w_i\tau_i}$ denotes the dispreferred output (abbreviated as $\mathcal{T}$), and $\widetilde{\mathcal{T}}_{w_i\tau_i}$ represents the preferred output (abbreviated as $\widetilde{\mathcal{T}}$), the optimal LLM policy π^* under the Bradley-Terry model [Rafailov *et al.*, 2023] satisfies the following preference probability:

$$p^*(\widetilde{\mathcal{T}} \succ \mathcal{T}|x) = \delta\left(\omega \log \frac{\pi^*(\widetilde{\mathcal{T}}|x)}{\pi_l(\widetilde{\mathcal{T}}|x)} - \omega \log \frac{\pi^*(\mathcal{T}|x)}{\pi_l(\mathcal{T}|x)}\right), \tag{15}$$

where δ denotes the logistic function, and ω is a scaling parameter that represents the divergence from the basic LLM policy π_l, which does not incorporate human preference adaptation.

To fine-tune the LLMs' preference policy π_p, we define a negative log-likelihood loss function based on the preference probability p^*:

$$\begin{aligned} &\mathcal{L}(\pi_p; \pi_l) \\ &= -\mathbb{E}_{D_p}\left[\log \delta\left(r^*_l(x, \widetilde{\mathcal{T}}) - r^*_l(x, \mathcal{T})\right)\right] \\ &= -\mathbb{E}_{D_p}\left[\log p^*(\widetilde{\mathcal{T}} \succ \mathcal{T}|x)\right] \\ &= -\mathbb{E}_{D_p}\left[\log \delta\left(\omega \log \frac{\pi_p(\widetilde{\mathcal{T}}|x)}{\pi_l(\widetilde{\mathcal{T}}|x)} - \omega \log \frac{\pi_p(\mathcal{T}|x)}{\pi_l(\mathcal{T}|x)}\right)\right], \end{aligned} \tag{16}$$

where $r^*_l(\cdot)$ is a reward model that scores LLMs' outputs to align with human preferences. This loss function guides the LLMs to assign higher probabilities to preferred responses while minimizing those for dispreferred ones. By iteratively updating the policy π_p through HPLM, the LLMs can adapt to nuanced preference shifts influenced by contextual factors such as day-of-week, time-of-day, or weather conditions, thereby generating more accurate and context-aware travel time estimates.

5 Experiments

5.1 Experimental Settings

In this section, we conduct extensive evaluations of HLM-Trans on two real-world datasets, comparing it with three state-of-the-art methods to demonstrate its effectiveness.

Parameters	Values
# of workers $\|W\|$ (CD)	100, 200, **300**, 500, 800
# of workers $\|W\|$ (XA)	50, 80, **100**, 200, 300
# of spatial tasks $\|\varGamma\|$ (CD)	5K, 8K, **10K**, 20K, 30K
# of spatial tasks $\|\varGamma\|$ (XA)	1K, 2K, **3K**, 5K, 10K
The deadline of tasks d_τ (CD)	1, 3, **5**, 8, 10 (min)
The deadline of tasks d_τ (XA)	1, 2, **3**, 4, 5 (min)
The large language models	ChatGLM3-6B, Qwen2-7B LLaMA2-7B, **LLaMA3-8B**

Table 1: Experimental Settings.

Datasets. The Chengdu (CD) and Xian (XA) datasets are selected to evaluate our proposed approaches. Each dataset comprises two main components: road network data and order data. The road network data is extracted from OpenStreetMap[1], representing the intricate road networks of Chengdu, with 36,630 nodes and 50,786 edges, and Xian, with 7,215 nodes and 9,643 edges. The order data, which forms the basis for task generation, is sourced from taxi orders collected by Didi Chuxing[2]. The Chengdu dataset contains 546,671 orders recorded from November 1 to 8, 2016, while the Xian dataset includes 102,674 orders recorded from October 1 to 8, 2016. Each order is a six-tuple, including order index, origin, destination, departure time, arrival time, and travel trajectory. In the experiments, we use the order's travel trajectory to calculate its real-world travel time, and the road network distance to calculate its simulated travel time. The experimental parameters are summarized in Table 1, where bold values indicate default parameter values.

Compared Methods. We compare the performance of HLMTrans with three approaches: dTrans [Wu *et al.*, 2024] (a direct transfer method which deploys the RLM trained in E_{sim} to make decisions in E_{real} without adaptation), MulT-TTE [Liao *et al.*, 2024] (a multi-faceted route representation learning model that captures spatial, attribute, and contextual information through sequential learning and transformer encoding, enabling more accurate travel time estimation), PromptGAT [Da *et al.*, 2024] (a prompt-based grounded action transformation model which leverages the inference abilities of LLMs to profile system dynamics and guide policy actions for mitigating the sim-to-real performance gap).

Metrics. We compare the performance of RL models trained in a simulation environment when transferred to real-world settings using commonly used task assignment metrics [Xia *et al.*, 2019; Li *et al.*, 2023]: average total revenue (*ATR*), average task completion ratio (*ACR*), and average elapsed time (*AET*). Additionally, we evaluate the travel time estimation performance of large language models using commonly used spatiotemporal prediction metrics [Wu *et al.*, 2024; Liao *et al.*, 2024]: *MAE* and *RMSE*.

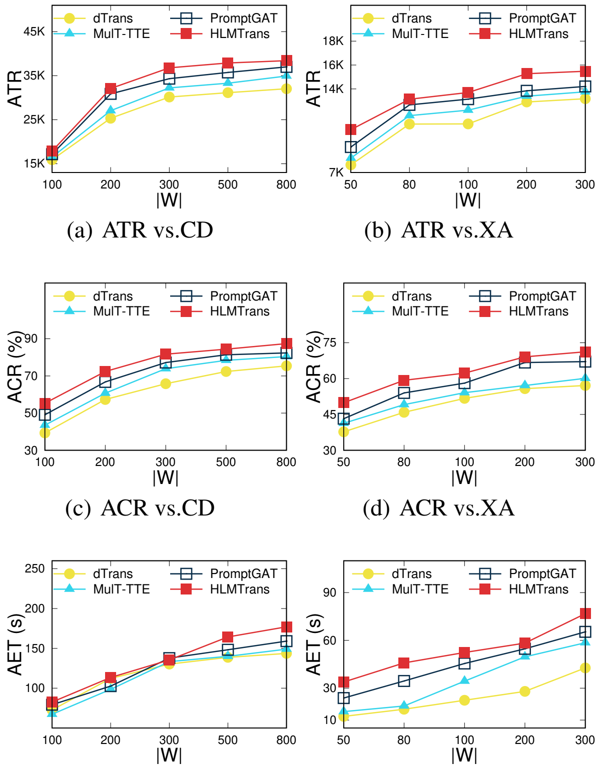
Environment. All machine learning methods are implemented with PyTorch 2.2 and Python 3.12, and trained with Intel i9-13900K@3.0GHz CPU, GTX3090 GPU and 32GB RAM. The platform ran on Ubuntu 18.04 LTS.

[1] https://www.openstreetmap.org

[2] https://www.didiglobal.com

Models	Chengdu		Xian	
	MAE	RMSE	MAE	RMSE
MulT-TTE	64.31	97.62	48.32	75.21
ChatGLM3-6B	15.90	23.18	15.23	22.44
Qwen2-7B	15.16	22.21	14.49	20.45
LLaMA2-7B	13.53	20.29	13.37	15.63
LLaMA3-8B	**12.26**	**19.35**	**11.43**	**13.64**

Table 2: Effect of language models.



(a) ATR vs.CD (b) ATR vs.XA

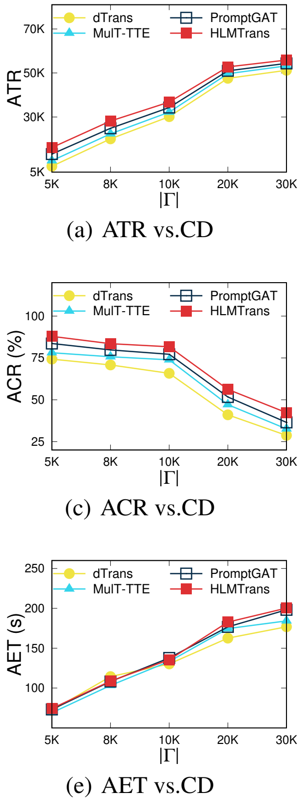
(c) ACR vs.CD (d) ACR vs.XA

(e) AET vs.CD (f) AET vs.XA

Figure 3: Effect of the number of workers $|W|$.

5.2 Analysis of Experimental Results

Effect of Language Models. The experimental results, as shown in Table 2, compare the travel time prediction errors of four large language models and the learning-based method MulT-TTE. The prompt for the large language models is set to:*"Please estimate the travel time in minute from nodes **worker location** to **task location** based on the following traffic perceptive information: on **day-of-week** at **time-of-day**, weather is **weather**, and the road network status (data format:[start node, end node, road type]) is **CIS**. Please only answer by replacing {value} in the format below: [travel time: {value}].".* It can be observed that the error metrics (*MAE* and *RMSE*) of the large language models are significantly lower than those of MulT-TTE, attributed to their robust zero-shot chain-of-thought reasoning capabilities. Among the models, the LLaMA series outperforms ChatGLM and Qwen in travel time estimation. Furthermore, larger parame-

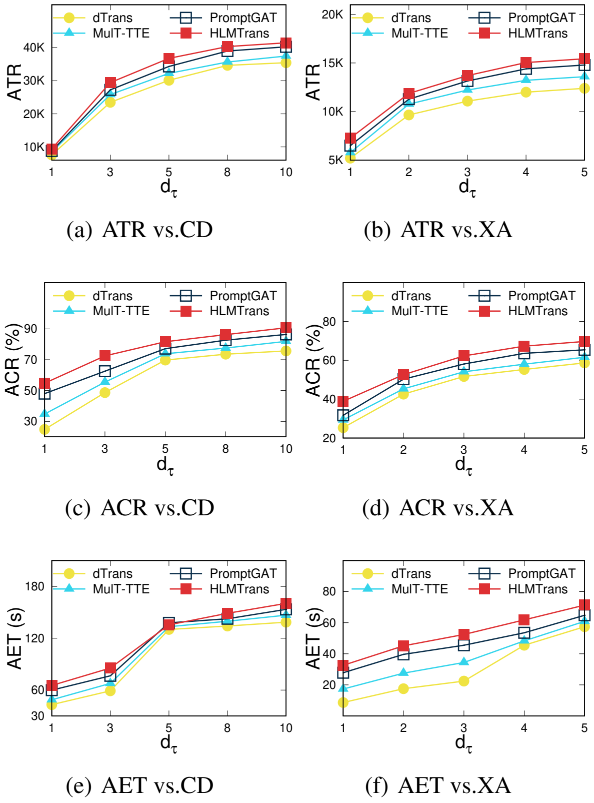


(a) ATR vs.CD (b) ATR vs.XA

(c) ACR vs.CD (d) ACR vs.XA

(e) AET vs.CD (f) AET vs.XA

Figure 4: Effect of the number of tasks $|\Gamma|$.



(a) ATR vs.CD (b) ATR vs.XA

(c) ACR vs.CD (d) ACR vs.XA

(e) AET vs.CD (f) AET vs.XA

Figure 5: Effect of the deadline of tasks d_τ.

ter models in the LLaMA series (*e.g.,* LLaMA3-8B) demonstrate superior performance compared to smaller parameter models (*e.g.,* LLaMA2-7B).

Effect of $|W|$. To evaluate the effect of worker number $|W|$ on the sim-to-real gap, we varied $|W|$ from 50 to 800 on both the CD and XA datasets. The experimental results, as shown in Fig. 3, indicate that as $|W|$ increases, all four methods exhibit a rapid initial improvement in total revenue and task completion ratio, followed by a gradual plateau. This trend arises because the increase in worker availability allows more tasks to be completed, but once the worker number surpasses a certain threshold, the limited number of tasks constrains further revenue growth. HLMTrans achieves the best performance on both datasets, with total revenue improvements of approximately 21.9% and 23.7% over dTrans on the CD and XA datasets, respectively. However, HLMTrans incurs slightly higher time costs compared to the other methods, primarily due to its ability to allocate workers more effectively for rapid task completion, which necessitates a higher number of model decisions.

Effect of $|\Gamma|$. Fig. 4 presents the experimental results with task number $|\Gamma|$ varying from 5K to 30K. As $|\Gamma|$ increases, total revenue and time costs show an upward trend, while the completion ratio declines. This is because an increase in task number leads to more tasks being completed, thereby increasing both revenue and time costs. However, due to the limited number of workers, a larger proportion of tasks expire without being assigned to available workers. Among the methods, HLMTrans consistently achieves the optimal performance, followed by PromptGAT, MulT-TTE, and dTrans.

Effect of d_τ. The experimental results in Fig. 5 illustrate the effect of varying task deadline d_τ from 1 to 10 on the CD and XA datasets. As the deadline increases, ATR, ACR, and AET all exhibit an upward trend. This is because an extended deadline allows more tasks to be completed, while simultaneously increasing the model's number of decisions. HLMTrans achieves significant improvements in total revenue compared to dTrans, with maximum gains of up to 25.4% on the CD dataset and 39.7% on the XA dataset.

6 Conclusion

In this paper, we propose HLMTrans, a human-guided LLMs-based sim-to-real transfer framework, designed to mitigate the performance gap when transferring RLMs trained in simulation environments to real-world task assignment scenarios. By leveraging the zero-shot reasoning capabilities of pre-trained LLMs and fine-tuning with human preference feedback, HLMTrans can improve the predictive accuracy of GAT's forward model, and effectively align with the system dynamics of real-world environments. Extensive experiments demonstrate that RLMs trained with the HLMTrans framework in simulation environments, benefiting from more accurate predictions of travel times, achieve superior performance in real-world settings compared to SOTA methods.

Acknowledgments

This work is supported by the following grants: NSFC Grants 62372416, 61972362, 62036010, 62325602, 62302460 and 62402453; HNSF Grant 242300421215; CPSF Grant 2022TQ0297.

References

[Alayrac *et al.*, 2022] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, and et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[Bousmalis *et al.*, 2018] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *ICRA*, pages 4243–4250, 2018.

[Bradley and Terry, 1952] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952.

[Christiano *et al.*, 2017] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, pages 4299–4307, 2017.

[Cully *et al.*, 2015] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[Cutler *et al.*, 2014] Mark Cutler, Thomas J. Walsh, and Jonathan P. How. Reinforcement learning with multi-fidelity simulators. In *ICRA*, pages 3888–3895, 2014.

[Da *et al.*, 2023] Longchao Da, Hao Mei, Romir Sharma, and Hua Wei. Uncertainty-aware grounded action transformation towards sim-to-real transfer for traffic signal control. In *CDC*, pages 1124–1129, 2023.

[Da *et al.*, 2024] Longchao Da, Minquan Gao, Hao Mei, and Hua Wei. Prompt to transfer: Sim-to-real transfer for traffic signal control with prompt learning. In *AAAI*, pages 82–90, 2024.

[Desai *et al.*, 2020] Siddharth Desai, Haresh Karnan, Josiah P. Hanna, Garrett Warnell, and Peter Stone. Stochastic grounded action transformation for robot learning in simulation. In *IROS*, pages 6106–6111, 2020.

[Fang *et al.*, 2018] Kuan Fang, Yunfei Bai, Stefan Hinterstoisser, Silvio Savarese, and Mrinal Kalakrishnan. Multi-task domain adaptation for deep learning of instance grasping from simulation. In *ICRA*, pages 3516–3523, 2018.

[Griffith *et al.*, 2013] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L. Isbell Jr., and Andrea Lockerd Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *NeurIPS*, pages 2625–2633, 2013.

[Hanna and Stone, 2017] Josiah P. Hanna and Peter Stone. Grounded action transformation for robot learning in simulation. In *AAAI*, pages 3834–3840, 2017.

[Jagadish *et al.*, 2024] Akshay K. Jagadish, Julian Coda-Forno, Mirko Thalmann, Eric Schulz, and Marcel Binz. Human-like category learning by injecting ecological priors from large language models into neural networks. In *ICML*, 2024.

[James *et al.*, 2019] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *CVPR*, pages 12627–12637, 2019.

[Ji *et al.*, 2023] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In *NeurIPS*, 2023.

[Lee *et al.*, 2023] Jiyoung Lee, Seungho Kim, Seunghyun Won, Joonseok Lee, Marzyeh Ghassemi, James Thorne, Jaeseok Choi, O.-Kil Kwon, and Edward Choi. Visalign: Dataset for measuring the alignment between AI and humans in visual perception. In *NeurIPS*, 2023.

[Li *et al.*, 2023] Yafei Li, Qingshun Wu, Xin Huang, Jianliang Xu, Wanru Gao, and Mingliang Xu. Efficient adaptive matching for real-time city express delivery. *IEEE Trans. Knowl. Data Eng.*, 35(6):5767–5779, 2023.

[Li *et al.*, 2024] Yafei Li, Yifei Li, Yun Peng, Xiaoyi Fu, Jianliang Xu, and Mingliang Xu. Auction-based crowdsourced first and last mile logistics. *IEEE Trans. Mob. Comput.*, 23(1):180–193, 2024.

[Li *et al.*, 2025] Yafei Li, Wei Chen, Jinxing Yan, Huiling Li, Lei Gao, and Mingliang Xu. Gradient-guided credit assignment and joint optimization for dependency-aware spatial crowdsourcing. In *AAAI*, pages 14301–14308, 2025.

[Liao *et al.*, 2024] Tianxi Liao, Liangzhe Han, Yi Xu, Tongyu Zhu, Leilei Sun, and Bowen Du. Multi-faceted route representation learning for travel time estimation. *IEEE Trans. Intell. Transp. Syst.*, 25(9):11782–11793, 2024.

[Liu *et al.*, 2024] Yu Liu, Qian Ge, Wei Luo, Qiang Huang, Lei Zou, Haixu Wang, Xin Li, and Chang Liu. Graphmm: Graph-based vehicular map matching by leveraging trajectory and road correlations. *IEEE Trans. Knowl. Data Eng.*, 36(1):184–198, 2024.

[Rafailov *et al.*, 2023] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[Shi *et al.*, 2021] Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi, and Nan Cao. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Trans. Vis. Comput. Graph.*, 27(2):453–463, 2021.

[Tang *et al.*, 2018] Kun Tang, Shuyan Chen, and Zhiyuan Liu. Citywide spatial-temporal travel time estimation using big and sparse trajectories. *IEEE Trans. Intell. Transp. Syst.*, 19(12):4023–4034, 2018.

[Vafa *et al.*, 2024] Keyon Vafa, Ashesh Rambachan, and Sendhil Mullainathan. Do large language models perform the way people expect? measuring the human generalization function. In *ICML*, 2024.

[Wang *et al.*, 2014] Yilun Wang, Yu Zheng, and Yexiang Xue. Travel time estimation of a path using sparse trajectories. In *KDD*, pages 25–34, 2014.

[Wang *et al.*, 2018] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. In *AAAI*, pages 2500–2507, 2018.

[Wang *et al.*, 2023] Yu Wang, Jingfei Wu, Xingyuan Hua, Chi Harold Liu, Guozheng Li, Jianxin Zhao, Ye Yuan, and Guoren Wang. Air-ground spatial crowdsourcing with UAV carriers by geometric graph convolutional multi-agent deep reinforcement learning. In *ICDE*, pages 1790–1802, 2023.

[Wu *et al.*, 2023] Jingda Wu, Yanxin Zhou, Haohan Yang, Zhiyu Huang, and Chen Lv. Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):14745–14759, 2023.

[Wu *et al.*, 2024] Qingshun Wu, Yafei Li, Guanglei Zhu, Baolong Mei, Jianliang Xu, and Mingliang Xu. Prediction-Aware Adaptive Task Assignment for Spatial Crowdsourcing . *IEEE Trans. Mob. Comput.*, 23(12):13048–13061, 2024.

[Xia *et al.*, 2019] Jinfu Xia, Yan Zhao, Guanfeng Liu, Jiajie Xu, Min Zhang, and Kai Zheng. Profit-driven task assignment in spatial crowdsourcing. In Sarit Kraus, editor, *IJCAI*, pages 1914–1920, 2019.

[Xie *et al.*, 2024] Yupeng Xie, Yuyu Luo, Guoliang Li, and Nan Tang. Haichart: Human and AI paired visualization system. *Proc. VLDB Endow.*, 17(11):3178–3191, 2024.

[Ye *et al.*, 2022] Yongchao Ye, Yuanshao Zhu, Christos Markos, and James J. Q. Yu. Cateta: A categorical approximate approach for estimating time of arrival. *IEEE Trans. Intell. Transp. Syst.*, 23(12):24389–24400, 2022.