UrbanLLM: Autonomous Urban Activity Planning and Management with Large Language Models

Yue Jiang[1,3], Qin Chao[1,3], Yile Chen[1*], Xiucheng Li[2], Shuai Liu[1], Gao Cong[1*], *
[1]Nanyang Technological University, Singapore,
[2]Harbin Institute of Technology(Shenzhen), China,
[3]DAMO Academy, Alibaba group, Singapore,
{yue013@e, chao0009@e, yile001@e, shuai004@e, gaocong@}ntu.edu.sg
lixiucheng@hit.edu.cn

Abstract

Location-based services play a critical role in improving the quality of our daily lives. Despite the proliferation of numerous specialized AI models within spatio-temporal context of location-based services, these models struggle to autonomously tackle problems regarding complex urban planing and management. To bridge this gap, we introduce **UrbanLLM**, a fine-tuned large language model (LLM) designed to tackle diverse problems in urban scenarios. UrbanLLM functions as a problem-solver by decomposing urban-related queries into manageable sub-tasks, identifying suitable spatio-temporal AI models for each sub-task, and generating comprehensive responses to the given queries. Our experimental results indicate that UrbanLLM significantly outperforms other established LLMs, such as Llama and the GPT series, in handling problems concerning complex urban activity planning and management. UrbanLLM exhibits considerable potential in enhancing the effectiveness of solving problems in urban scenarios, reducing the workload and reliance for human experts. Our code is available at: https://github.com/JIANGYUE61610306/UrbanLLM

1 Introduction

Location-based services are ubiquitous in urban spaces, supporting a range of scenarios from commuting assistance for travelers and daily activities for residents to event monitoring for city regulators. The diverse and substantial demand for location-based services has driven the development of specialized AI models tailored to specific tasks within spatio-temporal context. These tasks include (spatio-)temporal forecasting (Bai et al., 2020; Wu et al., 2020; Li et al., 2017), imputation (Cao et al., 2018; Liu et al., 2023), and anomaly detection (Goswami et al., 2023; Chen et al., 2022), as well as travel time estimation (Derrow-Pinion et al., 2021; Li et al., 2019), trajectory prediction (Ren et al., 2021; Chen et al., 2023), and POI recommendation (Wang et al., 2022; Lim et al., 2022), etc.

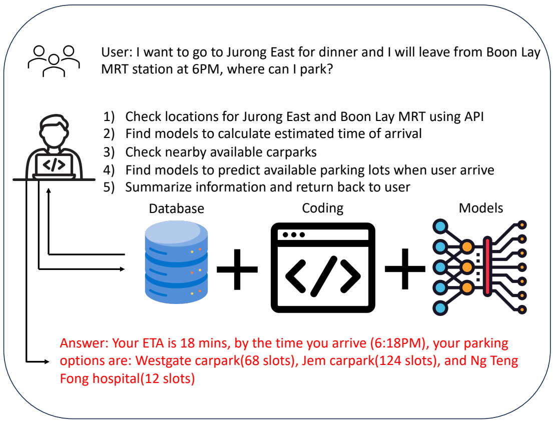

Figure 1: The process of solving a real-world problem in urban scenarios involving urban activity planning by human experts.

Despite the promising results of specialized AI models in addressing urban-related tasks with fixed input formats, these models are inadequate for handling complex queries in natural language that require strategic planning, reasoning, and collaboration among multiple specialized models across potentially various data modalities, such as GPS coordinates, addresses, and traffic records. As illustrated in Figure 1, to answer the given query, human experts need to decompose it into several sub-tasks, select and employ an appropriate combination of models for each sub-task, and synthesize the response based on model outputs. To enhance user experiences in location-based services, it is critical to design an autonomous and effective method for solving urban-related problems, particularly regarding complex urban activity planning and management.

Large language models (LLMs), such as Chat-

[*]Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 1810–1825
November 12-16, 2024

GPT [1], have attracted significant interest from both academia and industry (Manvi et al., 2023; Shen et al., 2023). LLMs have been applied in diverse domains, such as commerce, finance, healthcare, and the geospatial domain (Bommasani et al., 2021; Brown et al., 2020; Zhang et al., 2023b; Deng et al., 2023). Their exceptional capabilities in language comprehension and reasoning have positioned them as core modules in autonomous agents, where they function as problem solvers (Shen et al., 2023). However, conventional LLMs, as well as LLM agents built on them, encounter several challenges specific to our target. First, conventional LLMs, despite their intrinsic ability in reasoning and comprehension, possess limited geospatial knowledge about tasks and AI models within spatio-temporal context in urban scenarios (Manvi et al., 2023). Second, agents such as AutoGPT [2], AgentGPT [3], and BabyAGI [4] utilize tools for web search and code execution, which are not suited for location-based services. Moreover, they are designed to refine solutions for individual tasks in a recursive way, lacking the ability to holistically analyze and decompose urban-related problems for multiple specialized models. Third, while HuggingGPT (Shen et al., 2023) proposes to schedule text and image-related problems into sub-tasks and coordinate responses from corresponding models, it still relies on conventional LLMs as its backbones, thus inheriting their limitations in solving tasks in urban scenarios.

To tackle these challenges, we propose UrbanLLM, a novel urban foundation model to effectively decompose queries into sub-tasks and schedule specialized AI models within spatio-temporal context for each sub-task, thereby autonomously solving complex problems in urban scenarios. The idea is to align the problem-solving process with the established paradigms of LLMs, and enhance LLMs with the capability for urban activity planning and management through targeted fine-tuning. In this process, LLMs are equipped to serve as a universal interface capable of handling diverse tasks for various urban scenarios and producing appropriate responses.

Specifically, LLMs are initially fine-tuned using a structured template with high-quality examples that focus on spatio-temporal task decomposition and scheduling, thus augmenting the reasoning capabilities for the problems in targeted urban scenarios. Subsequently, the processing of new urban-related problems is conducted in three stages for inference phase. First, in the task analysis stage, a query of urban-related problems is effectively decomposed into a series of sub-tasks, each of which corresponds to a specific type of specialized AI models within spatio-temporal context. Next, in the model matching stage, the most suitable model is selected for each sub-task from a pool of candidate specialized AI models based on their descriptions. Finally, in the results generation stage, the selected models are executed to obtain output results, which are then organized into prompts to formulate a comprehensive response to the original query. By utilizing UrbanGPT, we facilitate the efficient and convenient resolution of urban-related problems without intensive manual efforts. Our contributions are summarized as follows.

- We propose UrbanLLM, the pioneering application of a fine-tuned LLM based on Llama-2-7B (Touvron et al., 2023a), to solve problems regarding complex urban activity planning and management. UrbanLLM is expected to improve in performance and adaptability as the community continues to develop AI models within spatial-temporal context.

- We develop an effective method for LLM instruction-tuning that enhances the reasoning capabilities in urban scenarios. Furthermore, we devise an autonomous pipeline to generate responses by minimizing the need for human intervention.

- Extensive experiments on real-world problems in urban scenarios demonstrate that our proposed UrbanLLM significantly outperforms other advanced LLMs, such as Llama-3-8B [5] and GPT-4o [6], by a substantial margin.

2 Related Work

Large language models (LLMs), due to their powerful capability in reasoning and comprehension, are increasingly utilized to assist in specific urban applications. For example, TrafficGPT (Zhang et al., 2023a) employs ChatGPT as a control agent to interact with various system components, such as

[1] https://platform.openai.com/docs/models
[2] https://github.com/Significant-Gravitas/Auto-GPT
[3] https://github.com/reworkd/AgentGPT
[4] https://github.com/yoheinakajima/babyagi
[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B
[6] https://platform.openai.com/docs/models/gpt-4o

databases, visualization and statistical tools, to perform basic analytical operations in traffic-related tasks. TravelPlanner (Xie et al., 2024) assessed the performance of Large Language Models (LLMs) functioning as travel planning agents for responding to travel-related queries under various constraints. The experimental results indicate that current LLMs, including GPT-4, achieved a success rate of only 0.6%. GeoGPT (Zhang et al., 2023b) utilizes ChatGPT to address similar analytical operations in Geographical Information Systems (GIS) domain. (Zhou et al., 2024) leverages LLMs to simulate roles such as planners and residents within a multi-agent framework to help urban land use and development planning. LLMob(Wang et al., 2024a) introduces a framework that considers individual activity patterns for urban mobility data generation. However, these studies typically rely on original LLMs and possess limited domain knowledge, which restricts their effectiveness in addressing challenges regarding complex urban planning and management.

To overcome such intrinsic limitations, various initiatives have deployed fine-tuned LLMs in targeted scenarios. For example, LLMlight (Lai et al., 2023) is fine-tuned based on Llama-2 to improve decision-making and policies in traffic signal control. TransGPT (Wang et al., 2024b) is trained on a corpus of examples from traffic prediction, public transportation, and autonomous driving, to address urban transportation tasks such as identifying traffic rules and signs. UrbanGPT (Li et al., 2024) aims to enhance spatio-temporal forecasting accuracy in a zero-shot setting by integrating a specialized decoder with an instruction-tuning paradigm. GeoLLM (Manvi et al., 2023) enhances urban regional questions such as population density and home value by fine-tuning LLMs with specifically designed templates. Moreover, several models focus more on text-related dimensions. PlanGPT (Zhu et al., 2024) is fine-tuned on a large corpus of urban planning regulations from numerous local governments in China, aimed at revising or generating texts for new regulations and evaluating planning documents. K2 (Deng et al., 2023) learns additional geospatial knowledge from a collection of geoscience text training corpus, enhancing NLP tasks such as summarization and text classification, specifically for the geoscience domain. Unlike all these studies, we are the first to equip LLMs with the capability to decompose queries regarding complex urban planning and management into manageable components that align with specialized AI models within spatio-temporal context, thereby autonomously tackling diverse urban-related problems.

3 UrbanLLM

Our proposed UrbanLLM is structured into two phases: the learning phase and the inference phase. In the learning phase, we fine-tune UrbanLLM, employing meticulously crafted examples from our constructed self-instruct dataset. This dataset includes examples that contain reasoning hints, various types of backbone spatio-temporal AI models, diverse queries, and the corresponding decomposed sub-tasks. This fine-tuning process effectively enhances the comprehension and reasoning capabilities of UrbanLLM tailored to address our targeted objectives in urban scenarios.

The inference phase consists of three stages: spatial-temporal analysis, model matching, and results generation. In the spatio-temporal analysis stage, UrbanLLM receives queries that are organized within the same prompt template used during the training stage with new queries, enabling it to effectively decompose the query into a series of types of spatio-temporal tasks, owing to enhancements achieved through fine-tuning. The model matching stage involves pairing each identified spatio-temporal task with suitable AI models and selecting the most appropriate one. Finally, in the results generation stage, the selected spatio-temporal models are executed, and their outputs are formulated into a prompt to produce the response to the query. The overall process for the two phases is depcited in Figure 2.

3.1 Urban Activity Planning Learning

The learning phase of UrbanLLM is designed to endow LLMs with the capability to comprehend knowledge for processing different types of spatio-temporal tasks, facilitating the decomposition of queries in urban scenarios into these tasks. This is achieved through the rigorous formulation of prompts and instructions, followed by the fine-tuning of a Llama2-7B model (Figure 2 left).

Specifically, we recorded 170 seed prompts in Singapore from human experts and employed the self-instruct method (Wang et al., 2023) to generate additional 15,249 training examples and 1,694 evaluation examples using `GPT-4-1106-preview`. The training examples are utilized for instruction-tuning

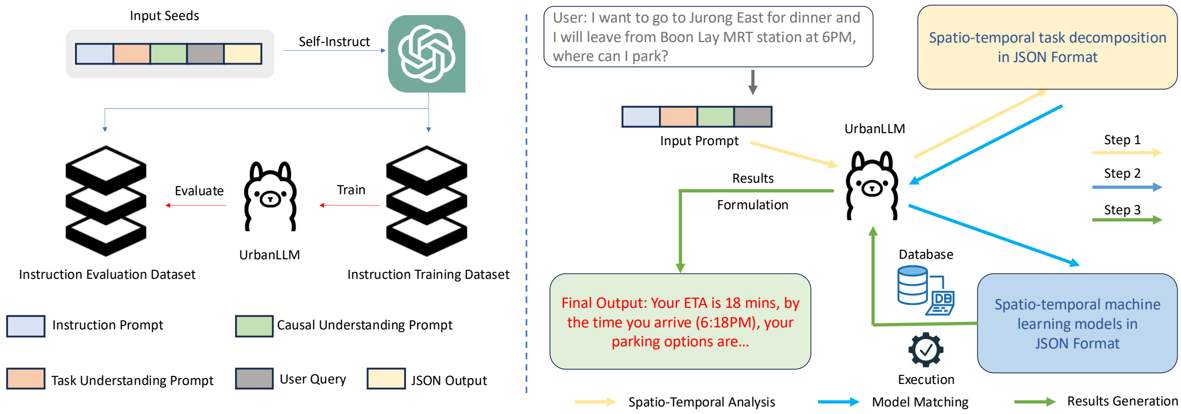

Figure 2: The overall process of the proposed UrbanLLM framework. The urban activity planning learning phase is on the left and the inference phase is on the right.

UrbanLLM in an unsupervised learning paradigm, while the evaluation examples serve to evaluate the performance of the all compared models in the experiments. Each example in the dataset comprises an instruction part and a QA part, with a sample showcased in Figure 3.

The instruction part features three key components: scenario formulation, task understanding, and causal understanding. Scenario formulation specifies the requirements of translating task decomposition into a machine-understandable format (i.e., JSON), and defines 13 types of potential spatio-temporal sub-tasks and their associated arguments in the format, resulting in a total of 34 task combinations (listed in Appendix A). Inspired by HuggingGPT (Shen et al., 2023), to demonstrate the dependency relationship among tasks, we use the `'dep'` field to denote the task ID of a previous task upon which the current task relies, and the `<resource>-task_id` to indicate the output from the previous task used as the input for the current task. Task understanding provides detailed explanations on each type of spatio-temporal task (listed in Appendix C), enabling UrbanLLM to understand their functions for subsequent query decomposition. Causal understanding identifies the connections and causal relationships among specific task combinations, allowing UrbanLLM to grasp and apply the underlying logic. The QA part includes the specific queries regarding complex urban activity planning and management, and its response adhering to the JSON format specified in the instruction section. Subsequently, we utilized the constructed training examples to fine-tune the Llama-2-7B model with QLoRA (Liu et al., 2021; Touvron et al., 2023b), a technique that significantly reduces the computational cost of fine-tuning. Through the fine-tuning process, UrbanLLM learns to adeptly schedule urban activity planning and management for diverse and complex queries in urban scenarios.

3.2 Spatio-Temporal Analysis

In the first stage of the inference phase, UrbanLLM employs the template in the training phase to craft a prompt. This prompt is then fed into the fine-tuned model to produce JSON outputs which present the results of spatio-temporal task analysis. Specifically, these outputs provide structured information on the dependencies and interactions among the 13 defined sub-tasks, outlining the task decomposition necessary to address the given query. Through the implementation of fine-tuning in Section 3.1, UrbanLLM gains the knowledge to decompose the query in urban scenarios into manageable spatio-temporal sub-tasks, each associated with its specialized AI models. This decomposition is critical for solving complex problems regarding urban activity planning and management, which typically exceed the capability of single models. Finally, the generated JSON outputs are utilized in the subsequent stages of model matching and results generation.

3.3 Model Matching

In the model matching stage, the chat log from previous interactions is used as input for UrbanLLM to select the appropriate model for each sub-task. To facilitate this process, we have organized a comprehensive model zoo consisting of more than 50 recent spatio-temporal AI models and tools (some are presented in Appendix D). Each model is associated with descriptions that cover model information

Urban Activity Planning Learning Phase

<s>[INST]You are UrbanLLM, a Large language model for urban activity that decompose user input into a list of spatio-temporal tasks with the following JSON format: {task: task_name, id: task_id, dep: dependency_task_ids, args: {domain: string, location_name_list: list, location_gps_list: list, time: time, input: sequence, service_no: int, bus_stop: string, task_specific: list}}. Task-specific information is stored using a string list and is used in specific models or tools as arguments. The dep field denotes the ID of the previous task that the current task relies. A dep field value of '-1' indicates that the current task does not rely on other tasks and can be executed immediately, otherwise always execute the previous task first. The executed output, such as the generated location, time, sequence, or POIs from the dependency task is marked as <resource>-task_id. This resource will be used in the next task. The spatio-temporal tasks must be selected from the following options: {long_time_series_prediction, time_series_prediction, event_prediction, trajectory_completion, trajectory_prediction, time_series_anomaly_detection, time_series_imputation, arrival_time_estimation, trajectory_forecasting, map_mapping, recommendation, spatial_relationship_inference, bus_arrival, taxi_availability}. For your better understanding, here is the explanation of each spatio-temporal task: 1) Long Time Series Prediction: This task involves forecasting future values in a time series over a long horizon. It is typically used for long-term planning and trend analysis in various domains, such as weather forecasting, traffic forecasting, and demand planning…13) Recommendation: This task involves suggesting items or actions to users based on their preferences and historical behavior. Applications include recommending points of interest, routes, or services in urban planning. Please note that there exists a logical connection and order between the tasks. 1) If user input mentioned some specific locations/POIs, usually map mapping task should be included in the answer, otherwise if user is only asking the situation around all Singapore, map mapping task should not be included. 2) If user do not specify the arrival time, usually estimated arrival time task should be included. 3) Include tasks to predict weather and PM2.5 according to user input if user is going to outdoor activities. 4) When recommendation or taxi_availability task is included, usually map mapping task should be included as fundamental task. In case the user input cannot be parsed, an empty JSON response should be provided. Please provide a task analysis JSON response based on the given question. Q: I'm at the IMM Outlet Mall now. How can I find the closest taxi stand?[/INST] A: [{task: recommendation, id: 0, dep: [1], args: {location_gps_list: <resource>-1, task_specific: ['taxi_stand']}}, {task: map_mapping, id: 1, dep: [-1], args: {location_name_list: ['IMM Outlet Mall']}}].</s>

Urban Activity Planning Inference Phase

Spatio-Temporal Analysis: Instruction Prompt + Task Understanding Prompt + Causal Understanding Prompt + Please provide a task analysis JSON response based on the given question. Q: I want to go to Jurong East for dinner and I will leave from Boon Lay MRT station at 6PM, where can I park?

JSON Output. A: [{task: time_series_prediction, id: 0, dep: [1], args: {location_gps_list: <resource>-2, time: <resource>-1, input: history_steps, domain: 'parking'}}, {task: arrval_time_estimation, id: 1, dep: [2], args: {location_gps_list: <resource>-2}}, {task: map_mapping, id: 2, dep: [-1], args: {location_name_list:['Jurong East', 'Boon Lay MRT']}}]."

Model Matching: {Chat log}+Given the chat log and the generated JSON answer, you serve as UrbanLLM, a large language model for autonomous urban activity planning, please match the most appropriate model from a spatio-temporal model zoo to generate results. In this stage, UrbanLLM should output the model_id from spatio-temporal model zoo and the reason to choose this model in a JSON format: {"id": "id", "reason": "detailed reason for the choice"}. Please select one model from the model zoo for each task in the JSON answer. Models in spatio-temporal model zoo and their descriptions are listed here: {{ Model Zoo }}.

Results Generation: {Chat log}+Given the chat log and the generated results, you serve as UrbanLLM, a large language model for autonomous urban activity planning, please refer to the user query, 'Q: I want to go to Jurong East for dinner and I will leave from Boon Lay MRT station at 6PM, where can I park?', answer the questions and state your {generated results} clearly.

Final Output: Answer: Your ETA is 18 mins, by the time you arrive (6:18PM), your parking options are: 1) Westgate carpark(68 slots), 2) Jem carpark(124 slots), and 3) Ng Teng Fong hospital(12 slots)

Instruction Prompt | Causal Understanding Prompt | Task Understanding Prompt | User Query | JSON Output | Dedicate Model Generated Results

Figure 3: A sample of detailed prompt template used in the learning phase and the process of a new query solved by UrbanLLM in the inference phase.

on the addressed problem settings and scenarios, as well as the data types or formats in which the model has been tested or evaluated. Utilizing these descriptions, UrbanLLM is prompted to match each sub-task identified during the spatio-temporal analysis stage with a suitable model from the model zoo. Finally, UrbanLLM outputs the selection of the most suitable spatio-temporal model for each sub-task in JSON format, ensuring accurate and efficient task execution.

3.4 Results Generation

In the results generation stage, UrbanLLM executes the selected spatio-temporal models determined by the JSON output from the model matching stage. The execution sequence follows the dependencies outlined in the JSON output from the spatio-temporal analysis stage. For each model, UrbanLLM retrieves necessary inputs either directly from the specified arguments, or from the outputs of previously executed models where dependencies exist. When the last model in the execution sequence is accomplished, the results are aggregated and compiled into a prompt that is then processed by UrbanLLM to generate the response. In this way, we ensure that the spatio-temporal models are logically executed, producing a final response to the initial query.

4 Experiments and Results Discussion

In this section, we compare UrbanLLM with several strong LLM baselines, such as Llama-3 and GPT-4o, to demonstrate the urban activity planning and management capability and superior performance of UrbanLLM after instruction fine-tuning. We then include a case study to visually illustrate the differences between UrbanLLM and GPT-4o. Furthermore, we conduct an ablation study to test the contribution of each component in UrbanLLM.

4.1 Experimental Setup

Datasets. We developed our UrbanLLM based in Singapore where extensive urban data sources are available. We recorded 170 seed prompts and applied the self-instruct method to generate 15294 training examples and 1694 evaluation examples using `GPT-4-1106-preview`. The training examples contains 4787 simple queries, which require a single model to derive the results, and 10507 complex requests, which require coordinating among multiple models for problem-solving. The evaluation examples contain 427 simple queries and 1267 complex queries. In addition to the self-instruct evaluation dataset, we have further constructed a human-annotated dataset consisting of 200 queries with corresponding responses focused on scenarios concerning complex urban activity planning and management, to rigorously evaluate UrbanLLM's performance. The data sources which serve as inputs for the specialized spatio-temporal AI models employed in the results generation stage are retrieved from the Singapore Open Data API, which provides access to over 4000 datasets from 69 government agencies. This API offers diverse spatio-temporal data sources from various domains, such as bus locations, POIs, passenger flow, car park availability, precipitation records, and PM2.5 levels.

Baselines. We apply our UrbanLLM, and several LLMs serving as baselines in the inference phase to evaluate their performances:

- **Llama-2-7B**: Llama-2-7B is a open-source LLM developed by Meta AI with 7 billion parameters.
- **Vicuna-7B-v1.5**[7]: Vicuna-7B-v1.5 is a open-source LLM based on Llama-2-7B with additional fine-tuning and supporting 16k context length.
- **Llama-3-8B**: Llama-3-8B is the latest model in the Llama series from Meta AI, which features an expanded architecture with 8 billion parameters. This model offers further enhancements in processing power and language comprehension.
- **GPT-3.5**[8]: GPT-3.5 model is a chatbot-based LLM (`gpt-3.5-turbo-0613`) developed by OpenAI. As the model is unavailable, we execute the inference phase using its API.
- **GPT-4o**: GPT-4o is a more advanced iteration of the GPT series after GPT-3.5. Similarly, we we execute the inference phase using its API.

Evaluation Metrics. We employ four metrics: accuracy, precision, recall, and F1 score, to evaluate the performance for each evaluation example and report the weighted average result. Accuracy is calculated as the proportion of predicted examples (including sub-tasks and their dependencies) that exactly match the ground truth among the total number of evaluated examples. Precision, recall, and F1 score are computed at a macro level for each example, specifically measuring sub-task predictions. More details of these evaluation metrics can be found in Appendix E.

Implementations. UrbanLLM is fine-tuned on the training examples for 5 epochs on a Linux workstation with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz and 8 32GB Tesla V100 GPU. We used 4-bit quantization (Liu et al., 2021) to obtain a more compact model representation, and low rank adaptation (LoRA) (Touvron et al., 2023b) to reduce the number of trainable parameters and decrease the GPU memory requirements. We set LoRA attention dimension to be 64 and initial learning rate to be 2e-4 with Adam optimizer.

4.2 Performance of UrbanLLM

To evaluate the analytical capabilities of LLMs, we report the metrics for all evaluated scenarios across all compared models in Table 1. In addition, we present their detailed performance in both simple and complex real-world examples in Table 2 and Table 3, respectively. The best result for each evaluation metric is highlighted in bold and the second best result is highlighted with an underline.

In these evaluation examples, we generally observe that baseline LLMs show excellent performance in completing both the model matching and results generation stages, echoing the observations from previous research(Shen et al., 2023) (detailed case studies demonstrated in Appendix B). However, a notable deficiency arises in the spatio-temporal analysis stage for the baseline LLMs due to the limited urban-specific training corpus. During this critical stage, baseline LLMs frequently encounter hallucination issues, such as generating non-existent tasks. Since this stage is crucial in translating queries into corresponding spatio-temporal sub-tasks to be solved using methods from the model zoom, we observe several LLMs collapse for the metrics. In contrast, our UrbanLLM significantly outperforms all baseline models on all evaluation metrics. Specifically, UrbanLLM achieved an overall accuracy of 68.3%, with 95.78% accuracy in single-task scenarios and 59.08% in complex real-world problems. By comparison, GPT-4o, the next best-performing model, managed only around 50% accuracy in spatio-temporal analysis, with other baseline models struggling to effectively complete task decomposition.

[7]https://huggingface.co/lmsys/vicuna-7b-v1.5-16k
[8]https://platform.openai.com/docs/models/gpt-3-5-turbo

This demonstrates UrbanLLM's superior capability in handling the intricate demands of urban scenarios.

Table 1: Evaluation for Spatio-Temporal Task Analysis

	Accuracy	Precision	Recall	F1
Llama2-7b	0.18%	10.52%	8.75%	9.18%
Vicuna-7b-v1.5	8.44%	14.08%	13.89%	13.95%
Llama3-8b	5.31%	12.96%	15.50%	13.08%
GPT-3.5	17.95%	23.25%	22.35%	22.54%
GPT-4o	49.99%	55.31%	54.42%	54.63%
UrbanLLM	**68.30%**	**80.05%**	**79.26%**	**79.49%**
% Improve	36.63%	44.73%	45.64%	45.50%

Table 2: Evaluation for Spatio-Temporal Single-Task Analysis

	Accuracy	Precision	Recall	F1
Llama2-7b	0.47%	0.57%	0.57%	0.57%
Vicuna-7b-v1.5	33.26%	33.26%	33.26%	33.26%
Llama3-8b	15.46%	17.51%	21.19%	17.97%
GPT-3.5	13.58%	13.70%	13.74%	13.71%
GPT-4o	67.44%	68.56%	68.60%	68.57%
UrbanLLM	**95.78%**	**96.78%**	**96.84%**	**96.80%**
% Improve	42.02%	41.16%	41.17%	41.17%

Table 3: Evaluation for Spatio-Temporal Multi-Task Analysis

	Accuracy	Precision	Recall	F1
Llama2-7b	0.00%	13.80%	11.44%	12.01%
Vicuna-7b-v1.5	0.08%	7.62%	7.36%	7.45%
Llama3-8b	1.81%	11.36%	13.52%	11.37%
GPT-3.5	19.35%	26.40%	25.20%	25.45%
GPT-4o	40.13%	50.89%	49.68%	49.97%
UrbanLLM	**59.08%**	**74.47%**	**73.40%**	**73.71%**
% Improve	47.22%	46.33%	47.75%	47.51%

We also evaluate the performance of UrbanLLM on a human-annotated dataset from 5 spatio-temporal domain experts, as detailed in Table 4. The results demonstrate that UrbanLLM continues to exhibit substantial improvements over all baseline models. Furthermore, the performance metrics on the human-annotated dataset are comparable to the datasets presented in Table 3, which are generated based on the self-instruct method. This consistency in performance suggests that our self-instruct dataset is well-constructed and effectively representative of real-world scenarios, thereby validating the robustness and reliability of UrbanLLM in addressing problems regarding urban activity planning and management.

4.3 Ablation Study

Based on the superiority of UrbanLLM over other established LLMs, we have validated the effectiveness of the training phase. To further demonstrate the contributions of the structured organization of our instructive prompts, we conduct an ablation study on UrbanLLM by removing different components within the spatio-temporal analysis stage. To this end, we define three variants of UrbanLLM as follows:

Table 4: Evaluation on Human Annotated Dataset

	Accuracy	Precision	Recall	F1
Llama2-7b	0.00%	14.75%	14.00%	14.21%
Vicuna-7b-v1.5	0.00%	5.37%	5.08%	5.18%
Llama3-8b	4.00%	13.77%	17.67%	14.67%
GPT-3.5	30.50%	37.22%	36.88%	36.99%
GPT-4o	40.50%	49.32%	48.71%	48.89%
UrbanLLM	**55.00%**	**74.79%**	**74.92%**	**74.83%**
% Improve	35.80%	51.64%	53.80%	53.06%

Table 5: Ablation Study of UrbanLLM

	Accuracy	Precision	Recall	F1
UrbanLLM	68.30%	80.05%	79.26%	79.49%
w/o SF	57.02%	77.88%	76.98%	77.24%
w/o TU	61.63%	78.92%	78.07%	78.31%
w/o CU	62.69%	79.10%	78.26%	78.50%

- **w/o SF**: The scenario formulation component is removed from the prompts in the spatio-temporal analysis stage.
- **w/o TU**: The task understanding component is removed from the prompts in the spatio-temporal analysis stage.
- **w/o CU**: The causal understanding component is removed from the prompts in the spatio-temporal analysis stage.

The results of different model variants on the four metrics are presented in Table 5. We observe that the removal of each component leads to a decline in performance across all metrics, with the scenario formulation component having the most significant impact. This is likely because the scenario formulation component provides the task scope and the definition of arguments, which are foundations for accurate task decomposition. Moreover, UrbanLLM, which includes all the components in prompts, consistently outperforms all other variants. This validates the effectiveness of the integration of scenario formulation, task understanding, and causal understanding in inputs to solve urban-related problems. The ablation study confirms that each component is beneficial in model's ability in complex urban activity planning and management.

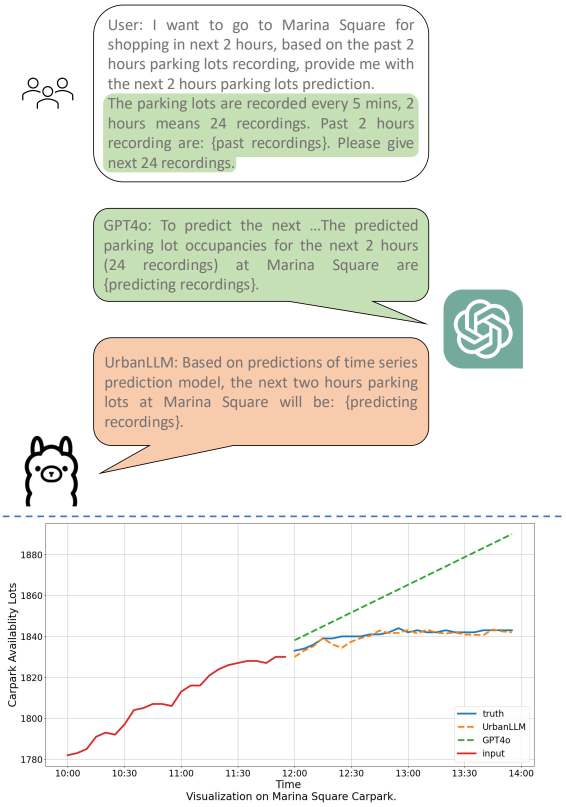

Figure 4: Visualization of responses and results of parking lot occupancy prediction for Marina Square Carpark. GPT-4o needs additional input data highlighted in green for prediction, while UrbanLLM retrieves corresponding data automatically and produces more accurate prediction.

4.4 Visualization and Generalization

We further demonstrate the effectiveness of UrbanLLM through a case study focusing on real-world carpark availability prediction problem. In this study, we compare UrbanLLM with the latest GPT-4o model, and their respective responses and predictions are presented in Figure 4. By leveraging a dedicated time series prediction method tailored for the parking domain, UrbanLLM provides robust and accurate predictions. In contrast, GPT-4o, which lacks the capability to to execute specialized spatio-temporal models, produce inaccurate predictions. This comparison shows that UrbanLLM's pipeline yields more reliable outcomes, thereby validating its superior performance in complex urban scenarios.

Although initially designed to address urban-related problems specific to Singapore, UrbanLLM are found to exhibit reasonable performance in generalizing to scenarios in other cities, owing to the zero-shot capabilities inherent in LLMs. As illustrated in Figure 5, UrbanLLM effectively decomposes urban-related problems in cities such as Beijing and New York City into relevant spatio-temporal sub-tasks. This generalization ability demonstrates UrbanLLM's versatility and robustness, making it a valuable tool for urban activity planning and management across urban environments.

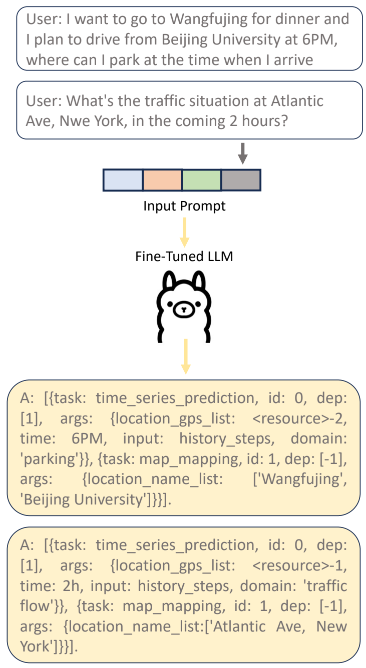

Figure 5: Demonstration of generalization ability across cities in UrbanLLM. The upper part is sample user queries in Beijing and New York City, and the lower part presents the resonable outcomes of spatio-temporal task decomposition.

5 Conclusions

In this study, we introduce UrbanLLM, a fine-tuned LLM developed to enhance the ability to perform autonomous urban activity planning management. After fine-tuning on a corpus of examples of problems in urban scenarios, UrbanLLM learns to decompose new queries into sub-tasks and identifies appropriate spatio-temporal AI models for each sub-task, thereby enhancing the accuracy of urban planning and the efficiency of management processes. Operating through both the learning and the inference phase and three meticulously designed stages, spatio-temporal task analysis, model matching, and results generation, UrbanLLM functions as a pipeline to achieve the problem-solving process and produce the response to the given query. Our experimental results demonstrate that Urban-

LLM significantly outperforms other LLM models, including Llama-3 and the GPT-4o, in the context of urban activity planning and management tasks by a large margin.(Manvi et al., 2023)

Limitations

Despite the promising results demonstrated by UrbanLLM in urban activity planning and management, several limitations need to be acknowledged. **1) Dependence on Pre-trained Models.** UrbanLLM relies heavily on the performance and capabilities of the underlying pre-trained Llama-2-7B model. While fine-tuning has enhanced its suitability for urban planning tasks, inherent limitations of the base model, such as insufficient geospatial understanding of the target city or specific urban contexts, may still affect outcomes. **2) Generalization Issues.** The fine-tuning process, while extensive, is based on a specific set of training data and scenarios. This means that UrbanLLM might not generalize well to urban tasks or environments significantly different from those it was trained on. Unexpected urban phenomena or novel planning and management problems may not be adequately addressed by the model. **3) Resource Intensity.** Although UrbanLLM reduces the need for continuous human intervention, the initial setup and fine-tuning process are resource-intensive. Additionally, as the number of spatio-temporal tasks increases, the self-instruct and fine-tuning process must be repeated, which in turn increases the resource cost. Addressing these limitations in future work will involve enhancing the model's robustness and geospatial knowledge of the target city, expanding its training datasets to include a more diverse range of scenarios, and developing more efficient fine-tuning techniques. Overcoming these challenges will maximize UrbanLLM's effectiveness in real-world urban planning applications.

Acknowledgement

This study is supported under Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6776–6786.

Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. 2022. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 3621–3633. PMLR.

Yuqi Chen, Hanyuan Zhang, Weiwei Sun, and Baihua Zheng. 2023. Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 829–842. IEEE.

Cheng Deng, Tianhang Zhang, Zhongmou He, Qiyuan Chen, Yuanyuan Shi, Le Zhou, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. 2023. Learning A foundation language model for geoscience knowledge understanding and utilization. *CoRR*, abs/2306.05064.

Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. 2021. ETA prediction with graph neural networks in google maps. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3767–3776. ACM.

Mononito Goswami, Cristian I. Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. 2023. Unsupervised model selection for time series anomaly detection. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. 2023. Large language models as traffic signal control agents: Capacity and opportunity. *CoRR*, abs/2312.16044.

Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. 2019. Learning travel time distributions with deep generative model. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1017–1027. ACM.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Graph convolutional recurrent neural network: Data-driven traffic forecasting. *CoRR*, abs/1707.01926.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. Urbangpt: Spatio-temporal large language models. *CoRR*, abs/2403.00813.

Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. 2022. Hierarchical multi-task graph recurrent network for next POI recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1133–1143. ACM.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Preprint*, arXiv:2107.13586.

Shuai Liu, Xiucheng Li, Gao Cong, Yile Chen, and Yue Jiang. 2023. Multivariate time-series imputation with disentangled temporal representations. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David B. Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *CoRR*, abs/2310.06213.

Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 1410–1419. ACM.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. 2022. Spatial-temporal interval aware sequential POI recommendation. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pages 2086–2098. IEEE.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, and Chuan Xiao. 2024a. Large language models as urban residents: An LLM agent framework for personal mobility generation. *CoRR*, abs/2402.14744.

Peng Wang, Xiang Wei, Fangxu Hu, and Wenjuan Han. 2024b. Transgpt: Multi-modal generative pre-trained transformer for transportation. *CoRR*, abs/2402.07233.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Siyao Zhang, Daocheng Fu, Zhao Zhang, Bin Yu, and Pinlong Cai. 2023a. Trafficgpt: Viewing, processing and interacting with traffic foundation models. *CoRR*, abs/2309.06719.

Yifan Zhang, Cheng Wei, Shangyou Wu, Zhengting He, and Wenhao Yu. 2023b. Geogpt: Understanding and processing geospatial tasks through an autonomous GPT. *CoRR*, abs/2307.07930.

Zhilun Zhou, Yuming Lin, and Yong Li. 2024. Large language model empowered participatory urban planning. *CoRR*, abs/2402.01698.

He Zhu, Wenjia Zhang, Nuoxian Huang, Boyang Li, Luyao Niu, Zipei Fan, Tianle Lun, Yicheng Tao, Junyou Su, Zhaoya Gong, Chenyu Fang, and Xing Liu. 2024. Plangpt: Enhancing urban planning with tailored language model and efficient retrieval. *CoRR*, abs/2402.19273.

A Planning and Management Tasks Combinations

We list the 34 planning and management tasks combinations and their corresponding spatio-temporal task decomposition answer in JSON format as follows. We use an even distribution across a total of 34 task combinations. These task combinations are generated from 13 sub-tasks. More fundamental subtasks, such as map mapping and recommendation, occur more frequently in task combinations. We will continue to generate high-quality data using our instruction template and fine-tune the model on the extended dataset, and post the latest results on Github: https://github.com/JIANGYUE61610306/UrbanLLM.

1) Q: I want to go to Jurong East for dinner and will arrive at 7PM, where can I park? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1, input: history_steps, domain: '
    parking'}}, {task: map_mapping, id: 1,
    dep: [-1], args: {location_name_list:['
    Jurong_East']}}].
```

2) Q: I want to go to Jurong East for dinner and I plan to drive from lake side station at 6PM, where can I park at the time when I arrive? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-2, time: <resource>-1, input:
    history_steps, domain: 'parking'}}, {task
    : arrval_time_estimation, id: 1, dep:
    [2], args: {location_gps_list: <resource
    >-2}}, {task: map_mapping, id: 2, dep:
    [-1], args: {location_name_list:['Jurong␣
    East', 'lake_side']}}].
```

3) Q: Do you have any bicycle parking location recommended nearby Lake Garden? Answer:

```
[{task: recommendation, id: 0, dep: [1], args:
     {location_gps_list: <resource>-1},
    task_specific:['bycycle_parking']}, {task
    : map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Lake_Garden']}}].
```

4) Q: I would like to go to Starbucks@J-walk, is it here? 3 Gateway Dr. Unit 02-04/04A Westgate, Singapore 608532. Answer:

```
[{task: spatial_relationship_infer, id: 0,
    dep: [1], args: {location_gps_list: <
    resource>-1}}, {task: map_mapping, id: 1,
     dep: [-1], args: {location_name_list:['
    Starbucks@J-walk', '3_Gateway_Dr._
    #02-04/04A_Westgate,_Singapore_608532'
    ]}}].
```

5) Q: I am waiting at the bus stop: 83139. When will be the next No. 15 bus coming? Answer:

```
[{task: bus_arrival, id: 0, dep: [-1], args:
    {bus_stop: '83139', service_no: 15,
    task_specific:'next'}}].
```

6) Q: I would like to aboard bus no.15 at the bus stop: 83139. How would the bus crowd situation be for the next 30 mins? Answer:

```
[{task: bus_arrival, id: 0, dep: [-1], args:
    {bus_stop: '83139', service_no: 15,
    task_specific:'next_30_mins'}}].
```

7) Q: My current location is inside Jem shopping centre, where are nearby taxi stands? Answer:

```
[{task: recommendation, id: 0, dep: [1], args:
     {location_gps_list: <resource>-1},
    task_specific:['taxi_stand']}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Jem_shopping_centre'
    ]}}].
```

8) Q: My current location is inside Jem shopping centre, how many available taxi arounds my location, like within 2km? Answer:

```
[{task: taxi_availability, id: 0, dep: [1],
    args: {location_gps_list: <resource>-1},
    task_specific:['2km']}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Jem_shopping_centre'
    ]}}].
```

9) Q: What's the traffic situation at Serangoon road right now? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 0, input: history_steps,
    domain: 'traffic_speed'}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Serangoon_road']}}].
```

10) Q: What's the traffic situation at Serangoon road in the coming 2 hours? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 2h, input: history_steps,
    domain: 'traffic_speed'}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Serangoon_road']}}].
```

11) Q: What's the traffic situation at PIE express way in the coming week? Answer:

```
[{task: long_time_series_prediction, id: 0,
    dep: [1], args: {location_gps_list: <
    resource>-1}, time: 1w, input:
    history_steps, domain: 'traffic_speed'},
    {task: map_mapping, id: 1, dep: [-1],
    args: {location_name_list:['PIE_express_
    way']}}].
```

12) Q: As a land and traffic regulator, can you tell whether there are any abnormal traffic speed with Jurong Area? Answer:

```
[{task: time_series_anomaly_detection, id: 0,
     dep: [1], args: {location_gps_list: <
    resource>-1}, input: history_steps,
    domain: 'traffic_speed'}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['Jurong_Area']}}].
```

13) Q: As a land and traffic regulator, can you tell whether there are any abnormal traffic speed in whole Singapore right now? Answer:

```
[{task: time_series_anomaly_detection, id: 0,
     dep: [-1], args: {input: history_steps,
    domain: 'traffic_speed'}}].
```

14) Q: As a land and traffic regulator, can you infer the missing traffic speed values with Jurong Area? Answer:

```
[{task: time_series_imputation, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, input: history_steps, domain: '
    traffic_speed'}, {task: map_mapping, id:
    1, dep: [-1], args: {location_name_list:[
    'Jurong_Area']}}].
```

15) Q: As a land and traffic regulator, can you infer the missing traffic speed values in whole Singapore right now? Answer:

```
[{task: time_series_imputation, id: 0, dep:
    [-1], args: {input: history_steps, domain
    : 'traffic_speed'}}].
```

16) Q: What is the weather nearby NTU right now, is it going to rain? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 0, input: history_steps,
    domain: 'precipitation'}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['NTU']}}].
```

17) Q: What is the weather nearby NTU right now, is it going to rain for the next 2 hours? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 2h, input: history_steps,
    domain: 'precipitation'}, {task:
    map_mapping, id: 1, dep: [-1], args: {
    location_name_list:['NTU']}}].
```

18) Q: What is the air quality nearby NTU right now? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 0, input: history_steps,
    domain: 'air'}, {task: map_mapping, id:
    1, dep: [-1], args: {location_name_list:[
    'NTU']}}].
```

19) Q: What is the weather nearby NTU for the next 2 hours? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, time: 2h, input: history_steps,
    domain: 'air'}, {task: map_mapping, id:
    1, dep: [-1], args: {location_name_list:[
    'NTU']}}].
```

20) Q: As a land and traffic regulator, can you infer the missing parking records for HDB carpark 655? Answer:

```
[{task: time_series_imputation, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1}, input: history_steps, domain: '
    parking'}, {task: map_mapping, id: 1, dep
    : [-1], args: {location_name_list:['HDB␣
    carpark␣655']}}].
```

21) Q: As a land and traffic regulator, can you infer the missing parking records all singapore residential carparks? Answer:

```
[{task: time_series_imputation, id: 0, dep:
    [1], args: {input: history_steps, domain:
     'parking', task_specific: ['residential␣
    carparks']}}].
```

22) Q: As a land and traffic regulator, can you tell whether there are any abnormal parking records for HDB carpark 655? Answer:

```
[{task: time_series_anomaly_detection, id: 0,
     dep: [1], args: {location_gps_list: <
    resource>-1}, input: history_steps,
    domain: 'parking'}, {task: map_mapping,
    id: 1, dep: [-1], args: {
    location_name_list:['HDB␣carpark␣655'
    ]}}].
```

23) Q: As a land and traffic regulator, can you tell whether there are any abnormal parking records form all singapore residential carparks? Answer:

```
[{task: time_series_anomaly_detection, id: 0,
     dep: [1], args: {input: history_steps,
    domain: 'parking', task_specific: ['
    residential␣carparks']}}].
```

24) Q: As a land and traffic regulator, can you infer traffic risk for the next one week? Answer:

```
[{task: event_prediction, id: 0, dep: [1],
    args: {input: history_steps, time: 1w,
    domain: 'traffic␣accident'}}].
```

25) Q: As a land and traffic regulator, can you infer traffic risk within Jurong area for the next one week? Answer:

```
[{task: event_prediction, id: 0, dep: [1],
    args: {location_gps_list: <resource>-1},
    input: history_steps, time: 1w, domain: '
    traffic␣accident'}, {task: map_mapping,
    id: 1, dep: [-1], args: {
    location_name_list:['Jurong␣area']}}].
```

26) Q: As a data network provider, you require abundant user trajectory data to improve signal service. Please infer the missing trajectory for the provided trajectory data? Answer:

```
[{task: trajectory_completion, id: 0, dep:
    [-1], args: {input: trajectory_records,
    domain: 'user␣trajectory'}}].
```

27) Q: As a data network provider, you require abundant user trajectory data to improve signal service. Based on the trajectory provided, predict the next day user trajectory? Answer:

```
[{task: trajectory_prediction, id: 0, dep:
    [-1], args: {input: trajectory_records,
    time: 1d, domain: 'user␣trajectory'}}].
```

28) Q: I would like to have dinner with my girlfriend at Jurong East at 7PM, can you recommend a Japanese restaurant with parking space? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [2], args: {location_gps_list: <resource
    >-2, time: 7PM, input: history_steps,
    domain: 'parking'}}, {task:
    recommendation, id: 1, dep: [2], args: {
    location_gps_list: <resource>-2,
    task_specific: 'Japanese␣restaurant'}}, {
    task: map_mapping, id: 2, dep: [-1], args
    : {location_name_list:['Jurong␣East']}}].
```

29) Q: I would like to have dinner with my girlfriend at Jurong East, we will depart from lake side at 5PM, can you recommend a Japanese restaurant with available parking space when we arrive? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [2], args: {location_gps_list: <resource
    >-2, time: 7PM, input: history_steps,
    domain: 'parking'}}, {task:
    recommendation, id: 1, dep: [2], args: {
    location_gps_list: <resource>-2,
    task_specific: 'Japanese␣restaurant'}}, {
    task: map_mapping, id: 2, dep: [-1], args
    : {location_name_list:['Jurong␣East']}}].
```

30) Q: I would like to play basketball at NTU SRC outdoor courts with my friends. I will drive to NTU and arrive around 7PM, do you have any suggestions? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [3], args: {location_gps_list: <resource
    >-3, time: 7PM, input: history_steps,
    domain: 'parking'}}, {task:
    time_series_prediction, id: 1, dep: [3],
    args: {location_gps_list: <resource>-3,
    time: 7PM, input: history_steps, domain:
    'air'}}, {task: time_series_prediction,
    id: 2, dep: [3], args: {location_gps_list
    : <resource>-3, time: 7PM, input:
    history_steps, domain: 'precipitation'}},
     {task: map_mapping, id: 3, dep: [-1],
    args: {location_name_list:['NTU␣SRC␣
    outdoor␣courts']}}].
```

31) Q: I would like to play basketball at NTU SRC outdoor courts with my friends. I will drive from lake side MRT at around 7PM to NTU, do you have any suggestions including weather and parking space? Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [4], args: {location_gps_list: <resource
    >-4, time: <resource>-3, input:
    history_steps, domain: 'parking'}}, {task
    : time_series_prediction, id: 1, dep:
    [4], args: {location_gps_list: <resource
    >-4, time: <resource>-3, input:
    history_steps, domain: 'air'}}, {task:
    time_series_prediction, id: 2, dep: [4],
    args: {location_gps_list: <resource>-4,
    time: <resource>-3, input: history_steps,
     domain: 'precipitation'}}, {task:
    arrval_time_estimation, id: 3, dep: [4],
    args: {location_gps_list: <resource>-4}},
     {task: map_mapping, id: 4, dep: [-1],
    args: {location_name_list:['NTU␣SRC␣
    outdoor␣courts', 'lake␣side␣MRT']}}].
```

32) Q: I would like find a gym for exercise and having dinner with my friends, we prefer the western food nearby the gym. Can you help to plan out the activities? Answer:

```
[{task: recommendation, id:0, dep: [1], args:
     {location_gps_list: <resource>-1,
    task_specific: 'western␣food'}}, {task:
    recommendation, id: 1, dep: [-1], args: {
    task_specific: 'gym'}}].
```

33) Q: I would like find a gym for exercise and having dinner with my friends, we prefer the western food nearby the gym. Can you help to plan out the activities within Jurong area? Answer:

```
[{task: recommendation, id:0, dep: [1], args:
     {location_gps_list: <resource>-1,
    task_specific: 'western␣food'}}, {task:
    recommendation, id: 1, dep: [2], args: {
    location_gps_list: <resource>-2,
    task_specific: 'gym'}}, {task:
    map_mapping, id: 2, dep: [-1], args: {
    location_name_list:['Jurong␣area']}}].
```

34) Q: I would like to find a gym for exercise and having dinner with my friends, we prefer the western food nearby the gym. Can you help to plan out the activities within Jurong area? I will arrive at 5PM to 5:30PM, please let me know where to park my car as well. Answer:

```
[{task: time_series_prediction, id: 0, dep:
    [1], args: {location_gps_list: <resource
    >-1, time: 2h, input: history_steps,
    domain: 'parking'}}, {task:
    recommendation, id:1, dep: [2], args: {
    location_gps_list: <resource>-2,
    task_specific: 'western␣food'}}, {task:
    recommendation, id: 2, dep: [3], args: {
    location_gps_list: <resource>-3,
    task_specific: 'gym'}}, {task:
    map_mapping, id: 3, dep: [-1], args: {
    location_name_list:['Jurong␣area']}}].
```

B Model Selection Case Studies

We utilized the spatio-temporal analysis results as input to the LLMs during the Model Selection phase to identify the most suitable machine learning models for car park lot prediction. Notably, all LLMs consistently selected the 'SAGDFN' model, specifically trained on parking lot data, outperforming other time series prediction models. The detailed response are listed in Figure 6.

C Task Understanding Prompt

In this section, we present a full task understanding prompt used instruction-tuning of UrbanLLM.

To better understand each spatial-temporal task, here is the explanation and numbering, along with the corresponding examples:

1) Long Time Series Prediction: This task involves forecasting future values in a time series over a long horizon. It is typically used for long-term planning and trend analysis in various domains, such as weather forecasting, economic forecasting, and demand planning.

2) Time Series Prediction: This task focuses on predicting future values in a time series over a shorter horizon compared to long time series prediction. It is commonly used for short-term forecasts like daily stock prices, temperature forecasts, or short-term sales predictions.

3) Event Prediction: This task involves predicting the occurrence of specific events based on historical data. Examples include predicting natural disasters, equipment failures, or social events like concerts or sports games.

4)Trajectory Completion: This task involves completing missing parts of a trajectory based on observed segments. It is useful in applications like tracking moving objects, filling in missing GPS data, or reconstructing incomplete travel routes.

5)Trajectory Prediction: This task involves forecasting the future path of a moving object based on its past trajectory. Applications include predicting the movement of vehicles, pedestrians, or animals.

6)Time Series Anomaly Detection: This task involves identifying unusual patterns or outliers in time series data that deviate from expected behavior. It is used in applications like fraud detection,

Model Selection Phase

Input Prompt: The AI assistant performs task parsing on user input, generating a list of tasks with the following format: task: task, id, task_id, dep: dependency_task_ids, args: {domain: domain, location_name_list: list, location_gps_list:list, time: time, input: sequence, service_no: int, bus_top:string, task_specific:list}}]. Task_specific information is stored using string list and used in specific model/tools. The dep field denotes the id of the previous task which generates a new resource upon which the current task relies. The dep field denotes of '-1' meaning current task does not rely on other tasks and can be excuted right now. Otherwise, excute the previous task with dep field id first. The tag <resource>-task_id represents the generated location, time, sequence, or POIs from the dependency task with the corresponding task_id. Given the question: I want to go to Jurong East for dinner at around 7PM, where can I park? The answer in JSON format is: [{task: time_series_prediction, id: 0, domain: 'carpark prediction', dep: [1], args: {location_gps_list: <resource>-1, Input:history_steps }}, {task: map_mapping, id: 1, dep: [-1], args: {location_name_list:['Jurong East'] }}]. Now, givn the user request question and generated JSON answer, the AI assistant (you) needs to help the user to select a suitable model from a list of candidate models to process the user request. The AI assistant merely outputs the model id of the most appropriate model. The output must be in a strict JSON format: {'id': 'id', 'reason': 'your detail reason for the choice'}. We have a candidate model list of models for you to choose from {{ Candidate Models }}. Please select one model from the candidate model list for each task in JSON answer. Candidate models: [{'model id':1, 'model name':'DCRNN', 'data domain':'road traffic speed', 'description':'DCRNN is a deep learning architecture designed for traffic forecasting tasks, particularly in urban areas. It combines techniques from convolutional and recurrent neural networks to effectively capture spatial and temporal dependencies in traffic data. The model takes advantage of the graph structure of traffic data, where nodes represent different locations (such as intersections or sensors) and edges represent connections between these locations.'}, {'model id':2, 'model name':'SAGDFN', 'data domain':'road traffic speed, carpark availability lots', 'description':'SAGDFN: A Scalable Adaptive Graph Diffusion Forecasting Network for Multivariate Time Series Forecasting aims to provide accurate multivariaate time series predictions for both normal and larger datasets and datasets span various times series domains such as traffic speed to carpark availability lots.'}, {'model id':3, 'model name':'GPS retrieval', 'data domain':'location name', 'description':'This tool receives location names list and outputs the location gps list accordingly.'}. Please select one most suitable model for each sub-tasks like time series prediction and map mapping. Your response should strictly in the JSON format containing 'id' and 'reason' only.

LLMs Response

GPT-4o: For the first task, which is time_series_prediction, the most suitable model would be 'SAGDFN' because it is designed for multivariate time series forecasting and can handle datasets related to carpark availability lots, which is relevant to the user's request. \n\nFor the second task, which is map_mapping, the most suitable model would be 'GPS retrieval' because it is designed to receive location names and output the corresponding GPS locations, which is exactly what the task requires.\n\nSo, the JSON response would be:\n\n[\n {'id': '2', 'reason': 'SAGDFN is designed for multivariate time series forecasting and can handle datasets related to carpark availability lots, which is relevant to the user request.'},\n {'id': '3', 'reason': 'GPS retrieval is designed to receive location names and output the corresponding GPS locations, which is exactly what the task requires.'}\n]

GPT-3.5: ["id": "2",\n "reason": "SAGDFN model is the most suitable for the task of time series prediction as it is designed for multivariate time series forecasting, which includes traffic speed and carpark availability lots data domains. This model can effectively capture the dependencies in the data and provide accurate predictions for the user\'s request."]

Llama2: [Great, let's get started! You have provided me with a user request question, a generated JSON answer, and a list of candidate models. I will now help you select the most appropriate model for each task. Firstly, let's analyze the user request question: "I want to go to Jurong East for dinner at around 7PM. Where can I park?" Based on the question, the most appropriate task for the AI assistant to perform is "map mapping" with the dep field set to [-1], as it requires the AI assistant to find the nearest parking location to Jurong East based on the user's request. Now, let's select a suitable model from the list of candidate models. Based on the description of the models, I will choose model ID 2, "SAGDFN", as it is designed for traffic forecasting tasks and can effectively capture spatial and temporal dependencies in traffic data, which is relevant to the user's request for finding a parking location. Therefore, the output in JSON format will be: { "id": "2", "reason": "SAGDFN is a suitable model for finding the nearest parking location to Jurong East based on the user's request, as it is designed for traffic forecasting tasks and can effectively capture spatial and temporal dependencies in traffic data." } Please let me know if you have any further questions or requests!"]

UrbanLLM: [Based on the provided user request and JSON answer, I have selected the GPS retrieval model with id 3 for the task of providing the location gps list for the carpark at Jurong East. The reason for this choice is that the user request requires the location gps list for the carpark at Jurong East, which can be obtained using the GPS retrieval model. The other tasks in the JSON answer require the prediction of traffic speed and the mapping of carpark availability, which can be handled by the SAGDFN model with id 2. Please select the next task in the JSON answer for which you require assistance in selecting a model from the candidate model list.]

Figure 6: A sample of detailed LLMs response in the Model Selection phase.

fault detection in machinery, and monitoring traffic conditions.

7) Time Series Imputation: This task involves filling in missing values in time series data to ensure completeness and consistency. It is crucial for maintaining data quality in various applications like traffic records and climate data.

8)Arrival Time Estimation: This task involves predicting the arrival time of a vehicle or person at a specific location based on current and historical data. It is commonly used in transportation systems for buses, trains, and delivery services.

9) Taxi Availability Prediction: This task involves predicting the availability of taxis in specific areas at given times. It helps optimize taxi dispatching and improve service for passengers by anticipating demand and ensuring timely availability.

10) Map Mapping: This task involves mapping addresses to GPS locations and mapping GPS locationsback to addresses .

11) Bus Arrival: This task involves predicting the arrival times of buses at specific stops based on real-time data and historical patterns. It enhances the efficiency of public transportation systems by providing accurate and timely information to commuters.

12) Spatial Relationship Inference: This task involves deducing spatial relationships between different entities or locations. It is used in urban planning to understand spatial dependencies and interactions, such as proximity analysis, clustering, and spatial correlations.

13) Recommendation: This task involves suggesting items or actions to users based on their preferences and historical behavior. Applications include recommending points of interest, routes, or services in urban planning.

D Spatio-Temporal Model Description

We demonstrate candidate model description for time series predictions as below:

['model id':1, 'model name':'DCRNN', 'data domain':'road traffic speed', 'description':'DCRNN is a deep learning architecture designed for traffic forecasting tasks, particularly in urban areas. It combines techniques from convolutional and recurrent neural networks to effectively capture spatial and temporal dependencies in traffic data. The model takes advantage of the graph structure of traffic data, where nodes represent different locations (such as intersections or sensors) and edges represent connections between these locations.', 'model id':2, 'model name':'SAGDFN', 'data domain':'road traffic speed, carpark availability lots', 'description':'SAGDFN: A Scalable Adaptive Graph Diffusion Forecasting Network for Multivariate Time Series Forecasting aims to provide accurate multivariaate time series predictions for both normal and larger datasets and datasets span various times series domains such as traffic speed to carpark availability lots.', 'model id':3, 'model name':'AGCRN', 'data domain':'road traffic speed', 'description':'Adaptive graph convolutional recurrent network for traffic forecasting takes advantage of the graph structure of traffic data to assist in traffic speed predictions.']

E Evaluation Metrics

Precision. Macro precision is the average precision for each sample.

For each task i:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

Where: - TP_i is the number of true positives for task i - FP_i is the number of false positives for task i

Macro precision:

$$\text{Macro Precision} = \frac{1}{N}\sum_{j=1}^{N}(\frac{1}{C}\sum_{i=1}^{C}\text{Precision}_i)_j$$

Recall. Macro recall is the average recall for each sample.

For each task i:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}$$

Where: - TP_i is the number of true positives for task i - FN_i is the number of false negatives for task i

Macro recall:

$$\text{Macro Recall} = \frac{1}{N}\sum_{j=1}^{N}(\frac{1}{C}\sum_{i=1}^{C}\text{Recall}_i)_j$$

F1 Score. Macro F1 score is the average F1 score for each sample.

For each class i:

$$\text{F1}_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

Macro F1 score:

$$\text{Macro F1 Score} = \frac{1}{N}\sum_{j=1}^{N}(\frac{1}{C}\sum_{i=1}^{C}\text{F1}_i)_j$$

Where N is the total number of samples and N is the total number tasks types.