

**Real-Time Data Informed Traffic Analytics Framework Powered by  
Large Language Model (LLM)**

Bingzhang Wang

A thesis

submitted in partial fulfillment of the  
requirements for the degree of

Master of Science

University of Washington

2024

Committee:

Yinhai Wang

Edward McCormack

Program Authorized to Offer Degree:

Civil and Environmental Engineering

©Copyright 2024

Bingzhang Wang

University of Washington

**Abstract**

Real-Time Data Informed Traffic Analytics Framework Powered by  
Large Language Model (LLM)

Bingzhang Wang

Chair of the Supervisory Committee:

Yinhai Wang

Department of Civil and Environmental Engineering

The digitization of traffic sensing infrastructure has significantly accumulated an extensive real-time traffic data warehouse, which presents unprecedented challenges for traffic analytics. The complexities associated with querying large-scale multi-table databases require specialized programming expertise and labor-intensive development. Additionally, traditional analysis methods have focused mainly on numerical data, often neglecting the semantic aspects that could enhance interpretability and understanding. Furthermore, real-time traffic data access is typically limited due to privacy concerns. To

bridge this gap, the integration of Large Language Models (LLMs) into the domain of traffic management presents a transformative approach to addressing the complexities and challenges inherent in modern transportation systems. This thesis proposes an intelligent traffic analytics framework based on LLM for efficient customized transportation surveillance and management empowered by a large real-time traffic database. The innovative framework leverages contextual and generative intelligence of language models to generate accurate SQL queries and natural-language data interpretations by employing transportation-specialized prompts, Chain-of-Thought prompting, few-shot learning, multi-agent collaboration strategy, and chat memory. Experimental study demonstrates that the approach outperforms state-of-the-art baselines such as GPT-4 and PaLM 2 on a challenging traffic-analysis benchmark TransQuery. This study would aid researchers and practitioners in real-time transportation surveillance and management in a privacy-preserving, equitable, and customizable manner.

# TABLE OF CONTENTS

List of Figures .....	1
List of Tables .....	2
Acknowledgements.....	3
Chapter 1: Introduction and Motivation .....	4
1.1 Background and Challenges.....	4
1.2 Research Objective.....	8
1.3 Scope of Study .....	9
Chapter 2: Literature Review.....	12
2.1 Traffic Management with LLMs.....	12
2.2 Time Series Data Analysis with LLMs .....	14
2.3 LLMs in Database SQL Query.....	15
2.4 Prompt Engineering.....	16
2.5 Few-Shot and Zero-Shot Learning .....	16
Chapter 3: Methodology .....	18
3.1 Problem Statement .....	18
3.2 System Design.....	22
3.2.1 Task Management.....	22

3.2.2 Query Generation .....	24
3.2.3 Quality Check.....	24
3.2.4 Query Execution.....	25
3.2.5 Data Interpretation .....	25
3.3 Input Prompt Generation.....	28
3.3.1 Chain-of-Thought Iteration.....	28
3.3.2 Role Prompting.....	32
3.3.3 Context Prompting.....	33
3.3.4 Few-Shot Prompting.....	36
3.4 Multi-Agent Strategy.....	39
3.5 Few-Shot Learning .....	42
3.6 Chat Memory.....	45
3.7 ChatBot Interaction Web Interface .....	47
Chapter 4: Experiment .....	50
4.1 Experimental Setup .....	50
4.2 Performance Comparison .....	52
4.3 Results Analysis .....	57
4.4 Ablation Study.....	66

Chapter 5: Conclusion and Future Work .....	68
Bibliography .....	70
Appendix.....	77

## LIST OF FIGURES

Figure 1.1 Increasing Vehicle Miles of Travel in the Greater Seattle Area .....	5
Figure 1.2 Vehicle detection using inductive loops .....	6
Figure 3.1 Inductive loop detector locations shown as blue points .....	19
Figure 3.2 System architecture design demonstrating traffic analysis pipeline.....	27
Figure 3.3 Chain-of-thought prompting with intermediate reasoning highlighted (Wei, et al., 2022) .....	30
Figure 3.4 Chain of Thought prompting .....	32
Figure 3.5 Input prompt template .....	38
Figure 3.6 TP-GPT multi-agent collaboration framework for traffic data analysis.....	40
Figure 3.7 Few-shot learning exemplar inventory .....	44
Figure 3.8 Reading and writing chat memory to integrate past knowledge to current answer reasoning .....	46
Figure 3.9 Online web application interface of TP-GPT ChatBot.....	49
Figure 4.1 Statistical experimental results of TP-GPT and baselines on TransQuery benchmark.....	54
Figure 4.2 Ablation study of TP-GPT on TransQuery by removing prompts, few-shot learning, or multi-agent strategy .....	66

## LIST OF TABLES

Table 3.1 Network-wide traffic database introduction.....	20
Table 4.1 Sample questions in TransQuery.....	51
Table 4.2 Performance comparison of TP-GPT with baseline models on TransQuery traffic-analysis benchmark.....	53
Table 4.3 Exemplary answers generated by TP-GPT with ratings to the questions in TransQuery benchmark.....	62
Table 5.1 Complete experimental results of TP-GPT on TransQuery benchmark .....	77

## ACKNOWLEDGEMENTS

I would like to express my great gratitude to my advisor, Professor Yin Hai Wang, for constant support of my study and research. His patient guidance, seamless motivation, and profound expertise have inspired me to become a better person and researcher. I am also thankful to the member of my thesis committee, Professor Edward McCormack, for valuable insights and constructive feedback that have enriched my work.

Special thanks to my colleagues in the Smart Transportation Applications and Research (STAR) Lab, especially Dr. Muhammad Monjurul Karim and Chenxi Liu, whose insightful mentorship have made my work both productive and enjoyable. I am also grateful for the joyful times spent with my lab mate and friend, Nutvara Jantarathaneewat, whose companionship has made our collaborative efforts a delightful experience.

I wish to extend my heartfelt thanks to my family: my parents, my partner Candice Liu, and my cat Newboy, for their love, patience, and encouragement, which have provided me with the spiritual strength and solid foundation needed to complete this journey.

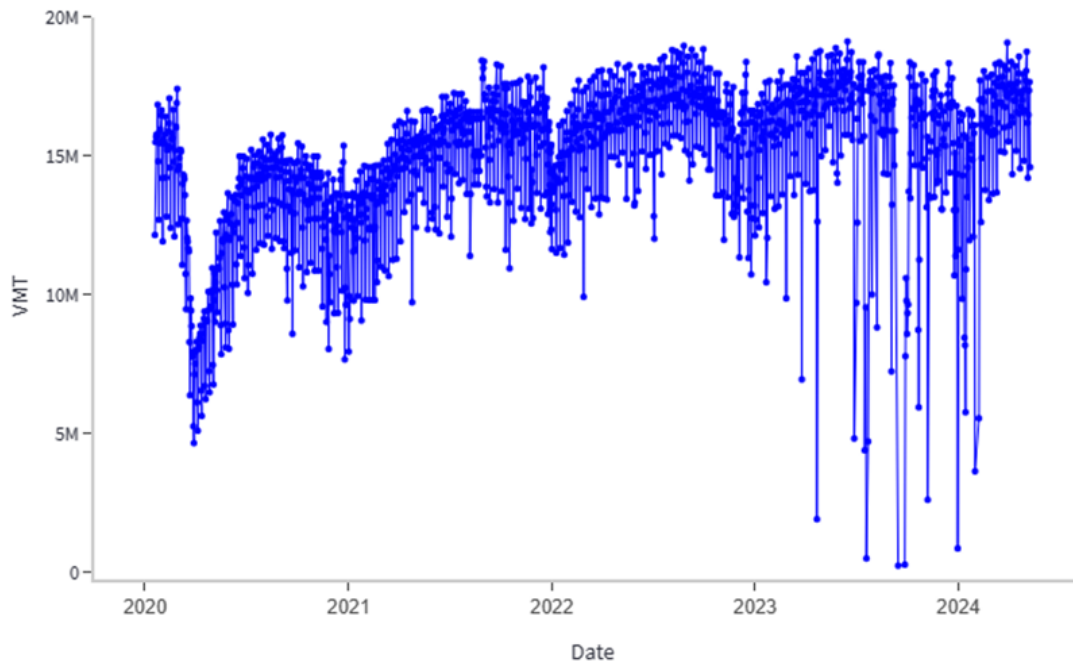
# CHAPTER 1: INTRODUCTION AND MOTIVATION

## 1.1 BACKGROUND AND CHALLENGES

A substantial magnitude of traffic intensity is being observed in modern mobility. As reported by the Washington State Department of Transportation, the annual Vehicle Miles of Travel (VMT) on the state highway of King County reached a total of 8,534 million in the year 2022. A consistent increase trend of VMT is also observed in the Greater Seattle Area after COVID-19 pandemic, shown in Figure 1.1. Meanwhile, with the rapid digitization of sensing infrastructure, such as inductive loop detectors shown in Figure 1.2, an immense volume of traffic data is being collected at an ever-increasing rate in the real-time and historical data warehouse. This presents both opportunities and challenges. On one hand, these unprecedented data resources hold the promise of propelling advancements in traffic analysis, making it more accurate and reliable; On the other hand, the unpredictable accumulation of data poses significant challenges for the development of sophisticated traffic analysis techniques.

Firstly, real-time traffic sensing data is typically stored in large-scale, multi-table databases. These databases are incredibly large and have complex relationships between different data points. Querying these databases can be labor-intensive and time-consuming, leading to significant latency. To properly manipulate databases even requires not only a deep preliminary understanding of context but also specialized expertise in database

programming.



**Figure 1.1 Increasing Vehicle Miles of Travel in the Greater Seattle Area**

(Image source: <http://tps.uwstarlab.org/>)

Secondly, traditional methods of traffic analysis have primarily focused on the numerical aspects of the data, using statistical methods and machine learning techniques. This overlooks the semantic and interpretability attributes of the data. As a result, the exploration of traffic datasets has been limited to numerical imputations and prescriptive visualization, neglecting their inherent natural-language significance.

Thirdly, access to real-time traffic data is typically restricted to authorized entities such as government agencies and academic institutions, due to privacy concerns, making it inaccessible for direct data handling by the general public. There is a pressing need for an intermediate framework that processes and interprets data for practitioners in a privacy-preserving manner, which has the potential to not only facilitate trip planning and policy-

making but also improve traffic analysis by making it more accessible, efficient, and equitable.



**Figure 1.2 Vehicle detection using inductive loops**

(Image source: <https://wsdot.wa.gov/>)

In recent years Large Language Models (LLMs) have emerged as a groundbreaking development in the field of artificial intelligence (AI), demonstrating unparalleled capabilities in understanding, and interpreting real-world scenarios in human-like ways (Chang, et al., 2023). These models, powered by advanced neural network architectures, have found applications across a wide range of domains, from natural language processing and machine translation to content generation and beyond. They offer unprecedented capabilities in data interpretation and decision-making.

LLMs have proven their versatility in different fields, including education, healthcare, software engineering (Fraivan & Khasawneh, 2023), among many others. While still a

new area of exploration, LLMs show promise for analyzing traffic data. Their general ability to adapt to different tasks makes them attractive for various intelligent transportation applications, including traffic signal control (Lai, Xu, Zhang, Liu, & Xiong, 2023), and accident risk assessment (Wang, et al., 2023). However, effectively using LLMs in these specialized areas requires them to have a deep understanding of the specific domain (Ge, et al., 2024). This necessitates the integration of domain-specific databases with LLMs to foster data-driven reasoning capabilities. Recent research has explored leveraging LLMs for database operations. Frameworks like LangChain facilitate efficient interaction with LLMs (Topsakal & Akinci, 2023), while models like DB-GPT are fine-tuned with domain-specific database knowledge (Zhou, Sun, & Li, DB-GPT: Large Language Model Meets Database, 2024), enabling the translation of textual semantics into database queries (text-to-SQL). These advancements offer exciting possibilities for advanced analysis of large-scale traffic data. However, existing models primarily target at general-purpose database tasks, neglecting the unique knowledge domain of traffic data coherent to transportation analytics and management. This omission of traffic-specific background knowledge during training may lead to diminished query result accuracy and limit their applicability in traffic research.

LLMs have marked their prominence by enabling a multitude of specific-role tasks across diverse fields. Their general capabilities and adaptability are recognized by many industries as research which is attracting more and more attention. Several recent studies have shown that LLMs have great potential in education (Lo, 2023; Rahman & Watanobe,

2023), healthcare (Fraiwan & Khasawneh, 2023), software engineering (Jin, et al., 2023; Ross, Martinez, Houde, Muller, & Weisz, 2023), among many others. In intelligent transportation areas, under more segmented scenarios like autonomous driving (Yang, Jia, Li, & Yan, 2023), signal control (Lai, Xu, Zhang, Liu, & Xiong, 2023), accident and risks assessment (Zhang, et al., 2024; Wang, et al., 2023; de Zarzà, de Curtò, Roig, & Calafate, 2023). Researchers in this field are promptly starting to require language models to equip with expert-level knowledge due to the highly specialized nature of these domain tasks. In order for LLMs to have specific reasoning and generation abilities, the corresponding domain knowledge reserve is proved to be very important (Peng, et al., 2023; Ge, et al., 2024).

## **1.2 RESEARCH OBJECTIVE**

Handling real-time traffic data involves navigating large, complex databases, leading to high latency and challenges in data integration due to intricate entity relationships. The advent of LLMs presents an inspiring potential to revolutionize such traffic data management by serving as the central intelligence of next-generation database systems. A few studies have concentrated on harnessing LLMs for database operations. Among these, some have developed integration frameworks, such as LangChain, for efficient interactions with LLMs, while others have fine-tuned models with domain-specific database knowledge, such as DB-GPT, aiming to translate text semantics into database queries (i.e., text-to-SQL). These techniques provide an exciting possibility for advanced large-scale

traffic data analysis, however, these models are aimed for general-purpose database tasks, no research so far is focused on traffic database. The lack of embedding traffic background knowledge in training these models may lead to low accuracy of query results and low applicability to traffic-domain research.

Despite the proliferation of LLM applications across various domains, their potential in traffic data analysis, especially within databases, remains an open research topic. Furthermore, there is a pressing need for a user-friendly intelligent platform that can effectively communicate and analyze real-time data. The study aims to delve into this gap by proposing an intelligent transportation analytics system that applies pre-trained large language models to complex traffic data analysis. It leverages their capability to generate accurate SQL queries and natural language interpretations based on contextual awareness, demonstrating the extensive pre-trained knowledge of LLMs and their proficiency in adapting to the transportation domain.

### **1.3 SCOPE OF STUDY**

This study focuses on the theory and application of the interaction between LLMs and network-level traffic databases, by extracting many potential shortcomings and gaps in cutting-edge research that need to be further explored, and proposing innovative solutions.

The contributions of this thesis are listed below:

- 1) An intelligent online chatbot, Traffic Performance GPT (TP-GPT), is proposed for efficient personalized transportation analysis and management leveraging the

support of big real-time traffic data. To the best of our knowledge, TP-GPT is the first real-time traffic analysis chatbot empowered by LLMs to be proposed.

- 2) Leveraging contextual and generative intelligence of the Generative Pre-trained Transformer (GPT), an innovative framework is constructed to serve as a connection between public users and authorized data resources in a privacy-preserving, equitable, customizable way.
- 3) The developed chatbot is able to generate reliable, responsive and accurate traffic analysis and management responses to input questions, by integrating designed prompts, few-shot learning module, multi-agent collaboration strategy and conversation memory module. The proposed method outperforms existing general-purpose LLMs regarding traffic-domain analysis performance on the proposed benchmark TransQuery.

The structure of the thesis is organized as follows: Chapter 1 introduces general background of the study, defines research gap existed in the current work, and illustrates the research objectives; Chapter 2 reviews and summarizes the state-of-the-art literatures related to traffic data analytics, traffic management with LLMs, and database query using LLMs; Chapter 3 describes the methodology by explaining details of the developed transportation analytics framework based on LLM, including problem statement, system design, input prompt generation, multi-agent strategy, few-shot prompting, chat memory, and interface system development; Chapter 4 presents the experimental results to demonstrate the effectiveness of the work by comparing its performance to baseline models

on proposed traffic analytics benchmark TransQuery, which shows the proposed method significantly outperforms other baselines; Finally, Chapter 5 concludes the conducted study in the thesis and proposes envisions on the future work.

## CHAPTER 2: LITERATURE REVIEW

### 2.1 TRAFFIC MANAGEMENT WITH LLMs

Traffic management for traffic networks especially in urban area encounters complex contexts and background knowledge. Previous research usually relied on traditional hardware infrastructure, such as surveillance cameras and ground-based sensors, as well as rule-based systems to monitor and adjust traffic flow. Researchers have employed both micro and macro methods (Yang & Koutsopoulos, 1996; Kurzhanskiy & Varaiya, 2010; Wang, Cai, Chen, Yang, & Chen, 2023; Mahmassani, 2001) based on theories and data from traditional methods to simulate traffic flow, promoting the advancement of theoretical research as well as management strategies.

Recent research addresses challenges of traffic more from data-driven perspectives with the emerging techniques of mobility and traffic data collection and analysis. There has been a shift towards leveraging machine learning and deep learning methods (Chen, et al., 2024; Ma, Yu, Wang, & Wang, 2015) to provide new solutions to traffic control and management. One example is traffic scenario at intersections. Authors in (Lai, Xu, Zhang, Liu, & Xiong, 2023) introduced a framework that instructs fine-tuned LLMs to perform traffic signal control tasks in real-world datasets. The problem is formulated as a Markov Game in which the LLM-based agent learns a policy to enhance traffic efficiency at intersections. The framework introduced in research (Wang, et al., 2024) integrated LLM

with perception and decision tool sets while each tool is paired with its own prompt. In addition to intersections, other scenarios have also been studied. Authors in (Villarreal, Poudel, & Li, 2023) discussed the use of ChatGPT in Ring and Bottleneck scenarios to improve the performance of robot vehicle (RV) for traffic flow control and optimization.

Other research expanded the scope of specific scenarios, such as (Da, et al., 2023; Zhang, et al., 2024) combines LLMs to assist human users in traffic network analysis and further decision-making. Users are able to interact with network-level data and complete various tasks with signal or network volume data. A traffic prediction model proposed by researchers in (Ren, et al., 2024) uses LoRA (Yu, et al., 2023) to fine-tune their model in order to better tackle flow prediction tasks under full-sample and few-shot historical data scenarios.

One other significant challenge in large-scale traffic network is model's capability to understand phenomena and content in traffic and transportation. Authors in (Wang, Wei, Hu, & Han, 2024) manually collected multi-modal traffic data sets, focusing on GPT's understanding of traffic scenes, such as text and traffic signs alignment for model training. Authors in (Tang, et al., 2024) focused on users' itinerary planning demands. They proposed ItiNera to meet user demands for citywalk responsively and spatially coherently through multiple modules such as POI Collection and retrieval. Among a variety of integration of Large Language Models into traffic management, this paper showcases a promising avenue for enhancing real-time traffic analysis and management in order to benefit practitioner, traveler and researcher.

## 2.2 TIME SERIES DATA ANALYSIS WITH LLMs

Extensive research has explored into time series data analysis and forecasting with LLMs. By encoding the data in a specific way (Sun, Li, Li, & Hong, 2023), LLMs can show good predictive performance. Hence, they are currently widely used to conduct research on time series and spatial-temporal data (Liu, et al., 2024; Chen, Wang, & Xu, 2023; Yuan, Ding, Feng, Jin, & Li, 2024), and their potential to analyze data of such modalities is being further explored, including univariate time series forecasting, multivariate time-series data.

Study (Jin, et al., 2023) proposed a framework to embed temporal information within LLM architectures, in which authors use a pre-trained LLM as a frozen component and reprogram multivariate general time-series data as input. Their results showed that LLMs have great potential to achieve leading performance even without fine-tuning. In (Rasul, et al., 2023), a foundation model for univariate probabilistic time series forecasting was established. Their model was claimed to having state-of-the-art performance across various domain with finetuning. Research in (Zhou, Niu, wang, Sun, & Jin, 2023) proposed a frozen pre-trained transformer (FPT) trained from NLP or CV modalities, to tackle downstream tasks and time series analysis. Findings in (Cao, et al., 2023) utilized decomposition of time series and prompt pool, along with multimodal input to leverage contextual information of time series forecast.

### 2.3 LLMs IN DATABASE SQL QUERY

LLMs exhibit remarkable capabilities in generating reliable and accurate SQL queries, advancing database interaction efficiency. Their inherent contextual intelligence enables them to understand data origins and relationships, thus enhancing the performance of data queries and analyses. By leveraging LLMs, databases can be queried more effectively, ensuring that target data is extracted with a high degree of precision. Query efficiency is also able to be enhanced with deep comprehension of LLMs in algorithm knowledge such as data structure, computation complexity, programming languages, and so on. This prowess underscores the transformative potential of LLMs in streamlining database management and analysis processes, setting a new benchmark for data-driven decision-making.

A recent study introduces BIRD (Li, et al., 2023), a benchmark for large-scale database text-to-SQL tasks highlighting challenges such as content quality, external knowledge integration, and SQL efficiency. Many scholars use SQL agent to enhance database-related tasks, such as (Hong, et al., 2024) employs a Data Expert LLM (DELLM) to provide necessary knowledge for text-to-SQL models to generate accurate queries by incorporating data expert knowledge. Authors in (Cai, et al., 2022) enhance text-to-SQL parsing in multi-turn conversations by leveraging contextual information from the dialogue history. Research (Li & Xie, 2024; Zhou, Li, & Liu, Llm as dba, 2023) share a common theme of utilizing LLMs to enhance database-related tasks, with the former proposing an LLM-

based database administrator that learns from text to diagnose and optimize databases, and the latter using LLMs to automatically generate test cases for selecting the most accurate SQL query from a set of candidates.

## **2.4 PROMPT ENGINEERING**

Prompt engineering emerges as a pivotal technique for enhancing the performance of LLMs in specific tasks. LLMs are considered having the capabilities to utilize and even create their own tools targeting specific tasks (Cai, Wang, Ma, Chen, & Zhou, 2023). Authors in (Zhang, Zhou, Wu, Xie, & He, 2024) established a model based on graph transformer to perform data imputation to address missing data in traffic data management. Users can pose queries without background knowledge while the model is able to capture spatial-temporal correlation within the data. This study has shown great performance of LLMs and application potential of prompt engineering in traffic management.

## **2.5 FEW-SHOT AND ZERO-SHOT LEARNING**

Few-shot and zero-shot learning in the context of traffic management enables LLMs to adapt to new traffic patterns or incidents with limited data. For instance, after a new road is constructed, a model trained with few-shot learning can quickly adjust to the flow of traffic without requiring extensive historical data from that specific location. In study (Li, et al., 2024), the proposed UrbanGPT model captures and understands complex spatial-temporal dependencies when training samples were limited, showing good ability to

complete zero-shot learning tasks. This indicates a good potential of fine-tuned language model in analysis and prediction performance across different cities.

## CHAPTER 3: METHODOLOGY

### 3.1 PROBLEM STATEMENT

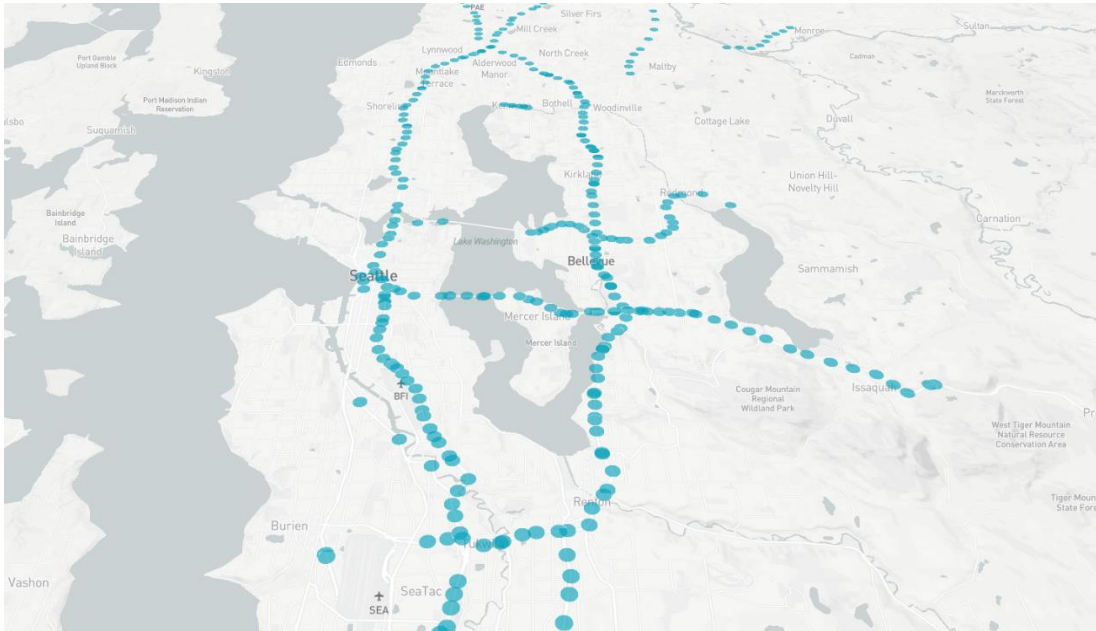
Unpredictable, fast-changing traffic patterns underlying the enormous numerical data pose an ever-challenging task for efficient and effective traffic analysis and management. To tackle this problem, we incorporate a large-scale network-wide mobility database hosted by STAR Lab (Cui, et al., STAR Lab Traffic Performance Score Platform, 2020), which integrates the real-time data resources of traffic counts (i.e., speed, volume, occupancy) collected from more than 8,000 inductive loop detectors, as well as the route segment-wide Traffic Performance Score (TPS) calculated using Equation (3.1 based on loop data (Cui, et al., Traffic Performance Score for Measuring the Impact of COVID-19 on Urban Mobility, 2020).

$$TPS_t = \frac{\sum_{i=1}^n V_t^i \cdot Q_t^i \cdot L^i}{\sum_{i=1}^n V_f \cdot Q_t^i \cdot L^i} \times 100\% \quad (3.1)$$

where  $V_t^i$  and  $Q_t^i$  represent traffic speed and volume of road segment  $i$  at time  $t$ .  $L^i$  is road segment length covered by the  $i$ -th detector.  $V_f$  is free flow speed. Therefore, TPS is a value ranging from 0% to 100% where 0% is the worst traffic condition and 100% is the best.

These loop detectors are deployed on freeways, including I-5, I-90, I-99, I-167, I-405, and SR-520, in the Greater Seattle Area, WA, as shown in Figure 3.1. Data have been

collected online in one-minute intervals from 2020 to the present, and it has accumulated around 1.89 Terabytes (TB) in the data warehouse. There are a total of 6 tables in the database, with the details shown in Table 3.1.



**Figure 3.1 Inductive loop detector locations shown as blue points**

This level of database has great potential for insightful network-wide analysis of real-time traffic performance. However, vast volume of data poses great challenges for practitioners' and researchers' inspection and management of data. Firstly, exploring the data warehouse necessitates participants' deep understanding of database setup details (i.e., tables, columns, data types, relations, and so on), as well as expertise in database programming especially processing large-scale databases. Even though, manually programming various queries and then executing commands for real-time information in such a huge database is particularly time-consuming and labor-intensive, making it almost a mission impossible. Second, interpreting from numeric data results to human-language

traffic advisory or impactful analysis results needs in-depth professional transportation-domain knowledge and a wide range of historical knowledge base support such as urban planning, and social events. Third, the lack of direct access to the database for security and privacy concerns limits the potential participants' exploration, hindering flexible investigation.

**Table 3.1 Network-wide traffic database introduction**

<b>Table Name</b>	<b>Columns</b>	<b>Description</b>
<i>dbo.cabinets</i>	17	Loop detector details of unit name, coordinate, route, milepost, and direction.
<i>dbo.cabinfo</i>	6	District location of loop detectors sorted by cabinet station ID.
<i>dbo.MinuteDataNW</i>	6	One-minute traffic speed, volume, occupancy data in Washington Northwest sorted by loop detector ID and timestamp.
<i>dbo.Segments</i>	6	Road segment definition with corresponding location information.
<i>dbo.SegmentTrafficIndex</i>	8	Segment-based traffic performance data on general-purpose lanes and carpool lanes, including speed, volume, and TPS.
<i>dbo.TrafficIndex</i>	9	Statistical traffic performance data for each

---

defined road segment.

---

In this context, we propose an intelligent traffic performance chatbot TP-GPT for real-time transportation surveillance and management in a privacy-preserving way. While, technical problems need to be tackled for comprehensive development:

- 1) ChatGPT does not have preliminary contextual knowledge of the database setup, structures and content. Even though ChatGPT has strong abilities in programming and inference, direct deployment of the language model to database analysis without prerequisites may result in hallucination that blind actions irrelevant to the raised question would possibly be taken. Thus, the compilation of appropriate input prompts is essential for employing ChatGPT.
- 2) ChatGPT has a reliable performance of executing sequential commands activated by human's input. However, the lack of monitoring the response relevance and query correctness regarding to the user's question may generate unexpected, unreliable answers. It is necessary to incorporate an intelligent autonomous streamline with self-management capability for quality control of answer generation. Specifically, iterative communications between ChatGPT and the database in the revision circulation is needed to keep modifying responses until certain standard is reached. This process necessitates the participation of multiple intelligent virtual agents of different roles, such as database query engineer, quality

- manager, consultant and more.
- 3) The demonstration of a set of typically asked questions with their corresponding answers are critical for ChatGPT to specify the work scope, furthermore, to improve the effectiveness and efficiency of target response.
  - 4) In order for users to be able to consecutively interact with the chatbot, a chatting memory function is needed so that the chatbot has the historical memory to support following generative process.

## **3.2 SYSTEM DESIGN**

TP-GPT aims to answer users' questions by integrating GPT with real-time traffic SQL database. However, directly applying generative model to the database has potential to cause several concerns, such as a tendency for reasoning hallucination, decreased answer accuracy, and unstable performance reliability. To overcome these obstacles, the proposed framework incorporates multiple strategic functional modules, forming a pipeline to process and interpret data, as shown in Figure 3.2. Specifically, once users' questions are input into the ChatBot, five steps are executed. Each step acquires contextual and generative intelligence by iteratively interacting with GPT using Chain-of-Thought (CoT) prompting, to generate the final answer.

### ***3.2.1 Task Management***

Since the fixed sequence of implementation steps is not adaptive to resolving dynamic and

diverse question inputs, it is essential that a customized execution plan is produced based on the input for properly arranging the task accomplishment schedule. Meanwhile, each execution step of the management plan is determined and adjusted flexibly by GPT's innate decision-making capability. For example, if the question is relevant to searching for the specific traffic data, in the execution plan, database schema will be queried to get landscape of table structures, SQL query will be generated and executed in the database to extract objective data, and a natural-language analysis report will be composed by interpreting queried data; Otherwise, the regular chat function will be activated to interact with users based on the general knowledge base. It is worth noting that to enhance the accuracy of the system in retrieving data relevant to user inputs, the temperature setting of the ChatGPT model is configured to zero. This setting ensures that the language model produces only concrete, deterministic results based on the factual data it has accessed. By doing so, the model eliminates the variability in responses that can be introduced by higher temperature settings. This adjustment is critical in applications where precision and reliability are prioritized.

This intelligent workflow control process is dependent on the semantic understanding and strategic decision-making ability to deconstruct complicated tasks into simpler sub-tasks and then assign them to diverse modules as described in the following sections. By integrating task management function, the system is enabled to resolve sophisticated problems with higher robustness and can effectively decrease the possibility of causing errors when an excessive number of commands are input into the model without

arrangement.

### ***3.2.2 Query Generation***

To accurately extract data from the database in response to user queries, the system deconstructs each input question into key informational components and query sub-tasks. As an example, for the question “*How is traffic condition on I-5 during today’s morning peak hours*”, the system will dissemble this question into three query sub-tasks: query Traffic Performance Score to assess “*traffic condition*”, which represents a quantitative measure of traffic flow and congestion levels; identify the corresponding route number to “*I-5*”, ensuring the query targets the correct geographical route; retrieve the standard definition used by the traffic system for the what time range constitutes “*morning peak hours*”. Subsequently, these sub-tasks are allocated to execution modules and respective commands are implemented to form a complete query sentence. Constraints on generating queries, such as programming syntax, quantity of queried data, and database table structure manipulation limits, are applied to the process.

### ***3.2.3 Quality Check***

An initial syntax check is performed on the generated query, which covers comprehensive aspects to ensure the query’s correctness. Inspection on whether the query is beyond the designated constraints is conducted, in order to avoid overloaded data instances are queried which may cause database runtime error. Besides, compatibility of the generated query

with database server is tested by checking if improper syntax is used in the query, which may lead to execution errors in the database. This type of error is frequently observed for timestamp operations. Furthermore, any error messages returned from database server to the system will be inspected to locate logic mistakes retained in the generated query, including inherent logic errors even if syntax is correct. If the query successfully passes quality check, it will be forwarded to the execution module; Otherwise, the query will be returned to query generation module along with error messages to regenerate a revised query, with necessary improvements integrated. This cycle of checking and adjusting queries ensures their accurate and efficient execution.

#### ***3.2.4 Query Execution***

The checked query is executed in the SQL database server. During this process, GPT is able to utilize database toolkit to perform operations based on its contextual understanding. Additionally, if necessary, an inspection of the execution log can be conducted. This step is crucial as it allows for the analysis of important information contained within the logs, which can help in identifying and debugging any errors that occur during the query process. Following the query execution, queried data is transmitted back from database server to the system for further exploration.

#### ***3.2.5 Data Interpretation***

Normally, queried data only contains table column names and numerical values, which is

unrealistic to understand if it is directly presented. Thus, the queried data combined with the user's question is analyzed and interpreted into a traffic advisory report. This process not only necessitates GPT's prior comprehension on the input question, but also incorporates a wide range of pre-trained transportation knowledge and preliminary knowledge of database contexts embedded in the prompts. The interpretation generates human-language responses adopting detailed explanations on queried data.

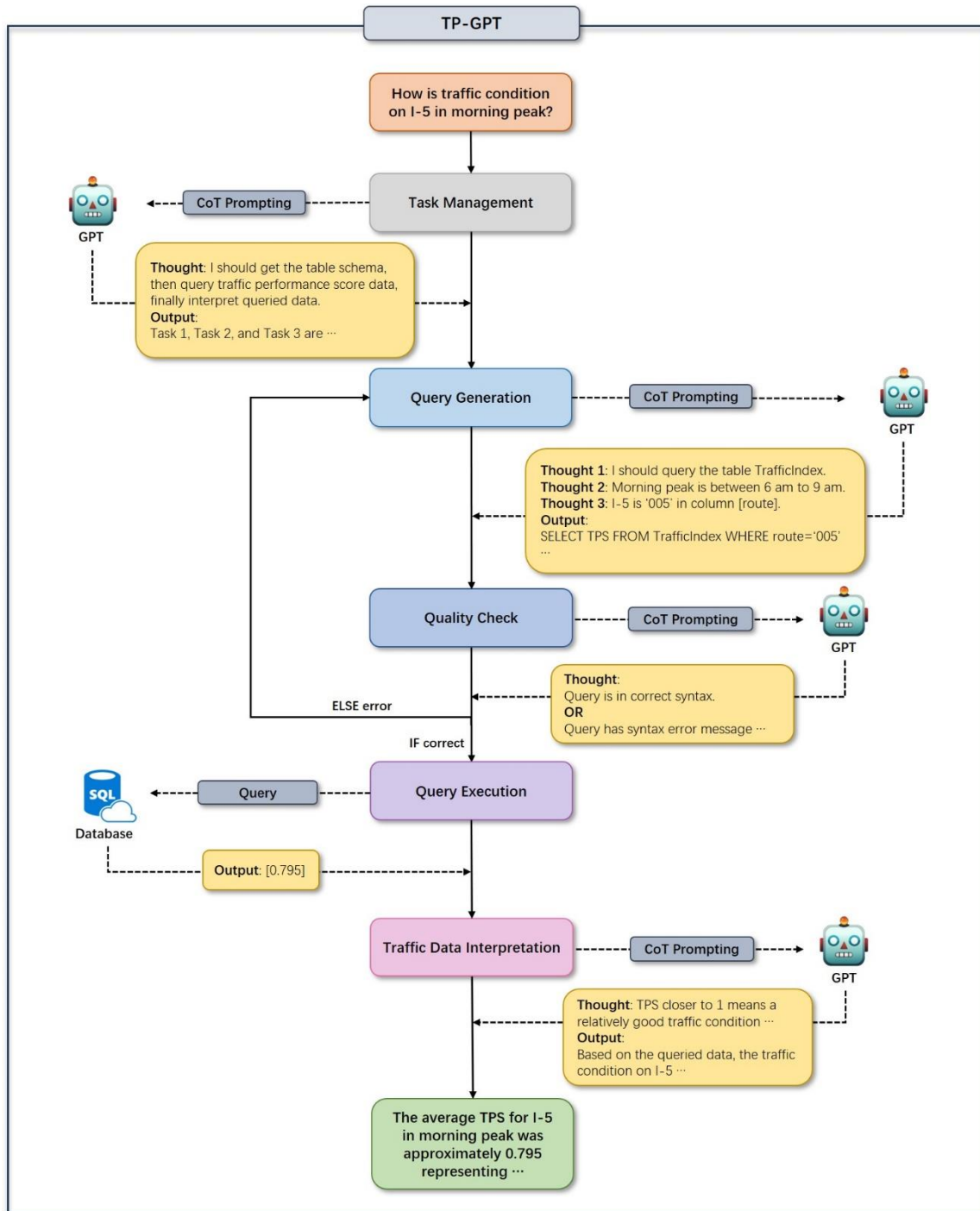


Figure 3.2 System architecture design demonstrating traffic analysis pipeline

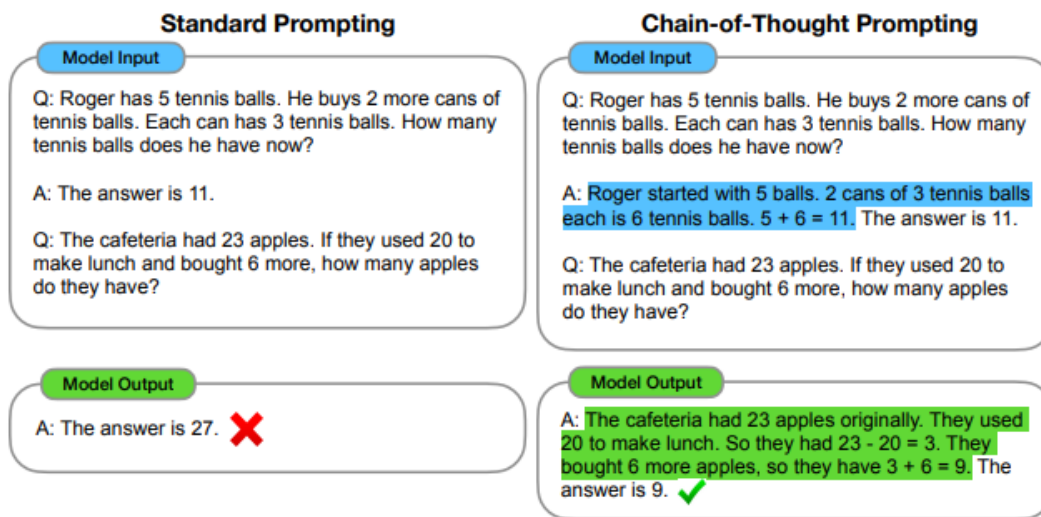
### **3.3 INPUT PROMPT GENERATION**

Prompt engineering is essential for LLMs to master preliminary contextual information before a practical question is fed. Such integration can not only enable language models to avoid the unnecessary repeating process of reasoning users' desired responses in each run when the contexts (i.e., database setup, question background) are constant, but also boost the ChatBot performance in the transportation domain. In the proposed framework, the iterative interface with GPT is achieved through the Chain of Thought (CoT) prompting (Wei, et al., 2022), which iteratively reasons a series of intermediate steps before the final answer is generated.

#### ***3.3.1 Chain-of-Thought Iteration***

Chain-of-thought prompting, a technique originally proposed to enhance the performance of LLMs, has demonstrated significant improvements in complex arithmetic calculations, commonsense answering, and symbolic reasoning tasks by incorporating intermediate inference steps into their processing. Typically, when humans tackle a complex problem, they begin by thoroughly analyzing the textual description to grasp the core objectives of the task. This is followed by a cognitive retrieval of relevant general knowledge, which aids in formulating preliminary thoughts on potential solutions. This mental process is analogous to group brainstorming, where discussions spark innovative ideas and solutions. However, unlike humans, pre-trained language models tend to bypass this incremental

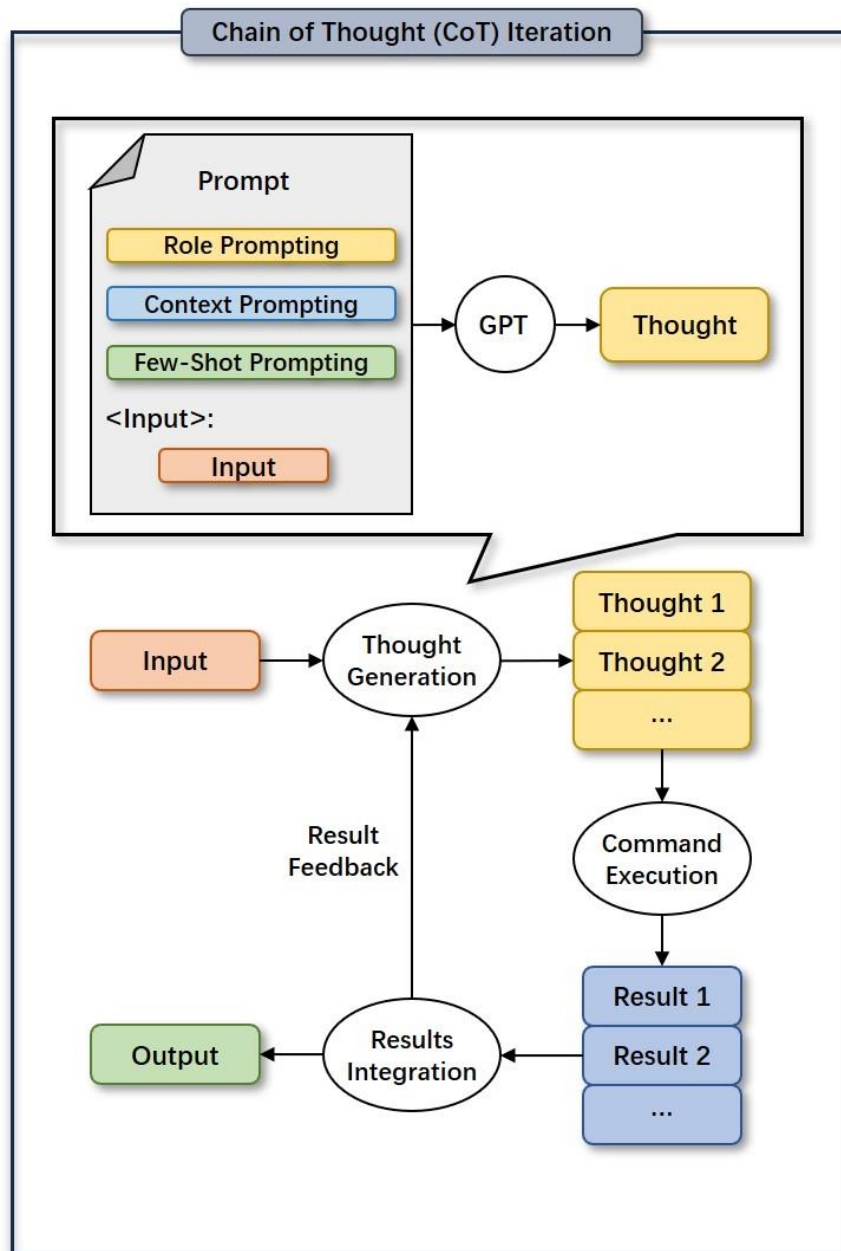
thinking process, preferring instead to generate responses based solely on patterns and information learned during their training. This approach can lead to errors, especially when the models are encountered with particularly tricky questions that require deeper understanding and inference. Chain-of-thought prompting addresses this gap by teaching LLMs to imitate human-like reasoning. By systematically breaking down a question into a series of interpretable thought steps, these models are trained to build a logical pathway that leads to the final answer, similar to a brainstorming session. Incorporating this explicit reasoning process not only makes the models' operations more explainable but also significantly enhances their ability to analyze and resolve questions with a higher level of accuracy and details. As a result, the outputs are generated in a notably more reliable, natural, and correct manner. This methodology significantly improves the practical utility of LLMs in real-world applications by enhancing their performance in handling complex scenarios.



**Figure 3.3 Chain-of-thought prompting with intermediate reasoning highlighted**  
(Wei, et al., 2022)

During exploration stage, we found that the common way of integrating ChatGPT into coding generation and commonsense reasoning is to input the demands with prompts, then results are generated sequentially, without refinement and reflection. This approach is tested unstable and unreliable as higher randomness is retained in the process. If the output results fail to correctly address input demands, no further actions are invoked to make modifications. To tackle this shortcoming, in TP-GPT, iterative interactions between different modules are designed and built upon the concept on CoT prompting for higher communication efficiency and answering accuracy. The fundamental conceptual workflow of CoT iteration integrated in TP-GPT is shown in Figure 3.4. Compared to the normal method, CoT iteration generates intermediate actionable thoughts on the commands to be executed. With a cycle of result feedback, the output is iteratively reviewed for regenerating updated thoughts on any potential adjustments and improvements to finetune the results

before they are finally output. Thoughts are generated by ChatGPT with CoT prompting, as well as customized role prompting, context prompting, and few-shot prompting to boost its expertise in the certain setup of transportation domain, which will be illustrated with more details in the following sections. Upon completion of CoT iteration, actionable thoughts along with the output results, such as SQL queries, retrieved domain knowledge to insert into the answer, debugging advice on any defects in codes, or instructions of next-step actions, are transmitted to the system for executions, of which results are iteratively refined in the aforementioned feedback loop. This scheme ensures that generative capabilities of LLMs are better explored and leveraged to obtain more reliable and accurate output efficiently.



**Figure 3.4 Chain of Thought prompting**

### ***3.3.2 Role Prompting***

The designed prompts are structured to provide comprehensive descriptions across three key aspects: the role to be performed, the contextual information relevant to the task, and a set of few-shot learning exemplars. An example template utilized to create these

customized prompts are shown in Figure 3.5. Role prompting is the most critical technique to inform ChatGPT of its expected duty in solving sub-tasks deconstructed from the original question. Detailed instructions on what role the model is expected to play in performing the tasks and how to generate the object responses are included in the prompt.

The first step in the prompt design involves explicitly defining the role ChatGPT is to assume, such as "*intelligent traffic assistant*". This helps set clear expectations about the nature of the tasks it needs to handle. Next, the scope of work is detailed, instructing the model on the specific actions to undertake, such as "*generating correct MSSQL queries by identifying relevant questions*". This step ensures that ChatGPT focuses on the correct elements of the problem. Lastly, the desired output is specified, directing the model to produce "*correct answers based on queried data and your knowledge in transportation*".

By structuring the prompts in this manner, ChatGPT is better equipped to understand the tasks it is expected to perform and can avoid producing outputs that are vague or unrelated to the user's demands in transportation inquiries. This approach enhances the relevance and accuracy of the responses, aligning them more closely with the specific needs and contexts of the inquiry.

### ***3.3.3 Context Prompting***

Context prompting is an essential strategy that equips the model with in-depth details about the environmental setup and the necessary contextual information. This preparation enables the model to handle incoming queries more effectively, reducing the need to engage in

time-consuming searches for related information when a question is posed. An emphasis is laid on detailed descriptions of the sophisticated database setup to streamline the interaction process. Frequently querying the database schema for column names, attributes, or content semantics each time data is retrieved not only represents redundant work but also consumes additional computing resources due to the execution of unnecessary queries.

To mitigate this inefficiency, context prompting incorporates comprehensive explanations of table relationships, column specifics, and the significance of data. For instance, abstract names like “*cabinfo*”, which stands for cabinet information, might be unclear without proper context; thus, clarifications are provided within the prompt to ensure understanding. Additionally, the prompt includes practical tips aimed at enhancing query efficiency. An example of such advice is the recommendation to “*always create queries in time-descending order*”, which helps to focus on the most recent data, especially critical when dealing with real-time traffic conditions. This approach avoids the retrieval of vast amounts of historical data, thereby optimizing computational efficiency.

Moreover, the prompt includes definitions of key terminology from the transportation domain to improve the model’s comprehension and application of metric data. For instance, Traffic Performance Scores (TPS) are integrated to evaluate traffic conditions, but without a clear definition of what TPS stands for or the range of values indicating optimal conditions, the model might overlook this crucial information. Therefore, the prompt explicitly defines such metrics. Similarly, the significance of loop detector IDs, which are composed of numbers and characters that denote specific details such as highway name,

mileage number, direction, or loop type, is elaborated. Detailed instructions are included to enable the model to decode this information accurately. By embedding this crucial context, the model is better prepared to process and respond to inquiries with greater precision and relevance.

Evidently, well-constructed prompts enhance the capability of ChatBot to produce correct outputs, which increases the answering accuracy. Typically, the inclusion of more detailed information within prompts leads to a remarkable improvement in the quality of the responses generated. However, this comes with a side-effect: lengthy prompts can consume a significant number of tokens, which may adversely affect computing efficiency by using additional computational resources. This necessitates a careful balance between the depth of information provided in the prompts and their textual length. Achieving this balance is crucial, as overly verbose prompts might slow down the response time and reduce the efficiency of the model.

In our approach to designing context prompts, we have strategically omitted extra narratives, focusing solely on the essential information that the language model requires to function effectively. This selective inclusion ensures that the prompts remain concise yet sufficiently informative to guide the ChatBot in producing accurate and relevant responses. By streamlining the content of the prompts, we aim to maintain high computational efficiency while still capturing the necessary details to support the accuracy of the ChatBot outputs. This method not only optimizes performance but also leverages the model's capabilities as much as possible, ensuring that it operates within the optimal balance of

accuracy and efficiency.

### ***3.3.4 Few-Shot Prompting***

Few-shot prompting is a method that enhances the performance of ChatGPT by clearly specifying the output format and embedding a selection of highly relevant examples directly into the prompts. This technique not only guides ChatGPT in structuring its responses but also provides exemplary insights that improve the accuracy and relevance of outputs.

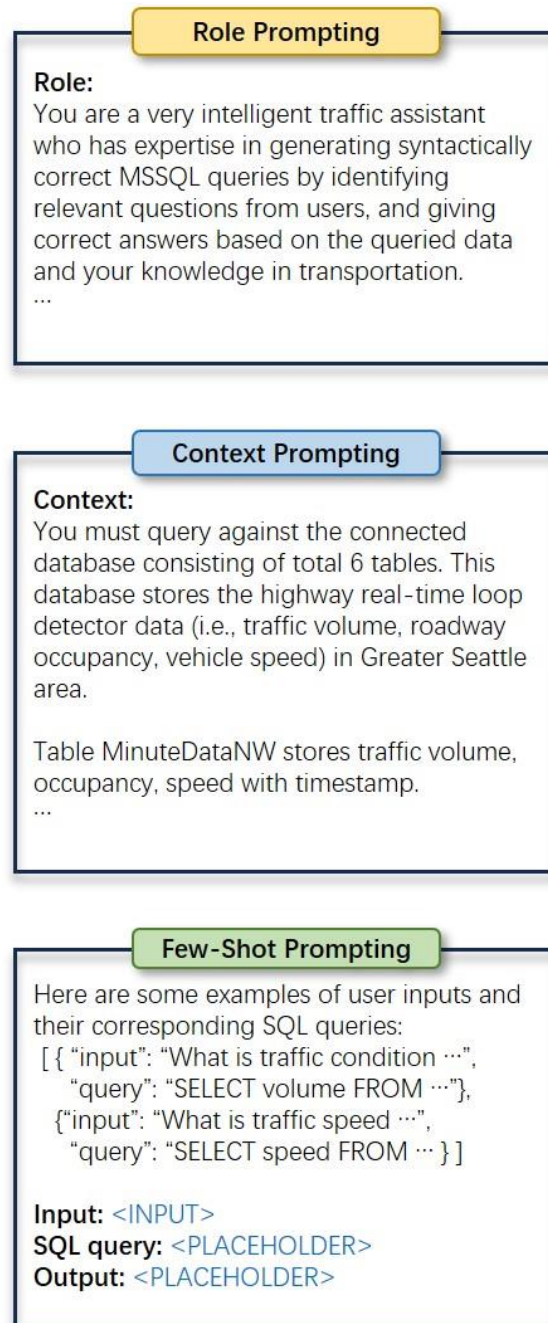
The output format is organized with three key components: *Input*, *SQL Query*, and *Output*. These elements serve distinct purposes:

- **Input:** Captures the user's question, centering the problem that needs to be addressed.
- **SQL Query:** SQL query to be generated based on the input. This awaiting part will bridge the gap between the natural language question and the database data retrieval required to answer the question.
- **Output:** Outlines the human-language response that ChatGPT should produce after executing the SQL query, providing a clear, comprehensible answer to the user's initial inquiry.

Additionally, the incorporation of the most relevant examples from a pre-established few-shot example inventory further enhances the model's ability to respond effectively. This inventory is manually assembled and includes approximately 20 typical questions along

with their corresponding SQL queries. These examples perform as references for ChatGPT to understand and align its responses with expected patterns and formats.

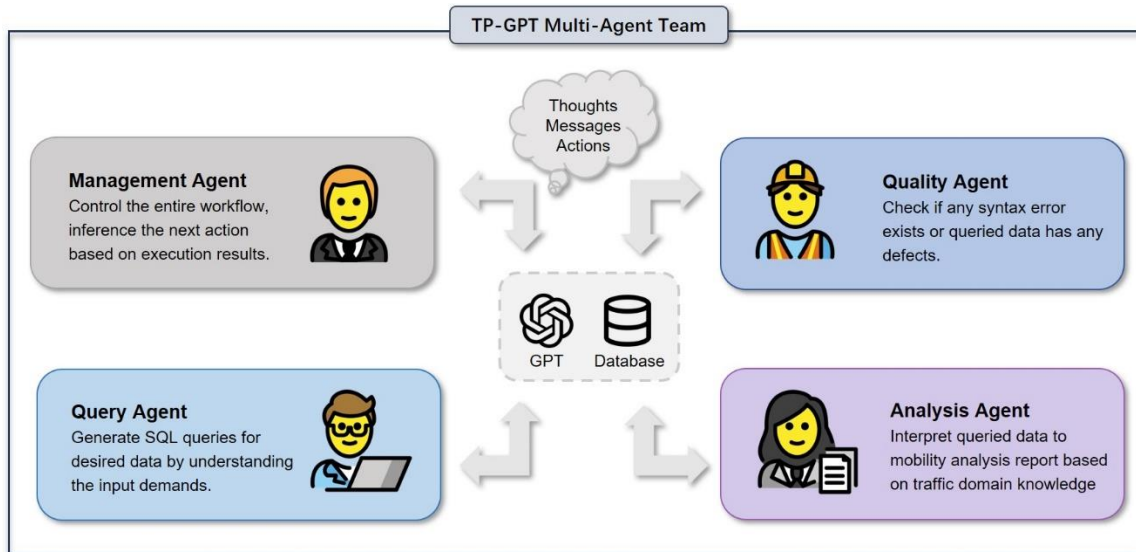
The concept for each prompting design, as illustrated in Figure 3.5, is expanded into complete prompts for actual implementation. By dynamically embedding these relevant input-query pairs into the prompt based on the current user's question, the few-shot prompting ensures that ChatGPT has immediate access to similar scenarios. This method not only aids in generating more accurate SQL queries but also in crafting responses that are contextually appropriate and technically precise, significantly boosting the overall answering performance of the model.



**Figure 3.5 Input prompt template**

### **3.4 MULTI-AGENT STRATEGY**

GPT achieves superior performance in solving individual tasks following instructional logic in a dialogue. However, highly sophisticated problems in real-world traffic analysis are challenging to tackle by directly adopting GPT's general intelligence. Here we introduce the multi-agent strategy in transportation scenarios, an innovative collaborative workflow incorporating GPT as different agents to work as a team, which simulates the human's real teamwork logistics in a research lab or industrial company. The overall objective is to decompose the original complex tasks into sub-tasks of different scopes and then assign each to a GPT agent expertized in the designated domain to solve by a streamlined, collaborative workflow. Various expertized agents are built upon GPT by leveraging role prompting described in Section 3.3.2 to inform of expected agent role to be performed in the respective deconstructed sub-tasks. The interaction between agents is enabled and achieved by incorporating a public scratchpad for the agent team to track the updated progress. Agents iteratively communicate with GPT and database based on the CoT prompting dialogue in JSON format, illustrated in the previous Section 3.3.1, to seek advisory and thoughts on task solutions and then formulate the execution plan.



**Figure 3.6 TP-GPT multi-agent collaboration framework for traffic data analysis**

The joint collaborative framework includes four virtual agents, shown in Figure 3.6, each of which conducts the designated scope of tasks corresponding to a teamwork role: 1) Project manager controls the entire team's workflow, which begins with deconstructing the original problem into different-domain fine-grained tasks for assignment, then manages solution logics by reasoning the next-step actions based on the execution results from the last step. Manger agent plays a vital role in connecting diverse system modules to ensure tasks are conducted continuously in the correct order; 2) SQL engineer generates synthetically correct queries for the desired data through contextual understanding of the users' input demand, system environment setup, and instructional thoughts form manger agent. The query agent leverages the programming capacity of GPT especially in SQL syntax, which is pre-trained on the massive online data sources containing extensive SQL-related training cases, to demonstrate the reliable query scripts generation features; 3) Quality analyst checks if any syntax error exists in the generated query that may cause the

system run interruption. The process differs from simple errors and exceptions handling in programming languages, which is only to transmit error names to users without the capability to correct emerged unexpected situations or to restore to the normal condition. Yet, the quality agent is empowered by GPT's intelligence to be able to read problematic codes and to debug the possible defects by understanding error messages. Specifically, necessary inspections into backend database execution log are implemented to figure out the potential cause to the error in order to correct the query, which ensures iterative checking and correcting process on generated queries until target data is successfully executed. Furthermore, the relevance of obtained data to the input question is measured by comparing data attributes to user's input. If data is not reasonably acceptable, the manager agent is activated to rearrange the entire workflow to generate an updated query by understanding the context in a corrected way; 4) Finally, data analyst interprets the queried data to mobility analysis report based on GPT's extensive pre-training in the transportation sector, to answer the user's demanded information in a natural-language illustrative method with real-time data and detailed explanation involved. The user's question is input along with the extracted data to the analyst agent to compose accurate and contextually appropriate answers. Contextual descriptions in the designed prompts provide the necessary preliminary information to understand and explain the data.

This multi-agent collaborative strategy significantly enhances the overall performance of the ChatBot in terms of accuracy, reliability, interpretability, and flexibility. Each agent's specialized role and the structured workflow ensure a seamless and effective response

generation process, tailored to meet the specific demands of users in the transportation domain.

### **3.5 FEW-SHOT LEARNING**

Although LLMs exhibit impressive zero-shot capabilities for general problems, they struggle to address complex problems in the transportation domain using only a zero-shot approach. Few-shot prompting can be used as a technique to facilitate in-context learning by providing examples that guide the model toward better performance. The demonstrations in the prompt contain exemplar user questions with their ideal queries crafted by developers that the model is expected to follow to generate subsequent responses. In each attempt to resolve the input question, the model vectorizes the user's input to text embeddings in order to retrieve several most relevant examples from inventory to insert into prompts for reference, which not only enables the model to master the contextual knowledge required to answer similar questions accurately but also improves response efficiency by significantly decreasing reasoning time and the risk of causing errors. To further enhance response reliability leveraging few-shot learning, an example repository covering multiple transportation scenarios is manually crafted as shown in Figure 3.7, including real-time traffic advisory, historical data statistical analysis, travel emission inquiry, lane-based traffic performance inspection, and so on, for providing comprehensive transportation evaluations. In implementations, the input question is converted to text embedding to search the designated number of most similar questions in the example

repository. Then, these question-query example pairs are used to populate the generated prompt for model inference. A typical few-shot prompt is displayed in Figure 3.5. This approach avoids offering irrelevant examples to target work scope, thereby increasing the few-shot efficiency and accuracy, as well as reducing the possibility of non-objective examples misleading the response. The integration of few-shot learning remarkably enhances ChatBot performance, especially when questions that are typically asked fall into the pre-defined scope. This also provides the potential for further extension of objective response domain by simply adding more exemplary use cases.

```

examples = [
  {
    'input': "Show me traffic statistics for I-5 during this morning's peak hours.",
    'query': "SELECT AVG(MDNW.vol) AS AverageVolume, AVG(MDNW.occ) AS AverageOccupancy, AVG(MDNW.spd) AS AverageSpeed "
      "FROM MinuteDataNW MDNW INNER JOIN cabinfo CBIN ON MDNW.id = CBIN.StationID "
      "WHERE CBIN.Route = 5 AND CAST(MDNW.time_stamp AS DATE) = CAST(GETDATE() AS DATE) AND CAST(MDNW.time_stamp AS TIME) BETWEEN '06:00:00' AND '09:00:00'",
  },
  {
    'input': "What was the traffic condition of I-90 during today's evening peak?",
    'query': "SELECT AVG(STI.TrafficIndex_GP) AS AvgTrafficIndex_GP, AVG(STI.TrafficIndex_HOV) AS AvgTrafficIndex_HOV "
      "FROM SegmentTrafficIndex STI INNER JOIN Segments S ON STI.segmentID = S.segmentID "
      "WHERE S.route = '90' AND CAST(STI.time AS DATE) = CAST(GETDATE() AS DATE) AND CAST(STI.time AS TIME) BETWEEN '16:00:00' AND '18:00:00'",
  },
  {
    'input': "What is the traffic speed on SR 520 now?",
    'query': "SELECT AVG(STI.AVG_Spd_GP) AS AverageSpeed_GP, AVG(STI.AVG_Spd_HOV) AS AverageSpeed_HOV "
      "FROM SegmentTrafficIndex STI INNER JOIN Segments S ON STI.segmentID = S.segmentID "
      "WHERE STI.time BETWEEN DATEADD(minute, -3, GETDATE()) AND DATEADD(minute, -2, GETDATE()) AND S.route = '520'",
  },
  {
    'input': "Query the most recent loop data on SR 99 Northbound sorted by locations.",
    'query': "SELECT DF.location AS Location, AVG(MDNW.vol) AS AverageVolume, AVG(MDNW.occ) AS AverageOccupancy, AVG(MDNW.spd) AS AverageSpeed "
      "FROM MinuteDataNW MDNW INNER JOIN MinuteDataDefnNW DF ON MDNW.id = DF.id "
      "WHERE DF.route = '99' AND DF.direction = 'NB' AND MDNW.time_stamp BETWEEN DATEADD(minute, -3, GETDATE()) AND DATEADD(minute, -2, GETDATE()) GROUP BY DF.location",
  },
  {
    'input': "Query the most recent loop data on SR 99 Southbound sorted by locations.",
    'query': "SELECT DF.location AS Location, AVG(MDNW.vol) AS AverageVolume, AVG(MDNW.occ) AS AverageOccupancy, AVG(MDNW.spd) AS AverageSpeed "
      "FROM MinuteDataNW MDNW INNER JOIN MinuteDataDefnNW DF ON MDNW.id = DF.id "
      "WHERE DF.route = '99' AND DF.direction = 'SB' AND MDNW.time_stamp BETWEEN DATEADD(minute, -3, GETDATE()) AND DATEADD(minute, -2, GETDATE()) GROUP BY DF.location",
  },
  {
    'input': "Query the sorted loop data by locations on SR 520 Eastbound at noon today?",
    'query': "SELECT DF.location AS Location, AVG(MDNW.vol) AS AverageVolume, AVG(MDNW.occ) AS AverageOccupancy, AVG(MDNW.spd) AS AverageSpeed "
      "FROM MinuteDataNW MDNW INNER JOIN MinuteDataDefnNW DF ON MDNW.id = DF.id "
      "WHERE DF.route = '520' AND DF.direction = 'EB' AND CAST(MDNW.time_stamp AS DATE) = CAST(GETDATE() AS DATE) AND CAST(MDNW.time_stamp AS time) = '12:00:00' GROUP BY DF.location",
  },
  {
    'input': "How was the traffic performance on I-405 Northbound at 9pm yesterday?",
    'query': "SELECT AVG(STI.TrafficIndex_GP) AS AverageTrafficIndex_GP, AVG(STI.TrafficIndex_HOV) AS AverageTrafficIndex_HOV "
      "FROM SegmentTrafficIndex STI INNER JOIN Segments S ON STI.segmentID = S.segmentID "
      "WHERE CAST(STI.time AS date) = DATEADD(day, -1, CAST(GETDATE() AS date)) AND CAST(STI.time AS time) = '21:00:00' AND S.route = '405' AND S.direction = 'N'",
  },
  {
    'input': "Show me the most recent 10 loop data at I-5 Southbound milepost 140.15.",
    'query': "SELECT TOP(10) * "
      "FROM MinuteDataNW MDNW INNER JOIN MinuteDataDefnNW DF ON MDNW.id = DF.id "
      "WHERE DF.route = '5' AND DF.milepost = '140.15' AND DF.direction = 'SB' ORDER BY MDNW.time_stamp DESC",
  },
]

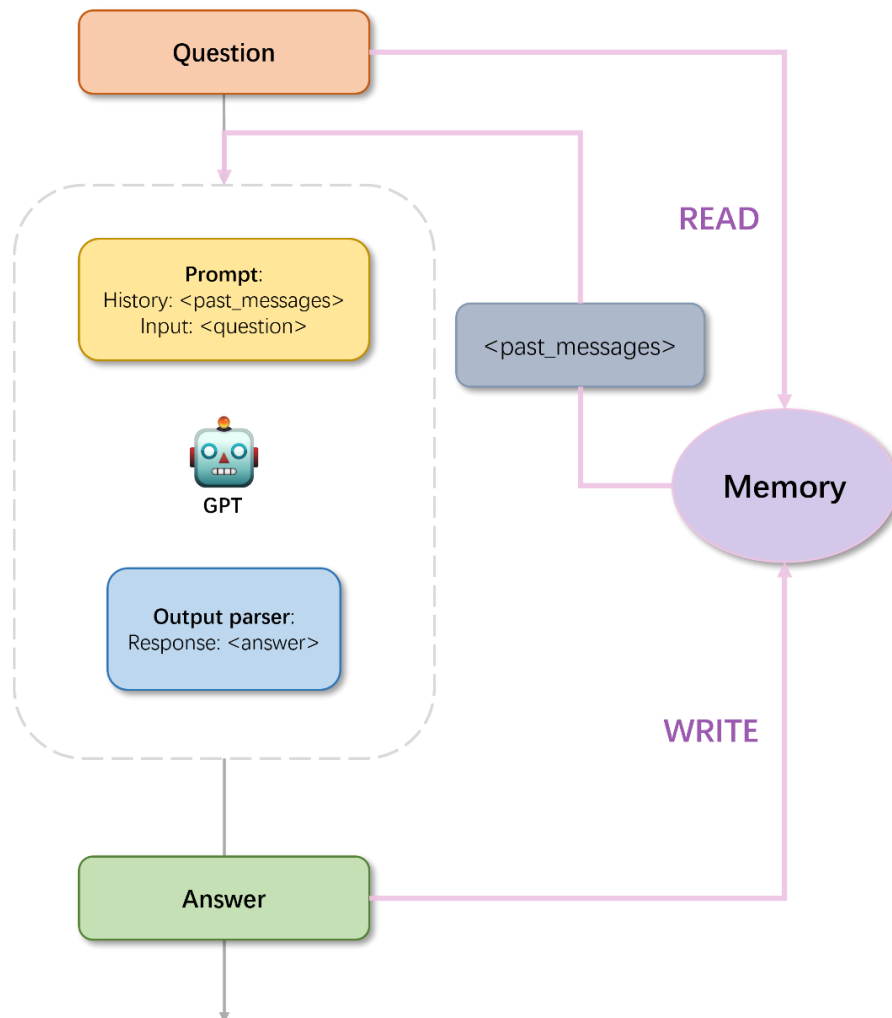
```

Figure 3.7 Few-shot learning exemplar inventory

### 3.6 CHAT MEMORY

To enhance interactivity, the ChatBot should retain a memory of past conversations, allowing it to infer and tailor future responses based on previous interactions. In most use cases, if the initial generated response does not fulfill users' expectations, normally subsequent questions will follow, assuming chat history is held. Thus, the ChatBot is integrated with the chat memory module by saving dialogue history in each conversation session. As shown in Figure 3.8, once a question is input, the system utilizes the memory management protocol to access the chat history. It extracts relevant records from previous interactions, which are then strategically combined with the new input in the prompts. This integrated approach allows the system to utilize historical context, enhancing the relevance and specificity of its reasoning process. Subsequently, the system generates an answer by referring to the relevant chat history embedded within the prompts. This ensures that the response not only addresses the current query but also reflects an understanding of previous exchanges, providing a more coherent and contextually appropriate answer. Generated answers are structurally stored in memory, creating a comprehensive knowledge base. This repository aids in future retrieval, enabling the system to reference past answers efficiently and enhance its responses to new inquiries. Specifically, only the part of chat records relevant to the current input question is retrieved from memory for the model's reference in response, which prevents from reading the whole lengthy dialogue list. Afterward, the generated answer, along with the question, is written into memory. This simple yet effective

implementation of storing and retrieving information from memory contributes to the system's learning and adaptability, allowing it to improve continuously over time. Chat memory module advances the ChatBot towards more efficient and intelligent interaction experience.



**Figure 3.8 Reading and writing chat memory to integrate past knowledge to current answer reasoning**

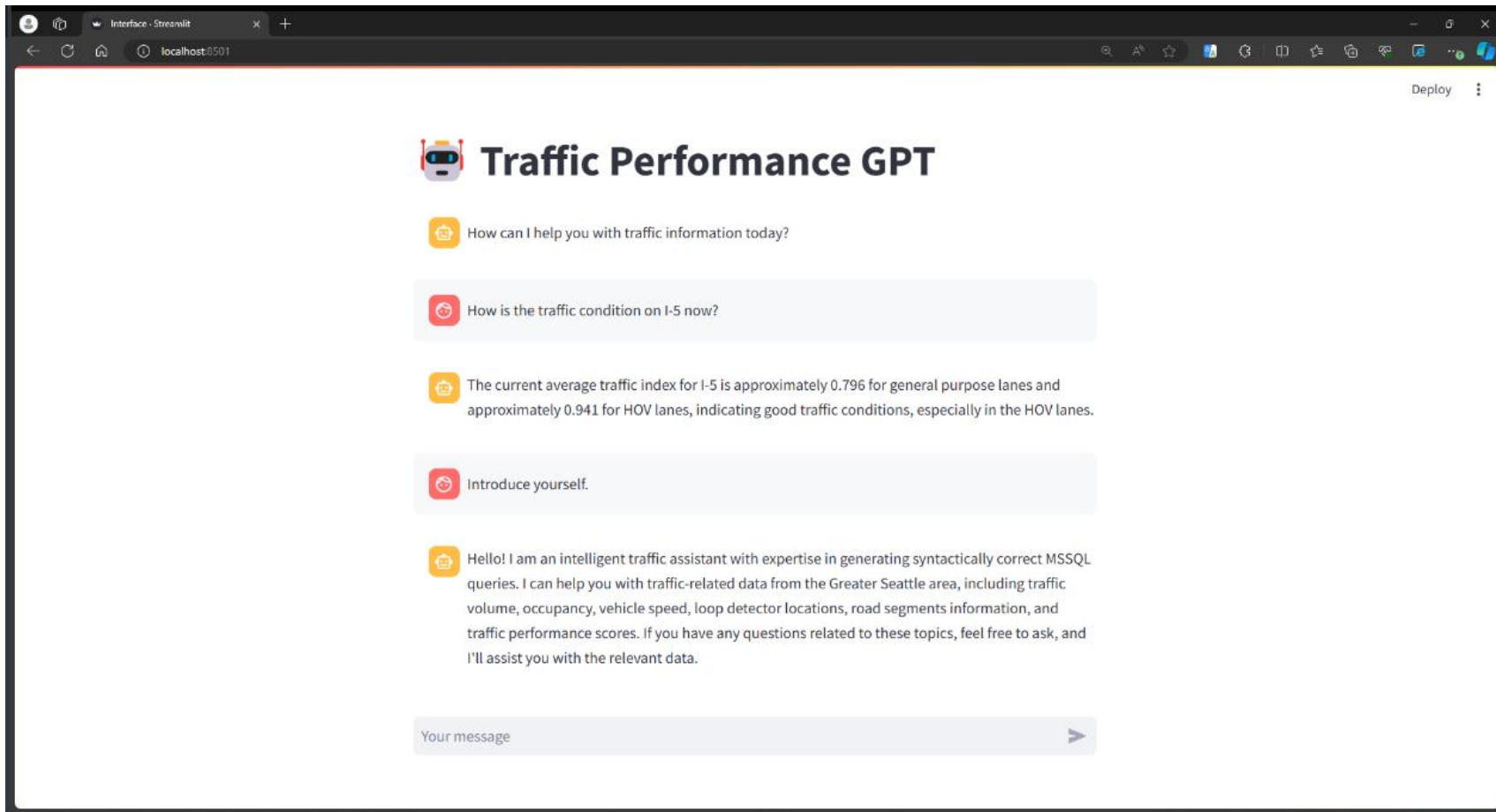
### 3.7 CHATBOT INTERACTION WEB INTERFACE

An innovative online ChatBot featuring an interactive web interface has been developed using the Streamlit framework in Python. This web interface is designed to maintain a continuous dialogue history, enabling it to respond to user inquiries by leveraging past interactions. An illustrative example of a conversation between a user and the TP-GPT chatbot is depicted in Figure 3.9. The response time for each question ranges from approximately half a minute to three minutes, varying with the complexity involved in processing the question.

In the technical setup behind the scenes, user inputs are transmitted to a computing server via the web interface. Here, the inputs are processed by the TP-GPT architecture that integrates strategic enhancements with the *gpt-4-turbo* model which is latest released version as of the study is conducted, known for its exceptional performance. The answers generated by TP-GPT are then sent back to the user's client web browser for both display and storage within the session memory. This storage facilitates reference in subsequent conversations, enhancing the continuity and contextuality of interactions. The ChatBot's backend leverages the OpenAI Application Programming Interface (API) to integrate the ChatGPT models effectively. This implementation not only utilizes the performance enhancement of the TP-GPT configuration but also incorporates the latest advancements in AI as provided by the *gpt-4-turbo* version, ensuring top-tier performance.

To quantitatively assess the benefits of the TP-GPT system over standard GPT models,

an experimental study is carried out to compare their performance in the next chapter. This study involves using the ChatBot interface to evaluate and compare its performance with baseline models.



**Figure 3.9 Online web application interface of TP-GPT ChatBot**

## CHAPTER 4: EXPERIMENT

### 4.1 EXPERIMENTAL SETUP

To demonstrate the model's performance in solving transportation-specific domain problems, we introduce our self-generated challenging traffic analysis benchmark named TransQuery based on the mobility database as mentioned in Section 3.1. The benchmark includes 50 manually designed complicated transportation surveillance and management tasks in real-world scenarios. Each instance is an input-output pair consisting of a question and its ground-truth response (i.e., SQL query, target data, and natural language answer). These tasks cover a wide scope of traffic analysis challenges including but not limited to: spatial-temporal traffic condition inquiry, traffic pattern recognition, peak traffic behavior analysis, real-time lane-based trip advisory, traffic counting statistics, and vehicle emission evaluation. Some of the example problems are listed in the Table 4.1. To demonstrate the validity of employing TP-GPT framework, we use TransQuery benchmark to compare our method with GPT-4 Turbo (OpenAI, 2023), PaLM 2 (Anil, et al., 2023), and SQLCoder (Ping, Aw, & Srivastava, 2023) which are state-of-the-art baseline LLMs having expertise in the database query domain. Standard prompting containing fundamental database information and contextual knowledge is integrated into baseline models.

**Table 4.1 Sample questions in TransQuery**

<b>Type</b>	<b>Question</b>
<i>Spatial-temporal traffic condition inquiry</i>	Show me the most recent loop data on SR 99 Northbound sorted by locations.
<i>Traffic pattern recognition</i>	Compare traffic performance of I-5 between Monday and Sunday in last month.
<i>Peak traffic behavior analysis</i>	What was the traffic condition of SR-520 during today's evening peak?
<i>Real-time lane-based trip advisory</i>	Should I use HOV or general-purpose lane on I-5 now?
<i>Traffic counting statistics</i>	How many cars are on each segment of I-405 on average now?
<i>Vehicle emission evaluation</i>	What is the difference in greenhouse gas emissions between weekdays and weekends in last month?

To statistically compare the performance of each model on TransQuery, the execution result of each instance of the benchmark is evaluated by the metrics with corresponding rate scores: **Non-functional**: SQL query is non-executable due to existing errors, with rate score of 0; **Runnable but imperfect**: SQL query can be executed successfully but not perfectly answers the input question, with rate score of 1; **Flawless**: correct data can be

queried, and interpretation of data can properly answer the input question, with rate score of 2. In this way, an average performance score  $S$  is calculated for each model using the Equation (4.1:

$$S = \frac{\sum_i n_i s_i}{s_{max} \sum_i n_i} \quad (4.1)$$

where  $i$  represents evaluation categories (e.g., non-functional),  $n$  represents the number of experimental instances classified to each category, and  $s$  represents their corresponding rate scores.

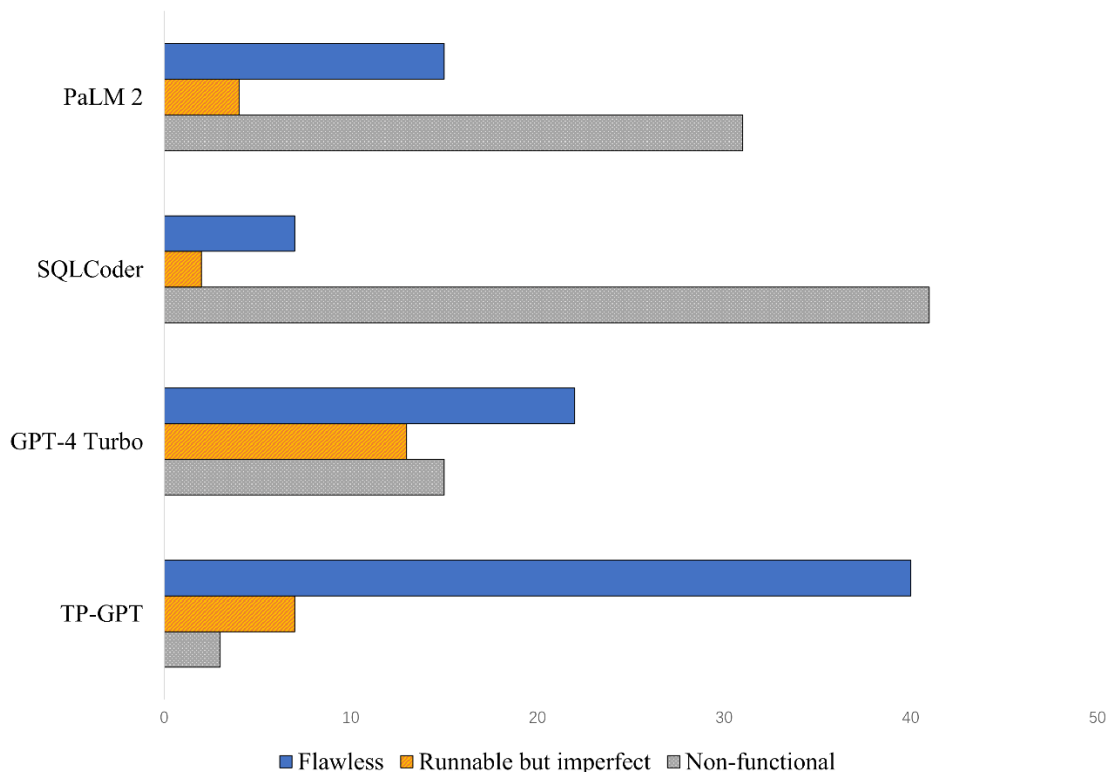
## 4.2 PERFORMANCE COMPARISON

In experiments, three baseline models are compared to TP-GPT using TransQuery benchmark that has 50 complex transportation problems. As shown in Table 4.2, TP-GPT significantly outperforms other baseline models on TransQuery with an average performance score of 0.87, surpassing GPT-4 Turbo of 0.57, PaLM 2 of 0.34, and SQLCoder of 0.16. Specifically, out of total 50 questions, TP-GPT generates 40 flawless responses accounting for 80% of the whole dataset, 7 runnable queries but imperfect responses, and only 3 non-functional queries. Whereas, the state-of-the-art LLM GPT-4 Turbo only generates 22 flawless responses accounting for 44% of the whole dataset, even with superior pre-trained reasoning and contextual capability. PaLM 2 and SQLCoder barely generate 15 and 7 flawless responses accounting for the percentage of 30% and 14% respectively. From Figure 4.1, the comparison results of TP-GPT and three baseline models

are visualized as bar plot by statistically counting the generated answers in each category from metrics. This plot evidently shows the significant enhancement of performance TP-GPT achieves on TransQuery benchmark, especially regarding the higher flawless response rate and lower non-functional response rate.

**Table 4.2 Performance comparison of TP-GPT with baseline models on TransQuery traffic-analysis benchmark**

Model	Question			Average
	Non-functional	Runnable but imperfect	Flawless	Score
PaLM 2	31 (62%)	4 (8%)	15 (30%)	0.34
SQLCoder	41 (82%)	2 (4%)	7 (14%)	0.16
GPT-4 Turbo	15 (30%)	13 (26%)	22 (44%)	0.57
<b>TP-GPT</b>	3 (6%)	7 (14%)	40 (80%)	0.87



**Figure 4.1 Statistical experimental results of TP-GPT and baselines on TransQuery benchmark**

This result not only demonstrates the complicity and challenge of TransQuery benchmark, but also indicates TP-GPT's superior performance in resolving transportation scenario problems and its effectiveness when GPT is integrated with our developed ChatBot system. A detailed comparison between the models is narrated in the following.

- PaLM 2:** The most capable PaLM 2 model in code generation named Bison, is tested as one of the baselines due to its compatibility with SQL query generation. This model has the fastest response time for its relatively small model size. However, the performance struggles in dealing with complex tasks requiring overloaded tokens that frequently exceed the limit. For example, the question *"Compare traffic performance of I-5 between Monday and Sunday in last month"*

requires querying a vast amount of minute-by-minute historical data in one month to summarize the traffic performance. This data volume cannot be processed by Bison causing the token limit error.

- **SQLCoder:** Defog's SQLCoder-34B is implemented for text-to-SQL performance comparison as one of baselines. The model is constrained by its model size achieving the lowest solve rate. It could have been trained using the dataset that consists of MySQL-syntax samples other than MSSQL dialect, which is potentially the major factor of disadvantageous performance. However, SQLCoder provides great possibilities of future fine tuning to adapt to the specific domain due to its open-source origin.
- **GPT-4 Turbo:** In order to compare TP-GPT with a model that is not limited by model size, we conducted experiments on the up-to-date GPT model, GPT-4 Turbo, which has the 128,000 token limit. This model is commonly regarded as the most powerful generative intelligence by the public for its extensive pre-trained knowledge in various domains, absolutely including transportation and SQL query programming. In the experiment, the fundamental prompt containing contextual information such as description of the expected role to perform, database setup, and table schema is input to the model's memory. The result shows that GPT-4 Turbo achieves a better performance than PaLM 2, especially regarding the lower non-functional rate. The strong reasoning and coding capabilities enable the model to generate more executable SQL queries. However, obtaining flawless answers with

correctly queried data is still challenging for its low solve rate compared to TP-GPT.

A fair portion of queries are runnable yet do not properly answer the input questions.

- **TP-GPT:** Our innovative system TP-GPT, integrating intelligence-boosting strategies to regular GPT-4 model without need of fine-tuning, achieves the highest performance among powerful models. In experiments, TP-GPT is capable of crafting lengthy SQL statements within minutes and also interpreting data to answer tricky questions that require deduction and derivation. For example, TP-GPT tackles the question “*What is the difference in greenhouse gas emissions between weekdays and weekends in the last month?*” by estimating emissions based on Vehicle Miles of Travel (VMT) using a conceptual formula.

Several observations are captured during experiments that potentially account for the superior performance of TP-GPT over GPT-4 Turbo: 1) TP-GPT's multi-agent strategy, especially the quality agent, enhances the query correctness by checking syntax errors. However, GPT-4 Turbo suffers from frequently executing incorrect queries without error checking. It also struggles to recognize the correct column name of a table even when the schema has been input; 2) TP-GPT's implementation of Chain of Thought enables the model to revise the response in multiple rounds of iterative communication. On the contrary, GPT-4 Turbo merely processes data in a sequential flow without the review loop, which causes a lack of understanding of the database environment and content. It is unable to inspect if the query will successfully extract correct data, having the potential to obtain empty results; 3) As TP-GPT is prompted by few-shot examples consisting of template

questions and queries, higher proficiency in generating certain SQL dialects (e.g., Microsoft SQL) is observed for the prior knowledge leaned from few-shot learning in consistent contexts. With the absence of this functional module, GPT-4 Turbo fluctuates its performance due to frequently caused errors converting date from string to timestamp in Microsoft SQL syntax.

### **4.3 RESULTS ANALYSIS**

This section analyzes in depth the answers generated by TP-GPT on TransQuery benchmark, in order to explain the details about the intrinsic advantages and weaknesses of TP-GPT based on its mechanism. As previously illustrated, TP-GPT is experimented with 50 challenging traffic-analysis tasks, the execution results and natural-language answers of which have been documented. The typical exemplary question-answer pairs that can represent the performance of TP-GPT are presented in Table 4.3, where answers under all the three categorical ratings are involved. The complete experimental results can be found in Appendix.

In general, TP-GPT is able to comprehensively understand input questions by deconstructing essential elements of tasks, such as route and temporal information, terminology definition, and user demands. Specifically, several observations are gathered regarding the strengths of the proposed system: 1) TP-GPT is sensitive to specific route names, such as “*I-5*”, “*SR 520*”, and then can retrieve accurate freeway data corresponding to the input. As route names are encoded to abstract numbers in the database tables, it

means that TP-GPT has the capability to decode them based on contextual information provided in the prompts and integrate into query generation. 2) TP-GPT perceives date or time-related information precisely, not only for explicit temporal symbols, but also for implicit human-language indications of a time period or timestamp that requires several further steps of reasoning. For a good example, some of input questions involve inquiry about peak-hour traffic conditions, which demands TP-GPT to equip with a clear understanding of the definition of peak hours specifically in Seattle city. As can be observed in the outcome queries, TP-GPT knows well that peak hours are 6 to 9 AM in the morning and 4 to 7 PM in the evening, which aligns with the commonsense. Additionally, TP-GPT makes almost zero mistakes in transforming the days of the week to MSSQL timestamp data format, indicating a reliable performance of interpreting demands to database queries, due to the solid coding foundation gained from language model pre-training process. 3) TP-GPT acquires powerful inferential ability to reason the concrete objectives out of input questions. For inquiring greenhouse gas emissions, although there is no specific data related to environmental factors, TP-GPT can estimate total emissions of a certain segment of highway by calculating Vehicle Miles Traveled (VMT), which is relatively similar to the way of human thinking. Besides, the system can recognize the dates of federal holidays using its general knowledge base to inference statistical measurements of holiday traffic conditions. 4) TP-GPT achieves satisfactory performance in traffic analysis and advisory. A few questions particularly emphasize general trip-planning advice in real-time traffic contexts, like driving lane choice between HOV or general-purpose lanes. To solve these

tasks, TP-GPT interprets the demands underlying obscure expressions in questions to apparent numerical indicators that are relevant to explanatory response to user's concerns. For instance, the system retrieves present speed and Traffic Performance Score data to provide users with insights to lane-based traffic conditions for optimal advisory travel planning. 5) TP-GPT is activated by the crafted prompts to make flexible utilization of various traffic indicator data stored in the database to reflect intrinsic transportation analytics to users. A simple question like "*how is Seattle's traffic condition now*" triggers the system to responsively query traffic metrics of every respective, such as speed, volume, occupancy, and Traffic Performance Score on either lane, which provides a comprehensive analysis report concluding the real-time situations.

On the other hand, for a minority of complicated tasks in TransQuery, TP-GPT does not perform as expected. Several findings regarding its weaknesses are concluded: 1) Occasionally, TP-GPT may query a database column that spans the entire historical data timeline by accident, leading to excessive runtime errors. This issue typically arises when the system attempts to calculate average values from the data, intending to represent a common condition across the dataset. However, the problem occurs because these queries do not specify a particular time period for the data aggregation. For example, the question "*How do weekday traffic volumes compare to weekend traffic volumes on I-5*" triggers the system to aggregate the entire historical data into weekday and weekend categories, which would require processing extensive amounts of data without temporal constraints. As an outcome, no results have been returned from the system after processing time exceeds the

limit. 2) TP-GPT is not active to resolve specific locations. From inspections, the system performs inadequately in connecting geographic coordinates consisting of longitude and latitude to urban landmarks, or in searching a segment of freeway close to the specific exit. Basically, this problem occurs due to the lack of precise mapping information provided by GPT. 3) TP-GPT can only provide an approximate time of arrival based on the average real-time speed data. While this estimation can be useful, TP-GPT lacks the capabilities for detailed trip planning. Specifically, it does not support advanced route planning and optimization functions, which are essential for accurate and efficient navigation. This limitation means that while TP-GPT can offer a rough idea of travel time, it cannot adjust for variables such as traffic conditions, road closures, or alternative route options that might affect the overall trip duration. 4) The insufficient exploratory reasoning ability of TP-GPT is obtained when faced with oblique questions that necessitate multi-step causal inference. For instance, when asked about road closures on SR-520 in March 2024, the optimal approach would involve identifying dates when traffic volumes dropped to zero, indicating road closures. However, TP-GPT struggles to perform this level of reasoning. Instead of deducing the closures from traffic data, it typically responds that it cannot provide specific dates and times due to the unavailability of direct data on road closures in the database.

Overall, TP-GPT demonstrates a strong capacity to decompose and understand input questions by accurately identifying essential task elements such as route information, temporal details, and specific user demands. Its strengths are notably evident in several areas: it effectively recognizes specific route names and retrieves corresponding freeway

data, accurately handles date and time-related information, and demonstrates advanced inferential skills in interpreting questions and formulating database queries accordingly. However, TP-GPT also shows limitations in certain complex scenarios. In summary, while TP-GPT excels in query generation and traffic data interpretation, leading to robust performance in many traffic analysis and advisory contexts, it faces challenges with complex spatial reasoning and detailed navigational planning. These insights from the TransQuery benchmark highlight both the potential and the areas for future improvement in TP-GPT.

**Table 4.3 Exemplary answers generated by TP-GPT with ratings to the questions in TransQuery benchmark**

Question	TP-GPT Answer	Rating
<i>How do TPS scores vary among different freeways at 9am today?</i>	<p>The average TPS scores for different freeways at 9am today are as follows:</p> <ul style="list-style-type: none"> <li>- I-405: General Purpose lanes 0.6600, HOV lanes 0.7893</li> <li>- I-5: General Purpose lanes 0.6458, HOV lanes 0.7598</li> <li>- SR 520: General Purpose lanes 0.8615, HOV lanes 1.0000</li> <li>- I-90: General Purpose lanes 0.7695, HOV lanes 0.9495</li> <li>- SR 99: General Purpose lanes 0.7611, HOV lanes data not available</li> </ul>	Flawless
<i>What is the difference in Vehicle Miles of Travel (VMT) between weekdays and weekends in last week?</i>	<p>The average Vehicle Miles of Travel (VMT) for weekdays in the last week was approximately 12249.02, and for weekends, it was approximately 11024.08.</p>	Flawless
<i>What is the difference in TPS scores between weekdays and weekends in last month?</i>	<p>The average Traffic Performance Score (TPS) for weekdays last month was approximately 0.924 for General Purpose lanes and 0.954 for HOV lanes. For weekends, the average TPS was approximately 0.972 for General Purpose lanes and 0.981 for HOV lanes. This indicates that the traffic performance was slightly better on weekends compared to weekdays.</p>	Flawless
<i>What is average traffic volume in federal holidays 2024?</i>	<p>The average traffic volume during the holidays of New Year's Day, Independence Day, and Christmas Day in 2024 is 8 vehicles on each segment of highway.</p>	Flawless
<i>Comparison of TPS during peak hours</i>	<p>The average Traffic Performance Score (TPS) during peak hours</p>	Flawless

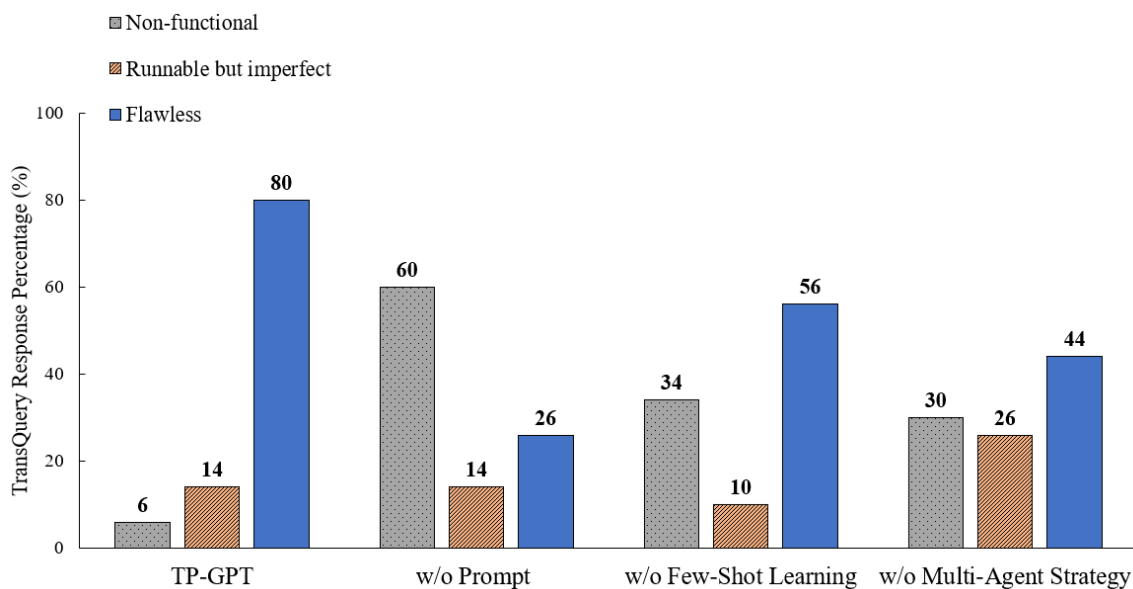
<i>versus non-peak hours?</i>	is approximately 0.9125, while during non-peak hours it is approximately 0.9653, suggesting better traffic conditions outside of peak hours.	
<i>Should I use HOV or general-purpose lane on I-5 now?</i>	Based on the most recent data, you should use the HOV lane on I-5 now as it has a higher average speed of 55.94 mph and a better traffic index of 0.817 compared to the general-purpose lane which has an average speed of 47.46 mph and a traffic index of 0.745.	Flawless
<i>What was the traffic condition of SR-520 during today's evening peak?</i>	The average traffic index for SR-520 during today's evening peak hours was approximately 0.812 for general purpose lanes and 0.989 for high-occupancy vehicle lanes, indicating better traffic flow in the HOV lanes.	Flawless
<i>What is occupancy on I-5 now?</i>	The average occupancy on I-5 right now is 10%.	Flawless
<i>How is Seattle's traffic condition now?</i>	The current average traffic conditions in Seattle are as follows: the average speed in general purpose (GP) lanes is 48.23 mph, the average volume in GP lanes is 19 vehicles, and the traffic index for GP lanes is 0.753. In high occupancy vehicle (HOV) lanes, the average speed is 56.03 mph, the average volume is 15 vehicles, and the traffic index is 0.8225. These values indicate the traffic performance in the last few minutes.	Flawless
<i>How was the traffic performance on I-405 Northbound at 9pm yesterday?</i>	The average traffic performance score (TPS) for the general-purpose lanes on I-405 Northbound at 9pm yesterday was approximately 0.974, and for the high occupancy vehicle (HOV)	Flawless

	lanes, it was approximately 0.988.	
<i>Which segment has the longest delay on I-5 in the last hour?</i>	The segment with the longest delay on I-5 in the last hour is segment 34, with a Traffic Performance Score (TPS) of approximately 0.071.	Flawless
<i>What was the highest volume on SR-520 yesterday?</i>	The highest volume on SR-520 yesterday was 124 vehicles per minute.	Flawless
<i>Which route has the longest traffic delay in the last hour?</i>	The route with the longest traffic delay in the last hour is I-5, with an average Traffic Performance Score (TPS) of approximately 0.5935.	Flawless
<i>How long is the average travel time for driving 5 miles on I-5 now?</i>	The average travel time for driving 5 miles on I-5 now is approximately 7.12 minutes.	Flawless
<i>What time should I travel to avoid congestion on I-5?</i>	To avoid congestion on I-5, you should consider traveling around 2 AM, as this is the time with the highest average Traffic Performance Score, indicating the least congestion.	Runnable but imperfect
<i>What is the difference in greenhouse gas emissions between weekdays and weekends in last month?</i>	The average vehicle miles traveled (VMT) for weekdays in the last month was approximately 12,180.70, while for weekends it was approximately 10,877.81. This suggests that there was a higher VMT on weekdays compared to weekends, which could imply higher greenhouse gas emissions on weekdays, assuming emissions are directly proportional to VMT.	Runnable but imperfect
<i>Tell me about traffic condition now near downtown Seattle.</i>	The current average traffic condition near downtown Seattle shows that the general-purpose lanes have an average speed of 50.305 mph with an average volume of 19 vehicles and a traffic	Runnable but imperfect

	index of approximately 0.774. The HOV lanes have an average speed of 58.37 mph with an average volume of 14.5 vehicles and a traffic index of approximately 0.861.	
<i>Estimate a rough travel time from downtown Seattle to downtown Bellevue now.</i>	The estimated rough travel time from downtown Seattle to downtown Bellevue is approximately 10 miles / 44.8875 mph × 60 minutes per hour = 13.4 minutes.	Runnable but imperfect
<i>When has SR-520 been closed in March 2024?</i>	I am unable to provide the specific dates and times when SR-520 was closed in March 2024 due to a lack of data from the executed query. If this information is critical, I recommend consulting alternative sources such as traffic reports or official transportation department announcements for the Greater Seattle area.	Non-functional
<i>How do weekday traffic volumes compare to weekend traffic volumes on I-5?</i>	N/A	Non-functional

## 4.4 ABLATION STUDY

This section examines the impact of removing the crafted prompt, few-shot learning, and multi-agent strategy from the proposed TP-GPT on TransQuery. Questions in the benchmark are input to TP-GPT and its three variations, and each response is evaluated as non-functional, runnable but imperfect, or flawless as described in Section 4.1. The percentage of responses in each of these three categories among all responses is calculated for the models, of which the results are shown in Figure 4.2. By comparing the flawless response rates, TP-GPT significantly outperforms the three variations, validating the effectiveness of incorporating each of modules embedded in TP-GPT.



**Figure 4.2 Ablation study of TP-GPT on TransQuery by removing prompts, few-shot learning, or multi-agent strategy**

- **Prompt:** The prompt elaborately customized for transportation-domain inquiry is essential to TP-GPT. The removal of prompt leads to a sudden drop in flawless

response rate from 80% to 26%, due to a lack of contextual knowledge and performing role description.

- **Few-shot learning:** Few-shot learning has the least impact on TP-GPT among modules regarding response metrics. This is because it only affects answering questions highly similar to the exemplar. For most questions not relevant to example inventory, the responses are barely impacted.
- **Multi-agent strategy:** Multi-agent collaboration plays a vital role in TP-GPT, suggested by the decrease in flawless response rate to 44% when it is removed. Intuitively, the problem-solving process has multiple stages, where reviewing outputs, generating feedback, and creating thoughts iteratively are fundamental to producing accurate results. However, a sequential process with no reflection loop in this variation frequently results in misunderstanding of contexts, causing syntax errors, or extracting mistaken data.

## CHAPTER 5: CONCLUSION AND FUTURE WORK

The integration of Large Language Models (LLMs) into the domain of traffic management presents a transformative approach to addressing the complexities and challenges inherent in modern transportation systems. This thesis has outlined the development of an intelligent traffic performance chatbot, TP-GPT, which leverages the power of real-time traffic data and the contextual understanding of LLMs to provide efficient, accurate, and privacy-preserving transportation surveillance and management.

TP-GPT demonstrates a novel framework that can effectively navigate the vast and intricate traffic database landscapes to extract meaningful insights. It has great capabilities to understand contextual information and respond to inquiries in the transportation domain by transforming input texts into queries and converting data into detailed natural-language analysis reports leveraging extensive prior knowledge. TP-GPT employs Chain of Thought prompting for iterative query generation, a multi-agent strategy to optimize intermediate results, few-shot learning to enhance exemplar performance, and chat memory to improve interaction quality. The experimental study devices a challenging traffic-analysis benchmark TransQuery to compare the performance of TP-GPT with state-of-the-art baseline LLMs. Quantitative results show that TP-GPT significantly outperforms GPT-4 Turbo, PaLM 2, and SQLCoder, demonstrating its superior performance in resolving real-world transportation inquiry tasks. The ablation study validates the effectiveness of employed modules in TP-GPT. Furthermore, an intelligent online ChatBot empowered by

TP-GPT is launched with an interactive web interface.

Future studies will improve TP-GPT to have the ability to predict future traffic conditions based on historical data, by possibly integrating traffic forecasting models. Besides, location recognition could be further enhanced leveraging the visual intelligence of language models in analyzing maps.

## BIBLIOGRAPHY

- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., . . . others. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Cai, T., Wang, X., Ma, T., Chen, X., & Zhou, D. (2023). Large language models as tool makers. *arXiv preprint arXiv:2305.17126*.
- Cai, Z., Li, X., Hui, B., Yang, M., Li, B., Li, B., . . . Si, L. (2022). Star: Sql guided pre-training for context-dependent text-to-sql parsing. *arXiv preprint arXiv:2210.11888*.
- Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., & Liu, Y. (2023). Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., . . . Wang, Y. (2023). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.
- Chen, J., Zheng, L., Hu, Y., Wang, W., Zhang, H., & Hu, X. (2024). Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Information Fusion, 104*, 102146.
- Cui, Z., Zhu, M., Wang, S., Wang, P., Zhou, Y., Cao, Q., . . . Wang, Y. (2020). *STAR Lab Traffic Performance Score Platform*. Retrieved from <http://tps.uwstarlab.org/>
- Cui, Z., Zhu, M., Wang, S., Wang, P., Zhou, Y., Cao, Q., . . . Wang, Y. (2020). Traffic Performance Score for Measuring the Impact of COVID-19 on Urban Mobility.

- arXiv preprint arXiv:2007.00648.*
- Da, L., Liou, K., Chen, T., Zhou, X., Luo, X., Yang, Y., & Wei, H. (2023). Open-TI: Open Traffic Intelligence with Augmented Language Model. *arXiv preprint arXiv:2401.00211.*
- de Zarzà, I., de Curtò, J., Roig, G., & Calafate, C. T. (2023). LLM Multimodal Traffic Accident Forecasting. *Sensors, 23.*
- Fraiwan, M., & Khasawneh, N. (2023). A Review of ChatGPT Applications in Education, Marketing, Software Engineering, and Healthcare: Benefits, Drawbacks, and Research Directions. *arXiv preprint arXiv:2305.00237.*
- Ge, Y., Hua, W., Mei, K., Tan, J., Xu, S., Li, Z., & Zhang, Y. (2024). Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems, 36.*
- Hong, Z., Yuan, Z., Chen, H., Zhang, Q., Huang, F., & Huang, X. (2024). Knowledge-to-SQL: Enhancing SQL Generation with Data Expert LLM. *arXiv preprint arXiv:2402.11517.*
- Jin, M., Shahriar, S., Tufano, M., Shi, X., Lu, S., Sundaresan, N., & Svyatkovskiy, A. (2023). Inferfix: End-to-end program repair with llms. *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, (pp. 1646–1656).
- Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., . . . Pan, S. (2023). Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728.*

- Kurzhanskiy, A. A., & Varaiya, P. (2010). Active traffic management on road networks: a macroscopic approach. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368, 4607–4626.
- Lai, S., Xu, Z., Zhang, W., Liu, H., & Xiong, H. (2023). Large language models as traffic signal control agents: Capacity and opportunity. *arXiv preprint arXiv:2312.16044*.
- Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., . . . Li, Y. (2023). Can LLM Already Serve as A Database Interface? A BIG Bench for Large-Scale Database Grounded Text-to-SQLs. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Ed.), *Advances in Neural Information Processing Systems*. 36, pp. 42330–42357. Curran Associates, Inc.
- Li, Z., & Xie, T. (2024). Using LLM to select the right SQL Query from candidates. *arXiv preprint arXiv:2401.02115*.
- Li, Z., Xia, L., Tang, J., Xu, Y., Shi, L., Xia, L., . . . Huang, C. (2024). UrbanGPT: Spatio-Temporal Large Language Models. *arXiv preprint arXiv:2403.00813*.
- Lo, C. K. (2023). What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. *Education Sciences*, 13.
- Ma, X., Yu, H., Wang, Y., & Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10, e0119044.
- Mahmassani, H. S. (2001, September). Dynamic Network Traffic Assignment and Simulation Methodology for Advanced System Management Applications. *Networks and Spatial Economics*, 1, 267–292.

- OpenAI. (2023). New models and developer products announced at DevDay. Retrieved from <https://openai.com/blog/new-models-and-developer-products-announced-at-devday>
- Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., . . . Chen, W. (2023). Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.
- Ping, W. J., Aw, W., & Srivastava, R. (2023). Open-sourcing SQLCoder2-15b and SQLCoder-7b. Retrieved from <https://defog.ai/blog/open-sourcing-sqlcoder2-7b/>
- Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Applied Sciences*, 13. doi:10.3390/app13095783
- Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., . . . Schneider, A. (2023). Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*.
- Ren, Y., Chen, Y., Liu, S., Wang, B., Yu, H., & Cui, Z. (2024). TPLLM: A Traffic Prediction Framework Based on Pretrained Large Language Models. *arXiv preprint arXiv:2403.02221*.
- Ross, S. I., Martinez, F., Houde, S., Muller, M., & Weisz, J. D. (2023). The programmer's assistant: Conversational interaction with a large language model for software development. *Proceedings of the 28th International Conference on Intelligent User Interfaces*, (pp. 491–514).

- Tang, Y., Wang, Z., Qu, A., Yan, Y., Hou, K., Zhuang, D., . . . Ma, W. (2024). Synergizing Spatial Optimization with Large Language Models for Open-Domain Urban Itinerary Planning. *arXiv preprint arXiv:2402.07204*.
- Topsakal, O., & Akinici, T. C. (2023). Creating large language model applications utilizing langchain: A primer on developing llm apps fast., *1*, pp. 1050–1056.
- Villarreal, M., Poudel, B., & Li, W. (2023). Can ChatGPT Enable ITS? The Case of Mixed Traffic Control via Reinforcement Learning. *arXiv preprint arXiv:2306.08094*.
- Wang, J., Cai, Z., Chen, Y., Yang, P., & Chen, B. (2023). An advanced control strategy for connected autonomous vehicles based on Micro simulation models at multiple intersections. *Physica A: Statistical Mechanics and its Applications*, *623*, 128836.
- Wang, L., Jiang, H., Cai, P., Fu, D., Wang, T., Cui, Z., . . . Wang, Y. (2023). AccidentGPT: Accident analysis and prevention from V2X Environmental Perception with Multi-modal Large Model. *arXiv preprint arXiv:2312.13156*.
- Wang, M., Pang, A., Kan, Y., Pun, M.-O., Chen, C. S., & Huang, B. (2024). LLM-Assisted Light: Leveraging Large Language Model Capabilities for Human-Mimetic Traffic Signal Control in Complex Urban Environments. *arXiv preprint arXiv:2403.08337*.
- Wang, P., Wei, X., Hu, F., & Han, W. (2024). TransGPT: Multi-modal Generative Pre-trained Transformer for Transportation. *arXiv preprint arXiv:2402.07233*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., . . . Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Ed.), *Advances*

- in Neural Information Processing Systems*. 35, pp. 24824–24837. Curran Associates, Inc.
- Yang, Q. I., & Koutsopoulos, H. N. (1996, June). A Microscopic Traffic Simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4, 113–129.
- Yang, Z., Jia, X., Li, H., & Yan, J. (2023). A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*.
- Yu, Y., Yang, C.-H. H., Kolehmainen, J., Shivakumar, P. G., Gu, Y., Ren, S. R., . . . Liu, Y.-C. (2023). Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 1–8). IEEE.
- Zhang, K., Wang, S., Jia, N., Zhao, L., Han, C., & Li, L. (2024). Integrating visual large language model and reasoning chain for driver behavior analysis and risk assessment. *Accident Analysis & Prevention*, 198, 107497.
- Zhang, K., Zhou, F., Wu, L., Xie, N., & He, Z. (2024). Semantic understanding and prompt engineering for large-scale traffic data imputation. *Information Fusion*, 102, 102038.
- Zhang, S., Fu, D., Liang, W., Zhang, Z., Yu, B., Cai, P., & Yao, B. (2024, May). TrafficGPT: Viewing, processing and interacting with traffic foundation models. *Transport Policy*, 150, 95–105.
- Zhou, T., Niu, P., wang, x., Sun, L., & Jin, R. (2023). One Fits All: Power General Time

- Series Analysis by Pretrained LM. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Ed.), *Advances in Neural Information Processing Systems*. 36, pp. 43322–43355. Curran Associates, Inc.
- Zhou, X., Li, G., & Liu, Z. (2023). Llm as dba. *arXiv preprint arXiv:2308.05481*.
- Zhou, X., Sun, Z., & Li, G. (2024). DB-GPT: Large Language Model Meets Database. *Data Science and Engineering*, 1–10.

## APPENDIX

### EXPERIMENTAL RESULTS OF TP-GPT ON TRANSQUERY BENCHMARK

This section holds the complete experimental results of TP-GPT. TransQuery benchmark involving 50 questions is used to test the traffic analytics performance of the system. Each pair of input questions and corresponding generated answers by TP-GPT are documented in Table 5.1. The shown results reveal details in TP-GPT generating answers to complex questions, which can provide an intuitive cognition of its performance.

**Table 5.1 Complete experimental results of TP-GPT on TransQuery benchmark**

Question	TP-GPT Answer
<i>How do TPS scores vary among different freeways at 9am today?</i>	<p>The average TPS scores for different freeways at 9am today are as follows:</p> <ul style="list-style-type: none"> <li>- I-405: General Purpose lanes 0.6600, HOV lanes 0.7893</li> <li>- I-5: General Purpose lanes 0.6458, HOV lanes 0.7598</li> <li>- SR 520: General Purpose lanes 0.8615, HOV lanes 1.0000</li> <li>- I-90: General Purpose lanes 0.7695, HOV lanes 0.9495</li> <li>- SR 99: General Purpose lanes 0.7611, HOV lanes data not available</li> </ul>

<i>What is the difference in Vehicle Miles of Travel (VMT) between weekdays and weekends in last week?</i>	The average Vehicle Miles of Travel (VMT) for weekdays in the last week was approximately 12249.02, and for weekends, it was approximately 11024.08.
<i>Estimate Vehicle Miles of Travel (VMT) on I-5 in the past hour.</i>	The estimated average Vehicle Miles of Travel (VMT) on I-5 in the past hour for the General Purpose (GP) lanes is approximately 2288.02 miles, and for the High Occupancy Vehicle (HOV) lanes, it is approximately 261.63 miles.
<i>Which day of week has the highest Vehicle Miles of Travel (VMT) in Seattle?</i>	The day of the week with the highest Vehicle Miles of Travel (VMT) in Seattle is Friday.
<i>What is the difference in TPS scores between weekdays and weekends in last month?</i>	The average Traffic Performance Score (TPS) for weekdays last month was approximately 0.924 for General Purpose lanes and 0.954 for HOV lanes. For weekends, the average TPS was approximately 0.972 for General Purpose lanes and 0.981 for HOV lanes. This indicates that the traffic performance was slightly better on weekends compared to weekdays.
<i>Compare traffic performance of I-5 between Monday and Sunday in last month.</i>	The average traffic performance score for I-5 in the last month was approximately 0.922 on Mondays and approximately 0.964 on Sundays, indicating better traffic conditions on Sundays.
<i>What is the difference in</i>	The average vehicle miles traveled (VMT) for weekdays in the last month was approximately

<i>greenhouse gas emissions between weekdays and weekends in last month?</i>	12,180.70, while for weekends it was approximately 10,877.81. This suggests that there was a higher VMT on weekdays compared to weekends, which could imply higher greenhouse gas emissions on weekdays, assuming emissions are directly proportional to VMT.
<i>How do greenhouse gas emissions correlate with TPS scores?</i>	The database does not contain information on greenhouse gas emissions, so I cannot provide a correlation between greenhouse gas emissions and TPS scores. Generally, higher traffic volumes and congestion can lead to increased greenhouse gas emissions due to idling and stop-and-go traffic. TPS scores, which represent traffic performance, may indirectly relate to emissions, with lower TPS scores potentially indicating higher emissions due to poor traffic conditions. However, without specific emissions data, this relationship cannot be quantified.
<i>How do greenhouse gas emissions correlate with Vehicle Miles of Travel (VMT)?</i>	Greenhouse gas emissions generally correlate positively with Vehicle Miles of Travel (VMT). As VMT increases, it typically indicates that more vehicles are on the road, which leads to higher fuel consumption and thus higher emissions of greenhouse gases. However, the exact correlation can vary depending on factors such as the types of vehicles (e.g., gasoline, diesel, electric), their fuel efficiency, traffic conditions, and driving behaviors. To accurately assess the correlation, one would need to analyze comprehensive data sets that include both VMT and emissions data, which is not available in the current database.
<i>What is the difference in rush hour traffic volume between different</i>	The average traffic volume during rush hour yesterday for each freeway was as follows: I-405 had 18, I-5 had 11, SR 520 had 15, I-90 had 19, and SR 99 had 17.

<i>freeways yesterday?</i>	
<i>How do weekday traffic volumes compare to weekend traffic volumes on I-5?</i>	N/A
<i>What is average traffic volume in federal holidays 2024?</i>	The average traffic volume during the holidays of New Year's Day, Independence Day, and Christmas Day in 2024 is 8 vehicles.
<i>Comparison of TPS during peak hours versus non-peak hours?</i>	The average Traffic Performance Score (TPS) during peak hours is approximately 0.9125, while during non-peak hours it is approximately 0.9653, suggesting better traffic conditions outside of peak hours.
<i>How does the morning rush hour TPS compare to the evening rush hour TPS on major freeways yesterday?</i>	The average Traffic Performance Score (TPS) for the morning rush hour on major freeways yesterday was approximately 0.842 for general purpose lanes and 0.922 for HOV lanes. For the evening rush hour, the average TPS was approximately 0.775 for general purpose lanes and 0.905 for HOV lanes. This indicates that the traffic performance was slightly better during the morning rush hour compared to the evening rush hour.
<i>Should I use HOV or general purpose lane on I-5 now?</i>	Based on the most recent data, you should use the HOV lane on I-5 now as it has a higher average speed of 55.94 mph and a better traffic index of 0.817 compared to the general purpose lane which has an average speed of 47.46 mph and a traffic index of 0.745.
<i>What time should I travel to avoid</i>	To avoid congestion on I-5, you should consider traveling around 2 AM, as this is the time

<i>congestion on I-5?</i>	with the highest average Traffic Performance Score, indicating the least congestion.
<i>Does HOV provide more efficiency on I-5?</i>	On I-5, the average Traffic Performance Score (TPS) for general purpose lanes (GP) is approximately 0.927, and for high-occupancy vehicle lanes (HOV) it is approximately 0.972. This indicates that HOV lanes provide more efficiency compared to general purpose lanes on I-5.
<i>Show me traffic statistics for I-90 during this morning's peak hours.</i>	The average traffic statistics for I-90 during this morning's peak hours are: Volume - 17 vehicles, Occupancy - 6%, Speed - 57 mph.
<i>What was the traffic condition of SR-520 during today's evening peak?</i>	The average traffic index for SR-520 during today's evening peak hours was approximately 0.812 for general purpose lanes and 0.989 for high-occupancy vehicle lanes, indicating better traffic flow in the HOV lanes.
<i>What is traffic speed on SR-99 now?</i>	The current average traffic speed on SR-99 for the general-purpose lanes is approximately 58.26 mph. There is no data available for the high-occupancy vehicle lanes.
<i>What is occupancy on I-5 now?</i>	The average occupancy on I-5 right now is 10%.
<i>How many cars are on each segment of I-405 on average now?</i>	The average number of cars on each segment of I-405 in the most recent data are as follows: Segment 84 has an average of 23 cars, Segment 83 has an average of 22 cars, Segment 82 has an average of 22 cars, and so on for the other segments.
<i>How is Seattle's traffic condition now?</i>	The current average traffic conditions in Seattle are as follows: the average speed in general purpose (GP) lanes is 48.23 mph, the average volume in GP lanes is 19 vehicles, and the

---

traffic index for GP lanes is 0.753. In high occupancy vehicle (HOV) lanes, the average speed is 56.03 mph, the average volume is 15 vehicles, and the traffic index is 0.8225. These values indicate the traffic performance in the last few minutes.

---

*Show me the most recent loop data on SR 99 Northbound sorted by locations.*

The most recent loop data on SR 99 Northbound would provide the average volume, occupancy, and speed for each location, sorted by the milepost in descending order. Unfortunately, I am unable to present the actual data without the query results.

---

*Query the most recent loop data on I-405 Southbound sorted by locations.*

The most recent average traffic volume, occupancy, and speed on I-405 Southbound at various locations are as follows:

- Location: 8.40000, Average Volume: 29, Average Occupancy: 17, Average Speed: 42
- Location: 8.03000, Average Volume: 24, Average Occupancy: 21, Average Speed: 36
- Location: 25.17000, Average Volume: 30, Average Occupancy: 8, Average Speed: 63
- Location: 7.00000, Average Volume: 39, Average Occupancy: 19, Average Speed: 40
- Location: 17.99000, Average Volume: 21, Average Occupancy: 6, Average Speed: 63
- (Additional results truncated for brevity)

Please note that the milepost values are represented as decimal numbers, and the average values are rounded to the nearest whole number.

---

*Query the sorted loop data by locations on SR 520 Eastbound at*

The average traffic volume, occupancy, and speed for various locations on SR 520 Eastbound at noon today are as follows (sorted by location in descending order):

---

---

*noon today.*

- West Bridge: Volume 10, Occupancy 3, Speed 49
  - WB Redmond Wy-WB: Volume 9, Occupancy 2, Speed 64
  - W Lk Samm-WB: Volume 11, Occupancy 3, Speed 58
  - W Lk Samm-EB: Volume 10, Occupancy 2, Speed 66
  - NE 76th St: Volume 6, Occupancy 2, Speed 61
  - NE 60th St: Volume 17, Occupancy 5, Speed 52
  - NE 51st St-EB: Volume 15, Occupancy 2, Speed 65
  - NE 40th St-EB: Volume 11, Occupancy 4, Speed 63
  - NE 36th St: Volume 17, Occupancy 4, Speed 64
  - Montlake Blvd-EB: Volume 10, Occupancy 6, Speed 55
  - Mid Bridge: Volume 16, Occupancy 5, Speed 62
  - I-405-EB SR 520: Volume 13, Occupancy 4, Speed 62
  - I-405 I/C: Volume 10, Occupancy 4, Speed 62
  - Evergreen Pt Rd: Volume 6, Occupancy 3, Speed 42
  - East Bridge: Volume 11, Occupancy 3, Speed 68
  - Bellevue Way-WB: Volume 10, Occupancy 2, Speed 58
  - 98th Ave NE: Volume 9, Occupancy 2, Speed 59
  - 92nd Ave NE-EB: Volume 7, Occupancy 2, Speed 57
-

- 
- 91st Ave NE: Volume 10, Occupancy 3, Speed 61
  - 88th Ave NE: Volume 13, Occupancy 3, Speed 63
  - 85th Ave NE: Volume 13, Occupancy 3, Speed 63
  - 84th Ave NE-WB: Volume 10, Occupancy 2, Speed 66
  - 20th Ave E: Volume 11, Occupancy 4, Speed 65
  - 148th Ave NE-EB: Volume 13, Occupancy 4, Speed 62
  - 140th Ave NE: Volume 21, Occupancy 6, Speed 63
  - 130th Ave NE: Volume 17, Occupancy 5, Speed 37
  - 124th Ave NE-WB: Volume 14, Occupancy 5, Speed 59
  - 116th Ave NE: Volume 23, Occupancy 11, Speed 43
  - 10th Ave E: Volume 25, Occupancy 10, Speed 55
  - 108th Ave NE-WB: Volume 1, Occupancy 1, Speed 65
  - 108th Ave NE-EB: Volume 11, Occupancy 3, Speed 57

Please note that these are average values for the specified time and location.

---

<i>How was the traffic performance on I-405 Northbound at 9pm yesterday?</i>	The average traffic performance score (TPS) for the general purpose lanes on I-405 Northbound at 9pm yesterday was approximately 0.974, and for the high occupancy vehicle (HOV) lanes, it was approximately 0.988.
<i>Show me the most recent 10 loop</i>	The query to retrieve the most recent 10 loop data at I-5 Southbound milepost 140.15 has

---

<i>data at I-5 Southbound milepost 140.15.</i>	<p>been executed, but the results are not available due to a missing observation. The query used was:</p> <pre> ```sql SELECT TOP(10) * FROM MinuteDataNW MDNW INNER JOIN cabinets CAB ON MDNW.id = CAB.UnitName WHERE CAB.Route = '5' AND CAB.Milepost = 140.15 AND CAB.direction = 'S' ORDER BY MDNW.time_stamp DESC ``` </pre>
<i>Which segment has the longest delay on I-5 in the last hour?</i>	<p>The segment with the longest delay on I-5 in the last hour is segment 34, with a Traffic Performance Score (TPS) of approximately 0.071.</p>
<i>What is the highest volume on SR-520 yesterday?</i>	<p>The highest volume on SR-520 yesterday was 124 vehicles.</p>
<i>Which segment has the fastest travel time on I-405 now?</i>	<p>The segment with the fastest travel time on I-405 right now is segment 60 with an average speed of 65.0 mph.</p>
<i>What is the lowest occupancy on SR-520 last Sunday?</i>	<p>The lowest occupancy on SR-520 last Sunday was 0%.</p>
<i>Should I choose I-5 or SR-520 to</i>	<p>Based on the current traffic performance scores, SR-520 has a better traffic index than I-5.</p>

---

<i>drive on now?</i>	You should choose SR-520 to drive on now.
<i>Which route has the longest traffic delay in the last hour?</i>	The route with the longest traffic delay in the last hour is I-5, with an average Traffic Performance Score (TPS) of approximately 0.5935.
<i>Which segment of I-90 experiences the highest and lowest traffic speed today?</i>	<p>The query I constructed is intended to identify the segments of I-90 that experience the highest and lowest traffic speeds today by querying the `MinuteDataNW` and `cabinets` tables. Due to the lack of actual query results, I cannot specify which segments these are. However, the query is designed to return the segment with the highest speed first, followed by the segment with the lowest speed, both for today's date.</p>
<i>Can you infer any incidents that have happened on I-5 during yesterday's evening peak hours?</i>	<p>Based on the queried data, here are some observations from I-5 during yesterday's evening peak hours:</p> <ol style="list-style-type: none"> <li>1. At 18:00, a loop detector with ID '005es21029:_MS_Stn' recorded a volume of 25 vehicles, occupancy of 2%, and speed of 65 mph.</li> <li>2. Another loop detector with ID '005es21029:_MS_O_1' recorded a volume of 3 vehicles, occupancy of 2%, and speed of 65 mph at the same time.</li> <li>3. Additional loop detectors recorded similar speeds with varying volumes and occupancies.</li> </ol> <p>Without a baseline for comparison, these observations alone cannot confirm incidents.</p>

---

---

	<p>However, the speeds seem relatively normal and there are no significant drops in speed or spikes in occupancy that would typically indicate an incident. More detailed analysis would be required to infer incidents, potentially including a comparison to historical data or further investigation into any outliers.</p>
<p><i>Tell me about traffic condition now near downtown Seattle.</i></p>	<p>The current average traffic condition near downtown Seattle shows that the general purpose lanes have an average speed of 50.305 mph with an average volume of 19 vehicles and a traffic index of approximately 0.774. The HOV lanes have an average speed of 58.37 mph with an average volume of 14.5 vehicles and a traffic index of approximately 0.861.</p>
<p><i>Estimate a rough travel time from downtown Seattle to downtown Bellevue now.</i></p>	<p>The estimated rough travel time from downtown Seattle to downtown Bellevue is approximately <math>10 \text{ miles} / 44.8875 \text{ mph} \times 60 \text{ minutes per hour} = 13.4 \text{ minutes}</math>.</p>
<p><i>How long is the average travel time for driving 5 miles on I-5 now?</i></p>	<p>The average travel time for driving 5 miles on I-5 now is approximately 7.12 minutes.</p>
<p><i>When has SR-520 been closed in March 2024?</i></p>	<p>I am unable to provide the specific dates and times when SR-520 was closed in March 2024 due to a lack of data from the executed query. If this information is critical, I recommend consulting alternative sources such as traffic reports or official transportation department announcements for the Greater Seattle area.</p>
<p><i>Which segment of I-5 has the most</i></p>	<p>The segment of I-5 that had the most vehicles exiting the highway at 6pm yesterday is</p>

---

<i>vehicles exiting highway at 6pm yesterday?</i>	segmentID 29 with a total volume of 23 vehicles.
<i>Introduce yourself. What information can you provide and what tasks are you able to solve?</i>	Hello! I am an intelligent traffic assistant specialized in generating syntactically correct MSSQL queries to provide traffic-related information. I can answer questions about traffic statistics, such as volume, occupancy, and speed, for specific highways in the Greater Seattle area during certain times. I can also provide information on traffic performance scores for road segments. My expertise includes querying real-time loop detector data from a connected database that contains tables with traffic data, loop detector information, road segment locations, and traffic indices. I can perform complex queries to calculate average values, but I will not list entire records or make any changes to the database. How can I assist you with traffic data today?
<i>What data do you have in the database? How many tables are in the database?</i>	The database contains 13 tables, but I am only permitted to query 6 of them: MinuteDataNW, cabinets, cabinfo, Segments, SegmentTrafficIndex, and TrafficIndex. These tables store information about traffic volume, occupancy, speed, loop detector locations, road segments, and traffic index for the Greater Seattle area highway system.
<i>Give a brief analysis report on Seattle's traffic performance yesterday.</i>	The average traffic performance in Seattle for yesterday was as follows: - Average Speed in General Purpose (GP) lanes: 59.95 mph - Average Volume in GP lanes: 13.30 vehicles

- 
- Average Traffic Index in GP lanes: 0.914 (out of 1)
  - Average Speed in High Occupancy Vehicle (HOV) lanes: 62.91 mph
  - Average Volume in HOV lanes: 7.79 vehicles
  - Average Traffic Index in HOV lanes: 0.949 (out of 1)

These values indicate that the traffic performance was relatively good, with higher speeds and traffic index scores closer to 1, which represents better traffic performance.

---

*Is traffic condition in December 2023 worse than that in March 2024 in Seattle?*

The average Traffic Performance Score (TPS) for general purpose lanes in December 2023 was 0.943 and for high occupancy vehicle (HOV) lanes was 0.963. In March 2024, the average TPS for general purpose lanes was 0.939 and for HOV lanes was 0.963. The traffic condition in December 2023 was slightly better than in March 2024 based on the average TPS.

---

*On which highways can you provide traffic data?*

We can provide traffic data for the following highways: I-405, I-5, SR 520, I-90, and SR 99.

---

*For what time period can you provide traffic data?*

I can provide traffic data for the time period from November 1, 2019, to April 25, 2024.

---

*Explain what is traffic performance score.*

The traffic performance score (TPS) is a metric used to evaluate the quality of traffic flow on roadways. It is typically a number between 0 and 1, where a higher score indicates better

---

---

traffic performance. This score can take into account various factors such as traffic volume, speed, and roadway occupancy. In the context of the database we are discussing, the TPS likely reflects the real-time conditions of the traffic as measured by loop detectors and other sensors on the highways in the Greater Seattle area. A TPS close to 1 would suggest free-flowing traffic with little to no delays, while a score closer to 0 would indicate poor traffic conditions, such as congestion or stop-and-go traffic.

---

*Give me some advise on trip  
planning for driving on I-5  
tomorrow.*

For trip planning on I-5 tomorrow, it's advisable to avoid typical peak traffic hours, which are usually during morning (6:00 AM to 9:00 AM) and evening (4:00 PM to 7:00 PM) rush hours. If possible, plan your drive outside these hours to avoid congestion. Additionally, checking real-time traffic updates before and during your trip can help you avoid unexpected delays. Remember that traffic conditions can vary due to many factors, including weather, accidents, and roadworks, so always stay informed with the latest traffic reports.

---

*Is there any segment of I-5 having  
free flow speed right now?*

Yes, there is a segment of I-5 (segmentID 40) that is currently experiencing free flow speed with an average speed of 65.0 mph and a perfect traffic index score of 1.0, indicating no traffic congestion.

---

---