Transportation Research Interdisciplinary Perspectives 38 (2026) 102124

Contents lists available at ScienceDirect

Transportation Research Interdisciplinary Perspectives

journal homepage: www.elsevier.com/locate/trip

Large language models for travel behavior prediction

Baichuan Mo [a], Hanyong Xu [c,*], Ruoyun Ma [d], Jung-Hoon Cho [b], Dingyi Zhuang [b], Xiaotong Guo [b], Jinhua Zhao [c]

[a] Department of Civil Engineering, Tsinghua University, Beijing, 100084, China
[b] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
[c] Department of Urban Studies and Planning, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
[d] Department of Management Science and Engineering, Stanford University, Stanford, CA 94305, United States of America

ARTICLE INFO

Keywords:
Large language models
Travel behavior prediction

ABSTRACT

This study evaluates large language models (LLMs) for travel behavior prediction under different levels of labeled-data availability. We compare three LLM-based frameworks: zero-shot direct prompting, textual-gradient prompt optimization from a small labeled budget, and supervised prediction using LLM text embeddings. These methods are benchmarked against multinomial logit, random forests, neural networks, and TabPFN under a budget-matched protocol on Swissmetro mode choice, London mode choice, and NHTS trip-purpose prediction. The results show a clear data-availability pattern. In scarce-label settings, direct LLM prediction is competitive with, and sometimes significantly better than, supervised/tabular baselines. Textual-gradient optimization can learn prompts that match expert hand-crafted prompts without manually encoded numerical cues, although its gains are task-dependent. As labeled budgets grow, conventional supervised and tabular models become stronger. Diagnostic tests further suggest that LLM predictions respond to supplied travel-time and travel-cost structure rather than simply memorizing benchmark records, while generated explanations should be treated as auditable but imperfect rationales.

1. Introduction

Understanding human travel behavior has always been a crucial part of transportation planning. Travel behavior is a broad concept that consists of many attributes, such as the mode of transportation, the purpose of the travel, the choice of destination, and the time of departure. Consider travel mode choice as an example, which refers to how people make decisions on their means of travel. In the task of predicting mode choices, making accurate predictions is useful for both agencies and individuals. For agencies, it facilitates providing better transit services and forecasting travel conditions in the short term, while informing transportation infrastructure investments, better policies, and land use planning in the long run. In the traditional Four-Step Model for future demand and performance analysis, the mode share of a given population is estimated using mode split models (Ben-Akiva and Lerman, 1985). For individuals, on the other hand, mode choice prediction embedded in map applications helps them to make sound travel decisions. Conventional mode choice analysis relies on using numerical data to construct a mathematical model to represent human preferences (Mo et al., 2018, 2021). One example is the discrete choice model, where mode choice probabilities are calculated assuming people make choices to maximize their utilities (Ben-Akiva and Lerman, 1985). More recently, machine learning methods such as random forests, support vector machines, and neural networks have been increasingly applied to mode choice prediction, often matching or exceeding discrete choice models in predictive accuracy while offering complementary behavioral insights (Kamkar et al., 2025; Srisurin et al., 2026; Ali and Fissha, 2026; Abulibdeh, 2023; Wang et al., 2024c). This line of work spans diverse contexts, from intercity and tourism travel to school and work commuting (Arreeras et al., 2025; Srisurin et al., 2026).

Recently, large language models (LLMs) have generated substantial research interest because of their ability to interpret instructions, integrate heterogeneous information, and produce natural-language reasoning. Trained on large text corpora and scaled to billions of parameters, LLMs can execute diverse tasks with limited task-specific supervision (Zhao et al., 2023). In transportation, recent studies have shown that GPT-style models can support mobility prediction and, in some sequential forecasting settings, outperform time-series machine learning models (Wang et al., 2023b). However, whether LLMs can support individual travel behavior prediction, and under what data conditions they are useful, remains unclear.

* Corresponding author.
E-mail address: hanyongx@mit.edu (H. Xu).

https://doi.org/10.1016/j.trip.2026.102124
Received 22 April 2026; Received in revised form 21 June 2026; Accepted 22 June 2026

In response to the above inquiry, this study investigates the capability of large language models (LLMs) to predict individual travel behavior, with two mode-choice datasets and one trip-purpose dataset serving as parallel case-study tasks. A central theme of our analysis is the *data-availability spectrum*: methods differ not only in accuracy but in how many task-specific labeled examples they require. We propose and evaluate three LLM-based prediction frameworks positioned along this spectrum. The first framework adopts a zero-shot prompting approach, in which prompts describe the travel behavior task, individual travel characteristics, and traveler attributes. The second framework, which is the methodological focus of this work, introduces a *textual-gradient automatic prompt optimization* approach that removes the need for hand-engineered reasoning cues: starting from a minimal prompt, an LLM repeatedly inspects its own prediction errors on a small labeled set, writes a natural-language critique of the prompt, and edits the prompt to reduce those errors. The third framework leverages LLM-generated text embeddings as high-level feature representations, which are subsequently used as inputs to supervised learning models for prediction under small-sample settings. To assess these frameworks rigorously, we adopt a *budget-matched* evaluation protocol that compares them against classical travel behavior models (multinomial logit, random forest, and neural networks) and a state-of-the-art in-context tabular foundation model (TabPFN). The results are organized as a sequence of questions: prompt generation, prompt transferability, model performance under matched budgets, and direct prompting versus hidden-feature prediction. This organization treats Swissmetro, London, and NHTS as equally important prediction tasks while making the empirical storyline explicit.

This paper makes the following contributions:

- We present a unified study of LLMs for travel behavior prediction that spans the data-availability spectrum, covering zero-shot prompting, textual-gradient prompt optimization, and LLM-embedding-based supervised learning, evaluated on Swissmetro mode choice, London mode choice, and NHTS trip-purpose prediction.
- We introduce a textual-gradient automatic prompt optimization method that *learns* the prediction prompt from a small labeled budget, replacing the hand-crafted reasoning cues used in prior LLM prompting. This directly addresses concerns of manual prompt engineering, domain-knowledge leakage, and prompt sensitivity, and yields interpretable, transferable prompts.
- We propose a budget-matched evaluation protocol and benchmark the LLM-based frameworks against classical models and a strong in-context tabular foundation model (TabPFN) at identical labeled-data budgets across three datasets.
- Through extensive experiments with multiple proprietary and open-source LLMs, we show that learned prompts can match or exceed expert hand-crafted prompts without manual reasoning cues, and that LLM-based methods are most competitive in the extreme low-data regime before supervised and tabular models overtake them at larger labeled budgets.

The remainder of this paper is organized as follows. The literature review is presented in Section 2. Section 3 describes the LLM-based prediction frameworks. Section 4 presents the case studies and empirical results. Section 5 concludes the paper and discusses future research directions.

2. Related work

2.1. Recent developments in large language models

Large language models (LLMs) typically refer to Transformer-based language models with very large numbers of parameters, which have been shown to excel in many complex tasks (Zhao et al., 2023). Wei et al. (2022a) demonstrates that some emergent abilities appear only in larger models and cannot be predicted from smaller models' performance. Newer models such as GPT-3.5, GPT-4 (OpenAI, 2024a), Gemini (Team, 2024), and Llama (Touvron et al., 2023a,b; Teams, 2024) demonstrate strong capabilities in language processing, quantitative reasoning, planning, and learning (Bubeck et al., 2023). Moreover, Gurnee and Tegmark (2023) show that LLMs can encode structured knowledge about space and time in their activations. Products such as ChatGPT have made LLM interfaces widely accessible, further accelerating research and public interest in this field (Zhao et al., 2023).

Beyond their performance on complex tasks, LLMs have introduced a new application-development paradigm based on prompt interfaces and APIs rather than training task-specific models from scratch (Zhao et al., 2023). Their performance often depends on prompt design. Common methods for improving LLM task performance include in-context learning, which expresses tasks and demonstrations in natural language (Brown et al., 2020; Dong et al., 2022); chain-of-thought prompting, which elicits intermediate reasoning (Chu et al., 2024); and plan-and-solve prompting, which decomposes complex tasks into steps (Wang et al., 2023a; Zhou et al., 2023).

LLMs also have limitations that must be considered when using them for prediction (Zhao et al., 2023). One limitation is hallucination: generated text may be internally inconsistent or unverifiable (Bang et al., 2023; Huang et al., 2025). Another challenge is limited recency, since large pretrained models are not updated in real time (Zhao et al., 2023). A third limitation is inconsistent reasoning, where the model's stated rationale and final answer are not always aligned (Wei et al., 2022b).

2.2. Large language model as a predictor in human mobility

Because LLMs perform well on many language-centered tasks, researchers have begun using them to support classification, reasoning, and decision-support workflows across fields. ChatGPT has been shown to achieve high accuracy on language-related classification tasks, such as genre recognition, personality prediction, political opinion inference, and hateful-speech detection (Liu et al., 2023; Kuzman et al., 2023; Amin et al., 2023; Zhang et al., 2023; Huang et al., 2023). Similarly, the use of LLMs in transportation has grown rapidly in recent years, including applications in traffic forecasting, human mobility prediction, demand prediction, and data imputation (Zhang et al., 2024).

In human mobility, a series of experiments have demonstrated the semantic and reasoning capabilities of LLMs. Before GPT-style models became widely available, several studies used pre-trained language models to predict mobility. Xue et al. (2022), Xue and Salim (2022) convert mobility data into language descriptions to fine-tune language models, while Kobayashi et al. (2023), Gong et al. (2024), and Wu et al. (2024) examine encodings and embeddings for mobility prediction in Transformer architectures. With commercial LLMs, Luo et al. (2024) show that GPT-style models can infer trajectory patterns from data, while Wang et al. (2023b) demonstrate that GPT-style models can infer next locations from semantic sequential-mobility prompts converted from historical locations. In their setting, LLM predictions outperform traditional time-series models such as LSTM and Multi-Head Self-Attentional (MHSA) neural networks. Additional studies show that trajectory or time-series prediction can be improved by incorporating behavioral theories (Shao et al., 2024), agentic logical thinking (Wang et al., 2024a; Li et al., 2024b,a; Feng et al., 2025), pattern seeking (Qin et al., 2025; Wang et al., 2024a; Li et al., 2024b), and additional semantic data (Liang et al., 2023).

LLMs have also been applied to trip planning, one downstream application of transportation prediction. Prior work shows that performance can improve by leveraging multiple information sources and queries (Xie et al., 2024; Fang et al., 2024; Singh et al., 2024) and by decomposing planning tasks into multiple steps (Tang et al., 2024; Xie and Zou, 2024).

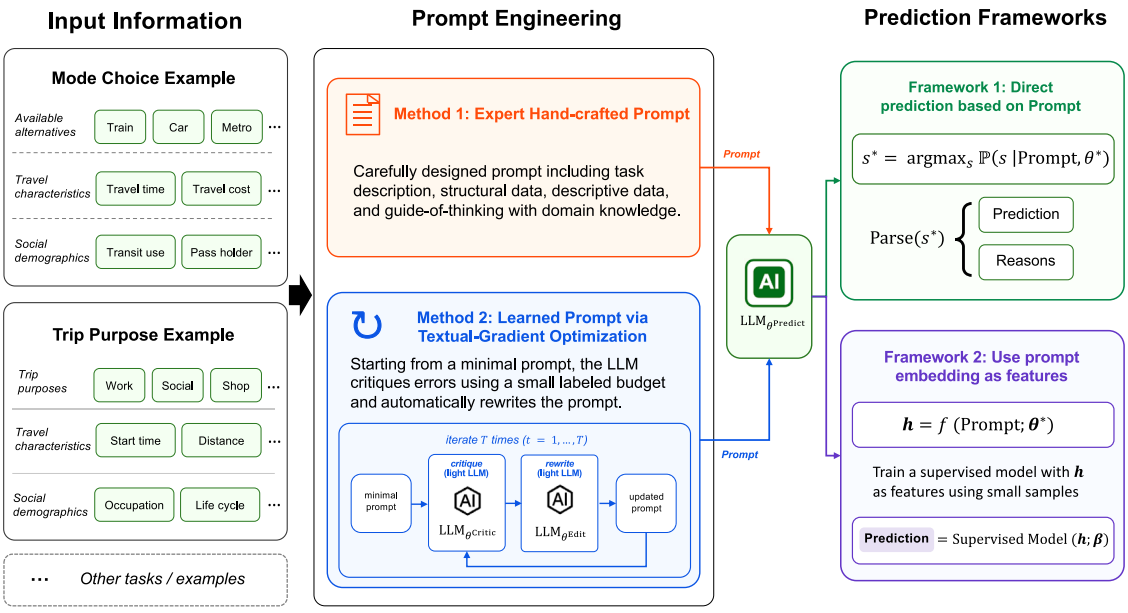

Fig. 1. Conceptual framework

Although existing research shows promise for using LLMs as predictors, most studies focus on sequential mobility prediction, and relatively little work has examined LLMs for travel behavior prediction tasks. For travel choice prediction, a few studies analyze how historical personalized information can improve prediction (Zhai et al., 2024; Wang et al., 2024b; Chen et al., 2024). Liu et al. (2024) introduce persona loading and few-shot examples to reduce the misalignment between LLM predictions and human behavior. However, there has not been a systematic study of zero-shot and scarce-label LLM prediction for mode choice and trip-purpose inference, which are the focus of this study.

3. Methodology

3.1. LLM prediction and embedding primitives

Large language models (LLMs) are neural network-based architectures designed to process and generate human language text. Recent LLMs are usually built upon the transformer architecture. This architecture employs a deep neural network with self-attention mechanisms, allowing for the modeling of long-range dependencies in sequences of text. The size and scale of LLMs have grown exponentially, with models containing hundreds of millions or even billions of parameters. The increase in model size directly correlates with improved performance on various NLP tasks.

Pre-training is a crucial phase in the development of LLMs, where the model is exposed to massive amounts of text data to learn the statistical properties of language, including grammar, semantics, and world knowledge. The primary objective of pre-training is to train the model to predict the next word in a sentence or sequence of words. This task, known as language modeling, allows the model to capture the statistical regularities and contextual dependencies within the training data. Note that different LLMs (e.g., BERT and GPT) may have different pre-training tasks. In this study, we introduce the pre-training based on GPT's framework for unsupervised multitask learning (Radford et al., 2019). The objective of pre-training can be expressed as:

$$\theta^* = \arg\max_{\theta} \sum_{u \in \mathcal{U}} \sum_{i=1}^{n_u} \log \mathbb{P}\left(w_i^{(u)} \mid w_1^{(u)}, w_2^{(u)}, \ldots, w_{i-1}^{(u)}; \theta\right) \tag{1}$$

where $\mathcal{U}$ is the training corpus, u indexes a text sequence in the corpus, n_u is the length of sequence u, and $w_i^{(u)}$ is the ith token of that sequence.

Given the trained parameter θ^*, we can use the model to generate answers for various tasks:

$$s^* = \arg\max_{s} \mathbb{P}(s \mid (\text{Input, Task}); \theta^*) \tag{2}$$

where s^* is the output sequence with the largest probability (or relatively large probability depending on the searching algorithm and degree of randomness). s^* is generated word by word until the "[End]" token is found. For example, asking the LLM to solve an addition problem can be expressed as

$$\text{"The answer is 28 [End]"} = \arg\max_{s} \mathbb{P}(s \mid \text{"What is 3+25? [End]"}; \theta^*) \tag{3}$$

In the real-world implementation, the "[End]" token will be automatically added to the input sequence and will not be displayed at the end of the output sequence.

Besides generating text sequences, LLMs can also be used to obtain text embeddings from the internal layers before the final text-output layer. Specifically, denote the embedding model as $f(\cdot; \theta^*)$. Then the text embedding (denoted as $\boldsymbol{h}$) for the input task can be obtained as:

$$\boldsymbol{h} = f\left((\text{Input, Task}); \theta^*\right) \tag{4}$$

$\boldsymbol{h} \in \mathbb{R}^H$ is usually used as a feature representation for downstream supervised learning tasks. Its dimension H depends on the specific LLM architecture. Because LLMs are pretrained on large and heterogeneous corpora, $\boldsymbol{h}$ may improve prediction performance when only small training samples are available.

3.2. Conceptual framework

Because LLMs provide a generalized multitask solver, they can be used as predictors for travel behavior. The central organizing principle of this paper is *data availability*: different transportation agencies and research settings have different amounts of labeled local data. We therefore consider three LLM-based prediction modes along a supervision spectrum. At one end, a zero-shot direct-prediction mode uses no task-specific labels and relies on a hand-crafted prompt. With a small labeled budget, a textual-gradient optimizer learns the prompt automatically. Finally, when a supervised training set is available, LLM embeddings can be used as high-level features in conventional predictive models.

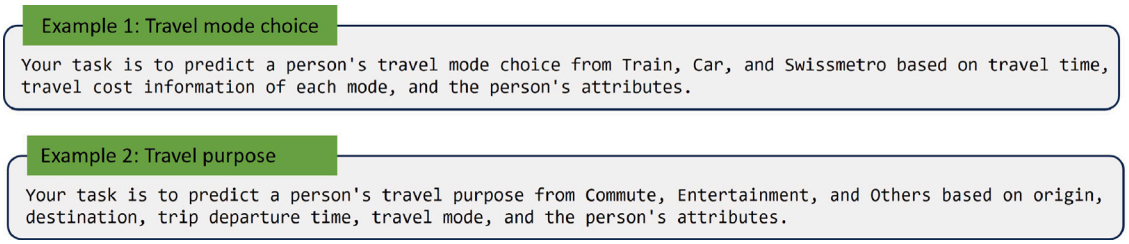

Fig. 2. Example prompts for task descriptions.

The "(Input, Task)" in Eq. (2) is referred to as a **prompt**. In this study, prompts are constructed in two ways. The hand-crafted prompt includes (1) task descriptions, (2) structural data, (3) descriptive data, and (4) reasoning guidance with domain knowledge (Section 3.3). Because this design requires expert effort and can embed manual reasoning cues, we also introduce an automatic alternative that *learns* the prompt from a small labeled budget (Section 3.4).

The overall framework of LLM-based travel behavior prediction is shown in Fig. 1, using a travel mode choice task and a trip purpose prediction task as illustration examples. The input information (available alternatives, travel characteristics, and social demographics) is first organized and embedded into a prompt. As shown in the middle of the figure, the prompt can be obtained by two prompt-engineering methods: (i) an expert *hand-crafted* prompt that encodes domain knowledge and reasoning guidance (Section 3.3), or (ii) an automatically *learned* prompt produced by textual-gradient optimization from a small labeled budget (Section 3.4). Both methods yield a prompt that is fed into the same LLM. Depending on how the LLM output is used, we obtain two prediction frameworks: Framework 1 directly generates the predicted behavior together with its supporting reasons (no supervised training step is needed), whereas Framework 2 uses the LLM's prompt embedding as a feature vector for a downstream supervised model (Section 3.5).

After processing by the LLM, the output is used in one of two ways. For direct prediction, a text-generation LLM returns both the predicted behavior and a natural-language reason. For embedding-based prediction, a text-embedding model (OpenAI, 2024b) maps each sample n to a hidden vector $\boldsymbol{h}_n$; a supervised model is then trained on these vectors and used to predict the travel behavior label.

3.3. Zero-shot direct prediction with hand-crafted prompts

Building upon existing prompting strategies, we carefully develop context-inclusive prompts that incorporate relevant contextual information to enhance travel behavior prediction. Detailed components are described as follows.

3.3.1. Task description

The task in this study is travel behavior prediction, which may include travel mode choice, trip purpose, departure time, duration, destination, and related outcomes. In the task description, we specify the available options and the input information provided to the model. Example prompts are shown in Fig. 2.

3.3.2. Structural data

People's travel behavior is strongly influenced by travel characteristics, such as origin, destination, travel time, and travel cost. The prompts should include this information in a concise and organized way. In this study, we use a dictionary format to organize the travel characteristics. Examples are shown in Fig. 3.

3.3.3. Descriptive data

Individual attributes may also affect people's travel behavior. This information is included in a descriptive way (instead of using dummy binary variables as typical mathematical models). Examples are shown in Fig. 4.

3.3.4. Reasoning guidance with domain knowledge

The ability of LLMs to perform complex reasoning can be improved by designing prompts with strategies such as chain-of-thought (Wei et al., 2022a) and plan-to-solve prompting (Wang et al., 2023a). The essential idea is to guide LLMs on how to use the given information. This guidance is especially important when using LLMs to predict travel behavior without any task-specific training data. The prompts should include task-relevant domain knowledge and common behavioral considerations. The example of travel mode choice is shown in Fig. 5.

The first three aspects are domain knowledge. This can be customized based on input features and the specific task. The last paragraph is used to guide LLMs for numerical comparison. It is known that arithmetic and symbolic reasoning are challenging tasks for LLMs without well-designed prompts (Rae et al., 2021). Even for a simple task like providing LLMs with three numbers A, B, and C, and asking them to sort the numbers, LLMs can make many mistakes (this problem may be mitigated with more advanced LLMs). Therefore, we add the last paragraph, telling LLMs which mode has the lowest travel time or travel cost, and how large the relative differences are. Experiments show that this tends to be an effective way to enable LLMs for arithmetic reasoning in this context.

3.3.5. Interpretation and output

The last part of the prompt specifies the LLM output. We ask the model to return both a prediction and a short reason supporting that prediction. The reason makes the output auditable and helps diagnose whether the model is using the intended information. Asking for an explanation may also encourage more structured reasoning, similar to plan-to-solve prompting (Wang et al., 2023a), where explicitly requesting intermediate calculation steps can improve performance. The example prompt for travel behavior prediction is shown in Fig. 6. We also specify a JSON-style output format so that predictions and reasons can be extracted consistently.

3.3.6. Summary

An example of the final prompt is shown in Fig. 7. The sequence for different components is reorganized. It is worth noting that we do not include any training data information in the prompt (i.e., zero-shot), which allows the model to be used in any new context for travel behavior prediction.

3.4. Textual-gradient prompt optimization

The zero-shot framework in Section 3.3 relies on a *hand-crafted* prompt: a domain expert manually specifies reasoning-guidance cues (e.g., which mode has the lowest cost and the associated percentage savings) and the decision heuristics. While effective, this design has three drawbacks that were also raised in review. First, it requires substantial expert effort and must be re-engineered for each new task or context. Second, injecting precomputed cues (such as the lowest-cost mode and savings percentages) blurs the line between the LLM's reasoning and human feature engineering, making it difficult to attribute performance to the model itself. Third, prompt performance is known to

Example 1: Travel mode choice

```
The travel time and cost for each mode is expressed as the following dictionary format: {Travel time:
{Train: 202, Car: 160, Swissmetro: 97}, Travel cost: {Train: 108, Car: 136, Swissmetro: 140}}
```

Example 2: Travel purpose

```
The trip information is expressed as the following dictionary format: {Origin: Home, Destination: Office,
Departure time: 8:00 AM, Travel mode: Train}
```

Fig. 3. Example prompts for including travel characteristics.

Example 1: Travel mode choice

```
The person is not a current Train user. He/She does not own the Train annual pass.
```

Example 2: Travel purpose

```
The person is a 32-year-old woman. She has a full-time job as a designer. Her office is in the CBD area.
```

Fig. 4. Example prompts for individual attributes.

Example 1: Travel mode choice

```
Please consider the following aspects:
1. People are more likely to choose a travel mode with less travel cost and travel time, especially those
with significant cost or time saving. The trade off between time and cost can be quantified using value of
time.
2. Regular Train users may prefer to use Train.
3. Owners of Train annual pass are more likely to choose Train.

Swissmetro has the lowest travel time. Choosing it will save 39% travel time compared to Car and save 52%
travel time compared to Train. Train has the lowest travel cost. Choosing it will save 21% travel cost
compared to Car and save 23% travel cost compared to Swissmetro.
```

Fig. 5. Example prompts for reasoning guidance with domain knowledge.

Example 1: Travel mode choice

```
Please infer what is the mostly likely travel mode that the person will choose. Organize your answer in a
JSON object with two keys: "prediction" (the predicted travel mode) and "reason" (explanation that
supports your inference).
```

Fig. 6. Example prompts for interpretation and output.

Example 1: Travel mode choice

```
Your task is to predict a person's travel mode choice from Train, Car, and Swissmetro based on travel
time, travel cost information of each mode, and the person's attributes.

The travel time and cost for each mode is expressed as the following dictionary format: {Travel time:
{Train: 202, Car: 160, Swissmetro: 97}, Travel cost: {Train: 108, Car: 136, Swissmetro: 140}}

Swissmetro has the lowest travel time. Choosing it will save 39% travel time compared to Car and save
52% travel time compared to Train. Train has the lowest travel cost. Choosing it will save 21% travel cost
compared to Car and save 23% travel cost compared to Swissmetro.

The person is not a regular Train user. He/She does not own the Train annual pass.

Please infer what is the mostly likely travel mode that the person will choose. Organize your answer in
a JSON object with two keys: "prediction" (the predicted travel mode) and "reason" (explanation that
supports your inference).

Please consider the following aspects:
1. People are more likely to choose a travel mode with less travel cost and travel time, especially those
with significant cost or time saving. The trade off between time and cost can be quantified using value of
time.
2. Regular Train users may prefer to use Train.
3. Owners of Train annual pass are more likely to choose Train.
```

Fig. 7. Example complete prompts for travel behavior prediction.

be sensitive to wording, so a single hand-written prompt may be neither optimal nor robust. To address these issues, we propose a method that *learns* the prompt automatically from a small labeled budget, rather than hand-crafting it.

Our approach adapts the idea of textual-gradient prompt optimization (Pryzant et al., 2023; Yuksekgonul et al., 2024) to travel behavior prediction. The key insight is to treat prompt optimization as an analogue of gradient descent, but in the space of natural-language text instead of continuous parameters. We denote a pretrained LLM by LLM_θ, where θ collects its (frozen) parameters and a superscript indicates the role the model plays in the loop—a predictor $\mathrm{LLM}_{\theta^{\mathrm{Pred}}}$, a critic $\mathrm{LLM}_{\theta^{\mathrm{Critic}}}$, and an editor $\mathrm{LLM}_{\theta^{\mathrm{Edit}}}$ (these can be the same or different underlying models). Let p denote an instruction prompt and let $\mathrm{LLM}_{\theta^{\mathrm{Pred}}}(p, x)$ be the prediction of the predictor LLM for input x under prompt p. Given a small labeled set $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$ (the entire supervision budget), the empirical prediction loss of a prompt is

$$\mathcal{L}(p) = \frac{1}{B} \sum_{(x_i, y_i) \in \mathcal{B}} \mathbb{1}\left[\mathrm{LLM}_{\theta^{\mathrm{Pred}}}(p, x_i) \neq y_i\right]. \tag{5}$$

Because $\mathcal{L}(p)$ is not differentiable with respect to the text p, we replace the numerical gradient with a *textual gradient*: a natural-language critique of why the current prompt produces errors. Concretely, we collect the misclassified examples $\mathcal{E}(p) = \{(x_i, y_i) \in \mathcal{B} : \mathrm{LLM}_{\theta^{\mathrm{Pred}}}(p, x_i) \neq y_i\}$ and query a critic $\mathrm{LLM}_{\theta^{\mathrm{Critic}}}$ to summarize what the prompt is missing or misleading,

$$\delta = \mathrm{LLM}_{\theta^{\mathrm{Critic}}}(p, \mathcal{E}(p)), \tag{6}$$

where δ is the textual gradient (e.g., "the prompt does not tell the model to weigh the travel time–cost trade-off, nor to account for habitual train use"). An editor $\mathrm{LLM}_{\theta^{\mathrm{Edit}}}$ then performs the *gradient step*, rewriting the prompt in the direction indicated by the critique,

$$p' = \mathrm{LLM}_{\theta^{\mathrm{Edit}}}(p, \delta), \tag{7}$$

under the constraint that the rewrite stay general and not reference specific data points or reveal answers. To reduce the variance of this stochastic update and avoid local optima, we generate m candidate edits per prompt and retain the best k prompts (a beam) ranked by their accuracy on $\mathcal{B}$. The procedure iterates for T steps and returns the best prompt found. Crucially, the *same* budget $\mathcal{B}$ is used both to compute the textual gradient and to select among candidates, so the method consumes only B labeled examples in total. The full procedure is summarized in Algorithm 1; a concrete optimization step taken from our experiments is shown in Fig. 8.

Algorithm 1 Textual-gradient prompt optimization

Require: minimal prompt p_0; labeled budget $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$; steps T; beam width k; candidates per prompt m; predictor $\mathrm{LLM}_{\theta^{\mathrm{Pred}}}$, critic $\mathrm{LLM}_{\theta^{\mathrm{Critic}}}$, editor $\mathrm{LLM}_{\theta^{\mathrm{Edit}}}$

$\mathcal{P} \leftarrow \{p_0\}$ ▷ initialize beam
for $t = 1, \ldots, T$ **do**
  $\mathcal{C} \leftarrow \mathcal{P}$ ▷ elitism: keep current prompts
  for $p \in \mathcal{P}$ **do**
    $\mathcal{E}(p) \leftarrow \{(x_i, y_i) \in \mathcal{B} : \mathrm{LLM}_{\theta^{\mathrm{Pred}}}(p, x_i) \neq y_i\}$ ▷ errors on the budget
    if $\mathcal{E}(p) \neq \varnothing$ **then**
      $\delta \leftarrow \mathrm{LLM}_{\theta^{\mathrm{Critic}}}(p, \mathcal{E}(p))$ ▷ textual gradient
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathrm{LLM}_{\theta^{\mathrm{Edit}}}(p, \delta)^{(1)}, \ldots, \mathrm{LLM}_{\theta^{\mathrm{Edit}}}(p, \delta)^{(m)}\}$ ▷ gradient step
  $\mathcal{P} \leftarrow$ top-k prompts in $\mathcal{C}$ by accuracy on $\mathcal{B}$ ▷ beam selection
return $\mathrm{argmax}_{p \in \mathcal{P}} \left(1 - \mathcal{L}(p)\right)$

To make the procedure concrete, Fig. 8 shows one real optimization step from our Swissmetro experiments (gpt-4o-mini, $B = 100$). Starting from the minimal prompt p, which merely states the task, the predictor mislabels several budget examples (e.g., predicting Swissmetro when the true choice is Car). The critic LLM turns these errors into a textual gradient δ—a natural-language critique observing that the prompt fails to weigh the value of time versus cost, to consider familiarity with each mode, and to account for irregular usage patterns. The editor LLM then rewrites the prompt accordingly, producing an optimized prompt $p^\star$ that explicitly asks the model to reason about these factors. Notably, the optimized prompt adds general reasoning guidance rather than data-specific cues, which distinguishes it from the hand-crafted prompt in Section 3.3.

This method occupies a distinct point on the data-availability spectrum: it requires far less supervision than training a model from scratch, yet, unlike pure zero-shot prompting, it can exploit a handful of labeled examples to tailor the prompt to the task. In practice we optimize the prompt using an inexpensive LLM and then deploy the resulting prompt across models, so the optimization cost is incurred only once. Because the learned prompt is plain text, it remains fully interpretable and can be inspected, edited, and reused.

3.5. Embedding-based supervised prediction

Large language models (LLMs) encode semantic and contextual information into high-dimensional embedding vectors that can be leveraged for downstream prediction tasks. In the context of travel behavior prediction, we extract an embedding $\boldsymbol{h}_n$ for each sample n, typically representing a traveler, trip record, or textual description of the travel context. This embedding $\boldsymbol{h}_n$ serves as a compact yet expressive representation of the traveler's behavioral features, preferences, and context.

To predict the travel behavior label y_n (e.g., mode choice, trip purpose, or destination), we use $\boldsymbol{h}_n$ as input to a supervised learning model parameterized by $\boldsymbol{\beta}$. Let $\mathcal{D}^{\mathrm{Train}} = \{(\boldsymbol{h}_n, y_n)\}_{n=1}^{N_{\mathrm{train}}}$ denote the supervised training set, where N_{train} is the number of labeled training examples. The model is trained to minimize the empirical loss over this set:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \sum_{n=1}^{N_{\mathrm{train}}} \ell_n\left(\mathrm{SupervisedModel}(\boldsymbol{h}_n; \boldsymbol{\beta}), y_n\right) \tag{8}$$

Here, $\ell_n(\cdot, \cdot)$ denotes a sample-wise loss function that quantifies the discrepancy between the predicted and true labels. For classification tasks, ℓ_n is typically the cross-entropy loss, while for regression tasks, it may be the mean squared error or other appropriate metrics.

This approach decouples representation learning (handled by the LLM) from the prediction model, allowing flexibility in choosing downstream models, such as logistic regression, random forests, gradient boosting machines, or neural networks, based on task-specific requirements and computational considerations. Notably, the use of $\boldsymbol{h}_n$ as input enables generalization across diverse travel contexts, especially when the embeddings capture high-level behavioral semantics informed by large-scale pretraining.

3.6. Diagnostic methodology for memorization and explanations

We explicitly test whether strong direct-prompting performance comes from reasoning over the provided traveler and trip attributes or from memorizing familiar dataset patterns. The diagnostic idea is simple: if an LLM is reasoning from the input, its prediction should change when we make a different alternative clearly dominant, and it should remain stable when we change only superficial numeric values while preserving the same relative decision structure. Let $\mathcal{D}^{\mathrm{Diag}}$ be a diagnostic set and let $\mathcal{A}$ denote the set of feasible labels or alternatives. Since the prompt contains the full task description and the sample-specific information, we write $p(x_n)$ as the complete prompt constructed for observation $x_n \in \mathcal{D}^{\mathrm{Diag}}$. For a predictor LLM with parameters θ, the direct prediction is

$$\hat{y}_\theta\left(p(x_n)\right) = \mathrm{LLM}_\theta^{\mathrm{Pred}}\left(p(x_n)\right). \tag{9}$$

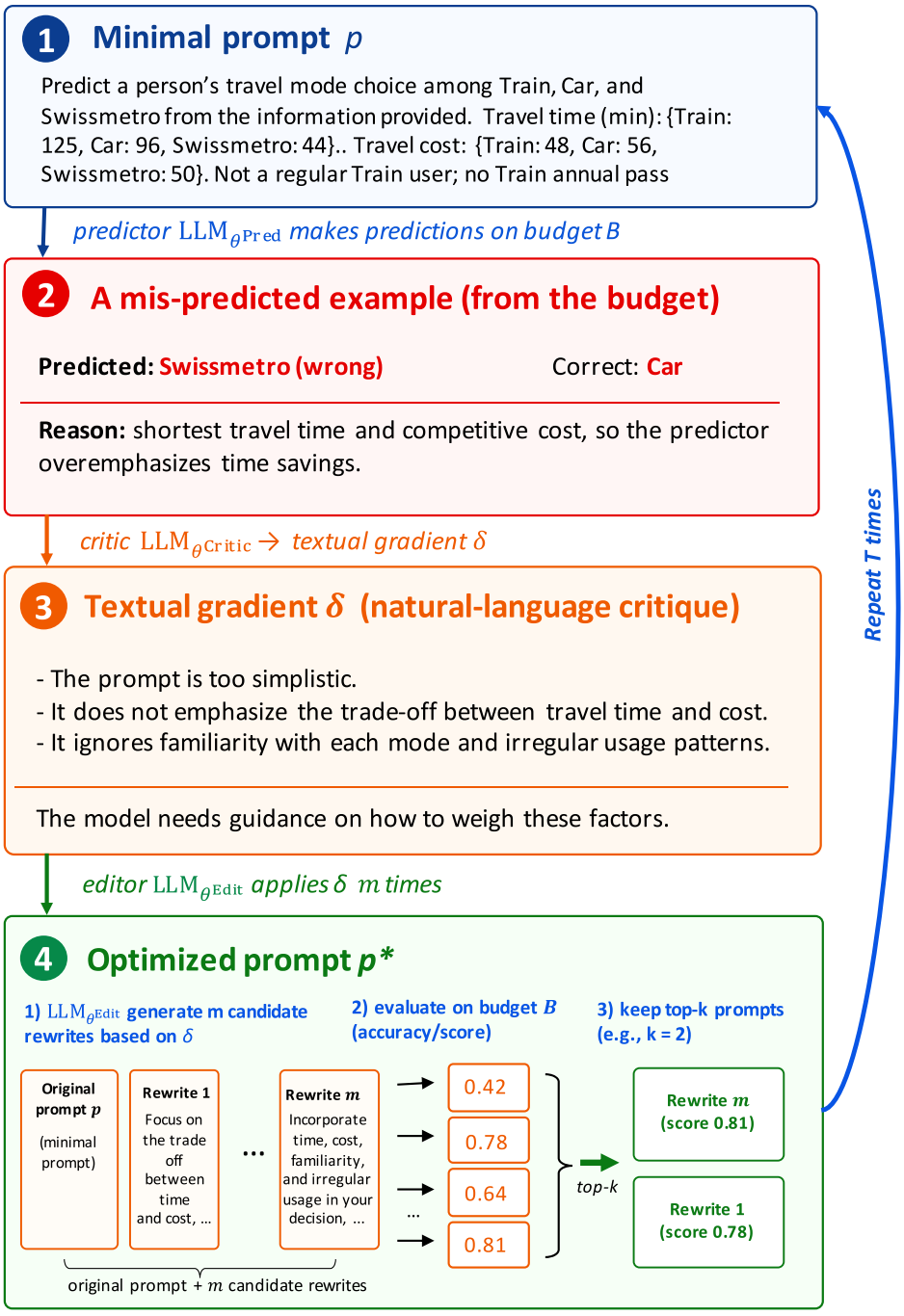

Fig. 8. Example textual-gradient optimization step.

The first diagnostic is a *dominance probe*. For each alternative $a \in \mathcal{A}$, define a counterfactual transformation $T_a^{\mathrm{dom}}(\cdot)$ that makes alternative a unambiguously preferable according to task-relevant observable attributes while keeping the rest of the input unchanged. A model that reasons from the supplied attributes should switch to the made-dominant alternative. We measure

$$\mathrm{DomAcc}(p,\theta) = \frac{1}{|D^{\mathrm{Diag}}| \cdot |\mathcal{A}|} \sum_{x_n \in D^{\mathrm{Diag}}} \sum_{a \in \mathcal{A}} \mathbb{1}\left[\hat{y}_\theta\left(p\left(T_a^{\mathrm{dom}}(x_n)\right)\right) = a\right]. \quad (10)$$

To inspect which alternatives drive this aggregate result, we also report the alternative-specific dominance accuracy

$$\mathrm{DomAcc}_a(p,\theta) = \frac{1}{|D^{\mathrm{Diag}}|} \sum_{x_n \in D^{\mathrm{Diag}}} \mathbb{1}\left[\hat{y}_\theta\left(p\left(T_a^{\mathrm{dom}}(x_n)\right)\right) = a\right], \quad a \in \mathcal{A}. \quad (11)$$

The second diagnostic is an *order-preserving transformation probe*. Let $T^{\mathrm{ord}}(\cdot)$ denote a transformation that changes surface numerical values while preserving the relative structure that should determine the decision, such as the ordering or trade-off pattern among alternatives. Memorization of exact records would make predictions sensitive to these value changes; reasoning over the input structure should preserve the original prediction. We report the agreement

$$\mathrm{OrdAgree}(p,\theta) = \frac{1}{|D^{\mathrm{Diag}}|} \sum_{x_n \in D^{\mathrm{Diag}}} \mathbb{1}\left[\hat{y}_\theta\left(p\left(T^{\mathrm{ord}}(x_n)\right)\right) = \hat{y}_\theta\left(p(x_n)\right)\right]. \quad (12)$$

We also diagnose explanation reliability. Let $\hat{r}_\theta(p(x_n))$ be the natural-language reason generated together with the prediction, and let $C(\hat{r}_\theta(p(x_n)))$ be the set of factual claims in the reason that can be checked against x_n. For each claim $c \in C(\hat{r}_\theta(p(x_n)))$, define a verifier $v(c, x_n) \in$

Table 1
Case-study tasks used in the unified evaluation.

Dataset	Prediction task	Choice labels	Primary input information
Swissmetro	Mode choice	Train, Car, Swissmetro	Alternative travel time/cost and traveler attributes
London LPMC	Mode choice	Walk, Cycle, Public transport, Drive	Trip attributes and traveler attributes
NHTS	Trip purpose	Working, Shopping, Social, Others	Household, person, trip timing, trip duration, and travel mode attributes

$\{1, 0, -1\}$ as follows: $v(c, x_n) = 1$ if c is directly supported by the attributes in x_n; $v(c, x_n) = 0$ if c cannot be verified from the available attributes in x_n; and $v(c, x_n) = -1$ if c contradicts the attributes in x_n. We then summarize factual consistency and contradiction as

$$\text{Cons} = \frac{\sum_{x_n \in D^{\text{Diag}}} \sum_{c \in C(\hat{r}_\theta(p(x_n)))} \mathbb{1}[v(c, x_n) = 1]}{\sum_{x_n \in D^{\text{Diag}}} |C(\hat{r}_\theta(p(x_n)))|}, \tag{13}$$

$$\text{Contr} = \frac{\sum_{x_n \in D^{\text{Diag}}} \sum_{c \in C(\hat{r}_\theta(p(x_n)))} \mathbb{1}[v(c, x_n) = -1]}{\sum_{x_n \in D^{\text{Diag}}} |C(\hat{r}_\theta(p(x_n)))|}. \tag{14}$$

These diagnostics do not prove causal reasoning in a philosophical sense. They directly test the empirical distinction needed in this paper: whether the LLM's prediction and rationale are grounded in the supplied attributes or in memorized surface records.

4. Case study

4.1. Evaluation design

The empirical study follows the same data-availability logic as the methodology. We evaluate three travel-behavior prediction tasks in parallel: Swissmetro mode choice, London mode choice, and NHTS trip-purpose prediction. This design compares how each model–method combination behaves across tasks as the labeled-data budget changes. We then use the combined results table to discuss label efficiency, cross-dataset consistency, prompt learning, and embedding-based feature augmentation. Finally, we report diagnostic analyses on prompt transferability, possible data contamination, and explanation reliability.

4.1.1. Datasets

The first mode-choice task uses the Swissmetro stated-preference survey dataset (Bierlaire et al., 2001). The survey analyzes the potential adoption of Swissmetro, a proposed maglev underground system, relative to car and train. The data contain 1004 individuals and 9036 responses, with travelers choosing among train, car, and Swissmetro under different travel-time, travel-cost, and traveler-attribute conditions. The second mode-choice task uses the London Passenger Mode Choice (LPMC) survey (Hillel et al., 2018), with four alternatives: walk, cycle, public transport, and drive. The third task uses the 2017 U.S. National Household Travel Survey (NHTS), which records daily trips, socio-demographic features, and trip purposes. We group trip purpose into four categories: working, shopping, social, and others. Table 1 summarizes the three tasks.

For each dataset, we use balanced sampling to construct labeled training subsets of three sizes — a labeled-data *budget* of $B \in \{10, 100, 1000\}$ examples — together with a fixed test set of 200 examples. Balanced sampling preserves the class proportions in each labeled subset. To enable a fair, budget-matched comparison across methods, all methods that consume labeled data (the supervised benchmarks and the textual-gradient prompt optimizer) receive the *same* sampled subset at each budget and are evaluated on the same held-out test set. For the textual-gradient method, the budget B is used both to compute the textual gradient and to select prompts (Section 3.4), so its total supervision equals B. Results are averaged over 5 random samples (seeds), and we report the mean and standard deviation.

Tables 2–4 report detailed summaries for the Swissmetro, London LPMC, and NHTS feature sets.

Table 2
Swissmetro training and testing data statistics.

Features	Mean	Std	Min	Max
Train travel time (min)	161.2	77.3	31	1049
Train cost (Swiss franc)	87.4	65.1	8	576
Car travel time (min)	138.7	96.8	32	1560
Car cost (Swiss franc)	86.9	46.9	8	520
Swissmetro travel time (min)	85.7	53.5	8	790
Swissmetro cost (Swiss franc)	108.3	82.7	11	768
Regular train user (Yes = 1)	0.35	0.48	0	1
Own train annual pass (Yes = 1)	0.13	0.34	0	1

Table 3
London LPMC training and testing data statistics.

Features	Mean	Std	Min	Max
Trip distance (m)	4605.3	4782.4	77.0	40941.0
Walking travel time (h)	1.129	1.118	0.025	9.278
Cycling travel time (h)	0.362	0.352	0.006	3.052
Public transport access time (h)	0.160	0.092	0.000	1.189
Public transport rail time (h)	0.090	0.177	0.000	1.467
Public transport bus time (h)	0.172	0.190	0.000	2.147
Public transport interchange time (h)	0.044	0.078	0.000	0.865
Public transport interchanges	0.369	0.619	0.000	4.000
Driving travel time (h)	0.282	0.252	0.000	2.061
Transit fare (pounds)	1.563	1.535	0.000	13.490
Driving fuel cost (pounds)	0.832	0.823	0.000	10.090
Driving congestion charge (pounds)	1.071	3.178	0.000	10.500
Driving traffic delay share	0.336	0.201	0.000	1.250
Age	39.5	19.2	5.0	99.0
Female (Yes = 1)	0.526	0.499	0.000	1.000
Driving license (Yes = 1)	0.617	0.486	0.000	1.000
Car ownership	0.981	0.752	0.000	2.000

Table 4
NHTS training and testing data statistics.

Features	Mean	Std	Min	Max
With child (Yes = 1)	0.39	0.49	0	1
Primary activity last week is work (Yes = 1)	0.89	0.31	0	1
Trip duration less than 10 min (Yes = 1)	0.39	0.48	0	1
Trip duration greater than 25 min (Yes = 1)	0.28	0.45	0	1
Travel mode is walk/bike (Yes = 1)	0.08	0.27	0	1
Travel mode is drive (Yes = 1)	0.90	0.31	0	1
Travel mode is taxi/rideshare (Yes = 1)	0.00	0.07	0	1
Travel mode is transit (Yes = 1)	0.01	0.12	0	1
Travel on weekends (Yes = 1)	0.24	0.43	0	1
Only one person in trip (Yes = 1)	0.59	0.49	0	1

4.1.2. Compared methods and implementation

We compare the LLM-based methods with four benchmark models: multinomial logit (MNL), random forest (RF), neural networks (NNs), and TabPFN (Hollmann et al., 2023, 2025). MNL is the canonical travel behavior prediction model, while random forests and neural networks are widely used machine-learning baselines for mode-choice prediction (Wang et al., 2024c; Kamkar et al., 2025; Abulibdeh, 2023). TabPFN is a transformer pre-trained on large collections of synthetic tabular tasks; it performs in-context tabular classification without per-task gradient training and is currently among the strongest small-data baselines. It is therefore a demanding comparator in the low-data regime and a useful analogue to the LLM setting: both TabPFN and LLMs bring a pre-trained

Table 5
Prompt-generation comparison.

Dataset	LLM configuration (predict / critic / edit)	Budget	Minimal	Hand-crafted	Text-gradient
Swissmetro	Predict/critic/edit: GPT-3.5-turbo	$B = 10$	0.432 ± 0.023	0.586 ± 0.052	0.576 ± 0.029 (↑ 33.3%)
		$B = 100$	0.432 ± 0.023	0.586 ± 0.052	0.559 ± 0.036 (↑ 29.4%)
	Predict/critic/edit: GPT-4o-mini	$B = 10$	0.547 ± 0.037	0.520 ± 0.047	0.562 ± 0.050 (↑ 2.7%)
		$B = 100$	0.547 ± 0.037	0.520 ± 0.047	0.592 ± 0.042 (↑ 8.2%)
London LPMC	Predict/critic/edit: GPT-3.5-turbo	$B = 10$	0.343 ± 0.024	0.433 ± 0.029	0.401 ± 0.051 (↑ 16.9%)
		$B = 100$	0.343 ± 0.024	0.433 ± 0.029	0.440 ± 0.018 (↑ 28.3%)
	Predict/critic/edit: GPT-4o-mini	$B = 10$	0.403 ± 0.011	0.376 ± 0.013	0.392 ± 0.031 (↓ 2.7%)
		$B = 100$	0.403 ± 0.011	0.376 ± 0.013	0.407 ± 0.058 (↑ 1.0%)
NHTS	Predict/critic/edit: GPT-3.5-turbo	$B = 10$	0.363 ± 0.012	0.362 ± 0.024	0.350 ± 0.026 (↓ 3.6%)
		$B = 100$	0.363 ± 0.012	0.362 ± 0.024	0.377 ± 0.021 (↑ 3.9%)
	Predict/critic/edit: GPT-4o-mini	$B = 10$	0.373 ± 0.013	0.361 ± 0.022	0.369 ± 0.016 (↓ 1.1%)
		$B = 100$	0.373 ± 0.013	0.361 ± 0.022	0.362 ± 0.027 (↓ 2.9%)

prior to a new prediction task, but they differ in whether the prior is tabular or language-based.

The LLM experiments use GPT-3.5 (gpt-3.5-turbo-1106), GPT-4 (gpt-4-turbo-2024-04-09), GPT-4o, GPT-4o-mini, Llama 3.1 8B, and Llama 3.1 70B, depending on the analysis. The final prompts used for the case studies are reported in Appendix A. We set the temperature to 0 to reduce output randomness. The supervised baselines are trained and tuned separately for each budget-specific training split, using only the training data. For MNL, model coefficients are estimated on the corresponding budget split; in the sklearn multinomial-logistic implementation used for London LPMC and NHTS, the inverse regularization strength is selected from $C \in \{0.1, 1, 10\}$ by stratified cross-validation, while the Swissmetro MNL uses the specified utility model estimated by maximum likelihood. For RF, we tune the number of trees, maximum depth, and minimum leaf size; for NN, we tune the hidden-layer structure, L_2 penalty, and learning rate. Grid search uses up to three stratified folds, capped by the smallest class count in the budget sample. TabPFN uses its standard in-context classification setting. The embedding experiments use `text-embedding-3-small` on the same prompt-style input text used for direct prediction; the native 1536-dimensional embeddings are standardized before fitting the MNL, RF, and NN heads, and no dimensionality reduction is applied. For the textual-gradient optimizer, we use $T = 6$ optimization steps, a beam width of $k = 2$, and $m = 2$ candidate edits per step. In the prompt-generation analysis, the predictor, critic, and editor LLMs are set either to GPT-3.5-turbo or to GPT-4o-mini; the optimized prompt is then evaluated either with the same predictor LLM or transferred to other predictor LLMs.

4.2. Results

4.2.1. Prompt generation by textual gradients

We first isolate the prompt-generation question before comparing LLMs with non-LLM baselines. The purpose of textual-gradient optimization is to replace the expert hand-crafted prompt with a prompt learned from a small labeled budget. Table 5 therefore compares three prompt designs across the case-study tasks: a minimal prompt with no expert reasoning cues, the expert hand-crafted prompt, and the learned prompt produced by textual-gradient optimization. We report the predictor–critic–editor configurations available in the experiments at budgets $B = 10$ and $B = 100$. Minimal and hand-crafted prompts use no labeled examples, so their values are repeated across budget rows. The percentages in parentheses report the textual-gradient change relative to the minimal prompt; green upward arrows indicate improvement and red downward arrows indicate decrease.

The results show that textual-gradient optimization can recover competitive prompts without manually injected numerical cues, but its gains are task-dependent. On Swissmetro, the learned prompt improves over the minimal prompt for both GPT-3.5-turbo and GPT-4o-mini. For GPT-3.5-turbo, the minimal prompt performs poorly (0.432), while the learned prompt reaches 0.576 at $B = 10$, essentially matching the expert hand-crafted prompt. For GPT-4o-mini, textual-gradient optimization improves over both the minimal and hand-crafted prompts, reaching 0.592 at $B = 100$. The newly completed GPT-3.5 runs for London LPMC and NHTS show a similar pattern when the starting prompt is weak: on London, textual-gradient optimization raises GPT-3.5 from 0.343 to 0.440 at $B = 100$, slightly exceeding the expert prompt (0.433); on NHTS, all GPT-3.5 prompt variants are closer, but the learned prompt is best at $B = 100$ (0.377). In contrast, for GPT-4o-mini the minimal prompts are already strong on London and NHTS, so the learned prompt is close to the minimal prompt rather than consistently better. This supports a more precise methodological claim: learned prompts can achieve performance comparable to expert hand-crafted prompts while avoiding hard-coded lowest-cost, lowest-time, and percentage-savings statements, and the largest gains appear when the starting prompt leaves room for improvement.

Table 6 compares the prompt content for one task, Swissmetro mode choice. The minimal prompt mainly states the prediction task and provides the raw travel attributes. The hand-crafted prompt adds expert domain knowledge and per-observation computed cues, including which mode has the lowest time or cost and the corresponding percentage savings. The textual-gradient prompt instead learns general behavioral principles. A representative optimized prompt states: *"Your task is to analyze and predict an individual's travel mode choice among Train, Car, and Swissmetro, taking into account various influencing factors such as familiarity with each mode, the perceived trade-offs between time and cost, and any irregular usage patterns that may affect their decision-making process. Consider how these elements interact to shape their preferences and choices in travel"*. Thus, the learned prompt recovers the same broad concepts as the expert prompt — habitual mode use and the time–cost trade-off — but does not encode row-specific arithmetic cues.

4.2.2. Prompt transferability across predictor LLMs

The previous subsection evaluates prompt optimization when the same LLM is used as predictor, critic, and editor. We next ask whether a prompt optimized with one predictor–critic–editor configuration remains useful when deployed with another predictor $\mathrm{LLM}_{\theta}^{\mathrm{Pred}}$. Table 7 focuses on deployment predictor LLMs that are not already shown as prompt-generation configurations in Table 5, and separates results by the LLM configuration that produced the textual-gradient prompt.

The transfer results show that textual-gradient prompts remain useful when deployed with larger GPT-family predictors, but they do not uniformly dominate fixed prompts. On Swissmetro, both GPT-3.5- and GPT-4o-mini-sourced prompts improve over the minimal and hand-crafted prompts for GPT-4-turbo and GPT-4o. On London, GPT-4o is already strong with fixed prompts, while the transferred GPT-4o-mini prompt remains competitive (0.499). On NHTS, the hand-crafted prompt is strongest for both GPT-4-turbo and GPT-4o, while

Table 6
Swissmetro prompt-design comparison.

Prompt design	Main content	Interpretation
Minimal prompt	States the task, available alternatives, travel time and cost, and traveler attributes.	Measures the LLM's unaided reasoning from the raw prompt representation.
Expert hand-crafted prompt	Adds domain rules about cost, time, value of time, regular train use, and annual-pass ownership; also includes per-row lowest-time/lowest-cost modes and percentage savings.	Strong but manually engineered; part of the reasoning is supplied by the researcher.
Textual-gradient prompt	Adds general learned guidance about familiarity with each mode, time–cost trade-offs, and irregular usage patterns.	Competitive with the hand-crafted prompt while replacing manual cues with data-driven natural-language guidance.

Table 7
Prompt transfer to additional predictor LLMs.

Dataset	Direct predict LLM	Minimal	Hand-crafted	Text-grad from GPT-3.5	Text-grad from GPT-4o-mini
Swissmetro	GPT-4-turbo	0.565 ± 0.043	0.569 ± 0.040	0.583 ± 0.031	0.583 ± 0.035
	GPT-4o	0.561 ± 0.023	0.558 ± 0.044	0.579 ± 0.032	0.575 ± 0.026
London LPMC	GPT-4-turbo	0.413 ± 0.032	0.433 ± 0.016	0.441 ± 0.040	0.440 ± 0.034
	GPT-4o	0.501 ± 0.046	0.497 ± 0.039	0.483 ± 0.068	0.499 ± 0.042
NHTS	GPT-4-turbo	0.364 ± 0.015	0.397 ± 0.018	0.380 ± 0.016	0.358 ± 0.015
	GPT-4o	0.368 ± 0.010	0.426 ± 0.026	0.366 ± 0.024	0.375 ± 0.013

the GPT-3.5-sourced textual-gradient prompt improves GPT-4-turbo over the minimal baseline (0.364 to 0.380). Overall, the table supports prompt transferability as a useful option, especially when the transferred prompt captures task structure not fully expressed in a minimal prompt, while also showing that expert prompts can remain competitive on some tasks.

4.2.3. Budget-matched model comparison

We now compare prediction frameworks under the same labeled-data budget. Table 8 reorganizes the evidence by budget: $B = 0$ corresponds to direct LLM prediction without supervised learning, $B = 10$ and $B = 100$ include textual-gradient prompt optimization and supervised baselines trained on the same number of examples, and $B = 1000$ reports only the supervised and tabular baselines. This scope reflects both the methodological purpose and the computational cost of the LLM-based procedures. Textual-gradient optimization is intended for the scarce-label regime; extending it to $B = 1000$ would require many additional predictor, critic, and editor API calls across seeds, datasets, candidate edits, and optimization steps, making the run time and API cost disproportionate to the main question of label efficiency.

To compare the LLM-as-direct-predictor framework against the supervised baselines, the "Direct LLM" rows report the best-performing LLM configuration at each budget (the underlying model and prompt are listed in the table footnote); gray cells mark the best method within each budget block. The comparison shows a cross-over rather than universal LLM dominance. At $B = 10$, the best direct-LLM configuration is the top method on all three tasks, exceeding every supervised and tabular baseline—consistent with the LLM prior being most valuable when labels are scarce. At $B = 100$ the picture is mixed: direct LLM remains best on Swissmetro (0.592), whereas MNL becomes strongest on London (0.512) and NHTS (0.407). At $B = 1000$, the conventional supervised and tabular models dominate because they can exploit the larger labeled sample, whereas prompt optimization is no longer the natural tool. No single LLM is uniformly best across tasks: GPT-3.5-turbo and the small, recent GPT-4o-mini account for most of the best direct-LLM entries.

The analysis in Appendix B tests these differences with paired McNemar exact tests on the same test samples, with Holm–Bonferroni correction within each dataset–budget family. The statistical results sharpen the budget-dependent interpretation. On Swissmetro at $B = 10$, the textual-gradient direct LLM is significantly better than every comparator, including the minimal prompt, the hand-crafted prompt, TabPFN, MNL, RF, and NN. At $B = 100$, however, its Swissmetro advantage is no longer significant relative to the minimal prompt, TabPFN, or MNL, although it remains significant relative to the hand-crafted prompt and the weaker RF/NN baselines. On London at $B = 100$, the direction is reversed: TabPFN, MNL, RF, and NN significantly outperform the textual-gradient LLM. On NHTS, almost all pairwise differences are not significant after correction. Thus, the significance analysis supports the main conclusion that the LLM advantage is strongest in the scarce-label regime and should not be generalized as uniform dominance across tasks or budgets.

4.2.4. Direct prompts versus hidden-feature prediction

Finally, we compare the two LLM-based prediction frameworks: using the LLM as a direct text predictor versus using LLM hidden representations as features for a supervised model. Table 9 reports the unified embedding sweep using text-embedding-3-small and supervised MNL, random forest, and neural-network heads under the same labeled budgets. We restrict this comparison to $B = 10$ and $B = 100$, where LLM-based methods are most relevant; extending the embedding sweep to $B = 1000$ would require embedding thousands of additional training records across datasets, seeds, and model heads, producing a large number of API calls and substantially longer run time.

The embedding sweep clarifies that hidden-feature prediction is useful but not uniformly superior to direct prompting in the scarce-label regime. At $B = 10$, direct prompt prediction is stronger on all three tasks, consistent with the idea that the LLM's internal prior is especially valuable when only a handful of labels are available. At $B = 100$, embedding-based heads become more competitive on London and NHTS, where the neural-network and random-forest heads slightly exceed or match the direct textual-gradient prompt, but they still lag behind the direct prompt on Swissmetro. Thus, direct prompting is the more label-efficient LLM framework at very small budgets, while embedding features become a reasonable supervised alternative as the labeled budget grows.

4.2.5. Robustness and explanation diagnostics

We use the diagnostic tests to answer a direct question: is strong Swissmetro zero-shot performance caused by reasoning over the given travel attributes, or by memorization of a canonical benchmark dataset? Following the methodology in Section 3.6, Table 10 reports DomAcc,

Table 8
Budget-matched model comparison.

Budget	Model / framework	Swissmetro	London LPMC	NHTS
$B = 0$	Direct LLM (best zero-shot)†	0.586 ± 0.052	0.433 ± 0.029	0.408 ± 0.022
$B = 10$	Direct LLM (best)†	0.576 ± 0.029	0.401 ± 0.051	0.369 ± 0.016
$B = 10$	TabPFN	0.488 ± 0.037	0.335 ± 0.059	0.326 ± 0.045
$B = 10$	MNL	0.455 ± 0.059	0.365 ± 0.056	0.325 ± 0.017
$B = 10$	Random forest	0.452 ± 0.077	0.365 ± 0.053	0.324 ± 0.033
$B = 10$	Neural network	0.464 ± 0.067	0.345 ± 0.063	0.310 ± 0.025
$B = 100$	Direct LLM (best)†	0.592 ± 0.042	0.499 ± 0.042	0.380 ± 0.016
$B = 100$	TabPFN	0.567 ± 0.039	0.501 ± 0.029	0.363 ± 0.046
$B = 100$	MNL	0.589 ± 0.018	0.512 ± 0.028	0.407 ± 0.016
$B = 100$	Random forest	0.540 ± 0.040	0.474 ± 0.037	0.394 ± 0.007
$B = 100$	Neural network	0.512 ± 0.052	0.459 ± 0.044	0.375 ± 0.023
$B = 1000$	TabPFN	0.646 ± 0.028	0.592 ± 0.031	0.443 ± 0.015
$B = 1000$	MNL	0.605 ± 0.032	0.568 ± 0.037	0.445 ± 0.004
$B = 1000$	Random forest	0.621 ± 0.027	0.567 ± 0.037	0.434 ± 0.025
$B = 1000$	Neural network	0.632 ± 0.030	0.575 ± 0.030	0.428 ± 0.017

†"Direct LLM" reports the best-performing model and prompt per task at each budget. B=0 (hand-crafted prompt): GPT-3.5-turbo on Swissmetro and London, Llama-3.1-70B on NHTS. B=10 (textual-gradient): GPT-3.5-turbo on Swissmetro and London, GPT-4o-mini on NHTS. B=100 (textual-gradient): GPT-4o-mini on Swissmetro; GPT-4o with the GPT-4o-mini-discovered prompt on London; GPT-4-turbo with the GPT-3.5-discovered prompt on NHTS. Gray cells mark the best result within the $B = 10$ and $B = 100$ comparison blocks.

Table 9
Direct prompting versus LLM embeddings.

Budget	Framework	Swissmetro	London LPMC	NHTS
$B = 10$	Direct prompt prediction	0.562 ± 0.050	0.392 ± 0.031	0.369 ± 0.016
$B = 10$	Embedding + MNL	0.458 ± 0.043	0.328 ± 0.032	0.314 ± 0.045
$B = 10$	Embedding + random forest	0.448 ± 0.047	0.323 ± 0.033	0.317 ± 0.053
$B = 10$	Embedding + neural network	0.448 ± 0.030	0.316 ± 0.040	0.324 ± 0.044
$B = 100$	Direct prompt prediction	0.592 ± 0.042	0.407 ± 0.058	0.362 ± 0.027
$B = 100$	Embedding + MNL	0.486 ± 0.029	0.429 ± 0.034	0.354 ± 0.018
$B = 100$	Embedding + random forest	0.512 ± 0.015	0.426 ± 0.015	0.364 ± 0.027
$B = 100$	Embedding + neural network	0.490 ± 0.015	0.437 ± 0.025	0.364 ± 0.020

Table 10
Memorization probe.

Model	DomAcc	DomAcc_a	OrdAgree
GPT-4o-mini	0.879	0.94/0.85/0.85	0.875
GPT-3.5-turbo	0.604	0.29/0.71/0.81	0.900

DomAcc_a reports Train/Car/Swissmetro. The dominance probe uses 80 held-out Swissmetro rows and constructs one counterfactual per alternative by setting that alternative's travel time and cost to near-zero. The rescale probe multiplies all travel times and costs by the same constant and reports OrdAgree.

Table 11
Explanation faithfulness diagnostics.

Model	Cons	Contr
GPT-3.5-turbo	100.0%	0.0%
GPT-4-turbo	98.6%	3.9%
Llama-3.1 8B	99.7%	0.7%
Llama-3.1 70B	99.3%	2.2%

DomAcc_a, and OrdAgree: whether the model switches to an artificially dominant mode overall and by alternative, and whether its prediction is stable when all exact time and cost values are rescaled while preserving their relative structure.

The high OrdAgree for both GPT-4o-mini and GPT-3.5-turbo shows that their predictions are not keyed to memorized exact Swissmetro numeric records. GPT-4o-mini also has high DomAcc, predicting the made-dominant alternative in 87.9% of counterfactual cases, which shows that its predictions respond to changes in the supplied attributes. GPT-3.5-turbo is less reliable in the dominance test, especially when Train is made dominant, indicating weaker counterfactual reasoning rather than exact-record memorization. Overall, these diagnostics support the central interpretation that the observed performance is mainly driven by input-sensitive reasoning over travel-time and travel-cost structure, while still leaving open the possibility of broader pre-training familiarity with the task domain.

We also evaluate the reliability of the natural-language explanations. For each generated reason, we check factual claims that can be verified from the inputs, including the lowest travel-time mode, the lowest travel-cost mode, regular train use, and annual-pass status. Table 11 reports Cons and Contr from Section 3.6. The explanations are largely consistent with the provided facts, but we do not treat them as guaranteed causal explanations. Instead, they are best viewed as auditable rationales that help diagnose when the model is using relevant attributes and when it is attributing preferences not present in the input.

Table 12 illustrates the same point qualitatively. In Case 1, the LLM correctly predicts Swissmetro and gives an explanation grounded in the input attributes. In Case 2, the prediction is wrong but the explanation is internally reasonable. Cases 3 and 4 show the limits of explanation-based interpretability: the model can make reasoning mistakes or attribute an unstated preference to the traveler. These examples motivate the systematic faithfulness diagnostics above and support a qualified interpretation of LLM explanations.

4.2.6. *Prompt sensitivity diagnostics*

The prompt-generation results show that learned prompts can match or exceed expert hand-crafted prompts, but it is still useful to understand which parts of the expert prompt matter. We therefore retain

Table 12
Example cases of using LLMs for travel mode choice prediction.

Case 1: *Correct prediction and reasonable explanations*	
Input:	{Travel time: {Train: 106, Car: 90, Swissmetro: 34}, Travel cost: {Train: 72, Car: 70, Swissmetro: 78}}. The person is not a regular Train user. He/She does not own the Train annual pass.
Prediction:	Swissmetro (MNL prediction is Car)
Actual:	Swissmetro
Reason:	Based on the given information, the person is not a regular Train user and does not own the Train annual pass. Swissmetro offers the lowest travel time, saving 62% compared to Car and 68% compared to Train. Although Car has the lowest travel cost, the significant time saving of Swissmetro makes it the most likely choice for the person.
Case 2: *Wrong prediction but reasonable explanations*	
Input:	{Travel time: {Train: 114, Car: 96, Swissmetro: 55}, Travel cost: {Train: 82, Car: 80, Swissmetro: 105}}. The person is not a regular Train user. He/She does not own the Train annual pass.
Prediction:	Car (MNL prediction is Car)
Actual:	Train
Reason:	Based on the given information, the person is not a regular Train user and does not own the Train annual pass. Considering the trade-off between travel time and cost, the person is likely to choose the Car as it has the lowest travel cost and the difference in travel time compared to Swissmetro is not significant enough to outweigh the cost savings.
Case 3: *Wrong prediction and unreasonable explanations*	
Input:	{Travel time: {Train: 124, Car: 96, Swissmetro: 59}, Travel cost: {Train: 83, Car: 80, Swissmetro: 133}}. The person is a regular Train user. He/She does not own the Train annual pass.
Prediction:	Train (MNL prediction is Train)
Actual:	Swissmetro
Reason:	The person is a regular Train user, and regular Train users prefer to use Train. Although Swissmetro has the lowest travel time, the person's preference for Train and the absence of a Train annual pass make it more likely for the person to choose Train over Swissmetro. Additionally, Car has the lowest travel cost, but the person's regular use of Train and preference for it make it more likely for the person to choose Train over Car despite the cost difference.
Case 4: *Wrong prediction and hallucinated explanations*	
Input:	{Travel time: {Train: 95, Car: 130, Swissmetro: 92}, Travel cost: {Train: 29, Car: 44, Swissmetro: 32}}. The person is not a regular Train user. He/She does not own the Train annual pass.
Prediction:	Swissmetro (MNL prediction is Train)
Actual:	Train
Reason:	Based on the given information, the person is not a regular Train user and does not own the Train annual pass. Swissmetro has the lowest travel time, saving 29% compared to Car and 3% compared to Train. Although Train has the lowest travel cost, the person's attributes suggest a preference for time saving, making Swissmetro the most likely choice.

the original ablation study on the Swissmetro mode-choice task. The ablation keeps the same input information but changes how it is represented:

- **Exclude reasoning guidance**: We remove the sentences of "Swissmetro has the lowest travel time. Choosing it will save 39% travel time compared to ..." from the prompt, which guide the LLM to compare cost and travel time.
- **Exclude domain knowledge**: We remove the sentences of "Please consider the following aspects: 1. People are more likely to choose ..." from the prompt, which provides domain knowledge on how people typically make their decisions.
- **Change structural data to non-structural data**: Instead of using structural data as inputs (for example, {Travel time: {Train: 202, Car: 160, Swissmetro: 97}}), we directly tell the LLM about this information with descriptive sentences (e.g., the travel time of train is 202 min).

As shown in Table 13, prompt design plays a markedly different role for GPT-3.5 and GPT-4. For GPT-3.5, excluding reasoning guidance leads to the largest performance degradation, with accuracy dropping by 5.8%, suggesting that explicit comparative reasoning cues are critical for this model to weigh attributes such as travel time and cost. Changing structural data to non-structural, descriptive inputs also decreases accuracy by 4.4%, highlighting the importance of organized input representation. In contrast, GPT-4 remains largely stable across ablation settings, with accuracy decreases below 1% in all cases. This robustness suggests that stronger LLMs are less dependent on manually specified prompt components, which is consistent with the transferability results above.

5. Conclusion and future work

This study investigates large language models (LLMs) as data-efficient complements to conventional travel behavior prediction methods. The paper is organized around a data-availability spectrum: zero-shot direct prediction when no labeled local data are available, textual-gradient prompt optimization when only a small labeled budget can be collected, and embedding-based supervised prediction when LLM representations can be combined with conventional classifiers. The case study treats Swissmetro mode choice, London mode choice, and NHTS trip-purpose prediction as parallel tasks, but presents the results through four focused comparisons: prompt generation, prompt transferability, budget-matched model performance, and direct prompting versus hidden-feature prediction. This framing clarifies that the relevant question is not whether LLMs universally replace models such as MNL, random forests, neural networks, or TabPFN, but where each framework is useful as labeled data become more available.

Across the three tasks, the empirical results show a consistent crossover pattern. In the extreme low-data regime, LLM-based prediction is highly competitive and often outperforms supervised or tabular baselines given the same labeled budget. As the budget grows, however, supervised and tabular foundation models overtake the direct LLM predictors. The paired significance tests confirm that this advantage is statistically strongest for Swissmetro at $B = 10$, while several apparent differences at $B = 100$ and on NHTS are not significant after correction. The textual-gradient method provides a practical middle point on this spectrum: it can learn a reusable prompt from a small labeled set, match or exceed expert hand-crafted prompts without manually injected reasoning cues, and transfer across predictor LLMs. Its gains are context-dependent, which reinforces the importance of

Table 13
Ablation study results for the Swissmetro mode choice dataset.

Model	Prompt design	Performance	
		Accuracy (Decrease %)[a]	F1 Score
LLM-GPT-3.5	Full prompt	0.586 ± 0.052	0.572 ± 0.058
	Exclude reasoning guidance	0.552 ± 0.036 (−5.8%)	0.526 ± 0.039
	Exclude domain knowledge	0.583 ± 0.052 (−0.5%)	0.571 ± 0.052
	Change structural data to non-structural data	0.560 ± 0.037 (−4.4%)	0.538 ± 0.040
LLM-GPT-4	Full prompt	0.570 ± 0.054	0.569 ± 0.054
	Exclude reasoning guidance	0.565 ± 0.044 (−0.8%)	0.562 ± 0.043
	Exclude domain knowledge	0.569 ± 0.050 (−0.2%)	0.568 ± 0.052
	Change structural data to non-structural data	0.567 ± 0.044 (−0.5%)	0.565 ± 0.043

[a] Values in the parenthesis represent the mean accuracy decrease compared to the full prompt.

evaluating prompt-learning methods under a budget-matched protocol rather than treating prompt optimization as a universally beneficial step. The embedding comparison further suggests that LLMs can contribute useful high-level representations for small-sample supervised learning, although the available results favor direct prompt prediction at the smallest budget and show embedding-based heads becoming more competitive at $B = 100$ on London and NHTS.

The diagnostic analyses directly test the reasoning-versus-memorization concern. Counterfactual tests on Swissmetro show that the LLM responds to relative travel-time and travel-cost structure rather than simply reproducing memorized benchmark values. The explanation analysis shows that generated rationales are often factually consistent with the input and useful for auditing model behavior, but they should not be treated as guaranteed causal explanations: the model can still make reasoning errors or attribute unstated preferences to travelers.

Several directions remain for future research. First, the evaluation should be extended to larger and more diverse datasets, additional travel behavior tasks such as departure time choice, route choice, and activity scheduling, and repeated experimental runs that better separate model capability from sampling and decoding variance. Second, prompt optimization can be improved through richer search strategies, alternative selection criteria for very small budgets, and reusable prompt libraries for transportation tasks. Third, future work should study few-shot and in-context learning strategies that expose LLMs to representative examples without leaking test labels, especially for learning quantitative trade-offs such as value of time or cost sensitivity. Finally, practical deployment requires continued work on privacy, computational cost, hallucination reduction, calibration, fairness, and explanation faithfulness before LLM-based predictors can be used in decision-critical transportation planning workflows.

CRediT authorship contribution statement

Baichuan Mo: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Hanyong Xu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Ruoyun Ma:** Writing – review & editing, Methodology, Data curation. **Jung-Hoon Cho:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Data curation, Conceptualization. **Dingyi Zhuang:** Writing – review & editing, Methodology, Data curation. **Xiaotong Guo:** Writing – original draft, Supervision, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Jinhua Zhao:** Writing – review & editing, Resources, Funding acquisition, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the author(s) used OpenAI's Codex in order to assist manuscript preparation process. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Funding source

This research is partially supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore-MIT Alliance for Research and Technology (SMART).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Prompts

For each of the three datasets we report the prompts used in the three direct-prediction settings: the *minimal* zero-shot prompt (task statement only), the *expert hand-crafted* prompt (with domain-knowledge cues), and the *textual-gradient* learned prompt. The minimal prompts are listed in Table A.14; the hand-crafted prompts are presented as examples below, where all changeable parts are underlined and differ across samples; and the learned prompts are listed in Table A.16.

In addition to the task statement, every prompt appends the corresponding structured input (travel times/costs and traveler attributes for mode choice; trip context for trip purpose) and the JSON output instruction (see Table A.15).

Appendix B. Statistical significance of the budget-matched comparison

We assess whether the accuracy differences in the budget-matched comparison (Table 8) are statistically significant using *McNemar's exact test*, the standard paired test for two classifiers evaluated on the same test instances. Unlike a comparison of mean accuracies, McNemar conditions on each individual example, so it tests whether two methods make *systematically* different errors rather than whether their averages differ. We pair predictions at the level of the individual test sample and pool the five seeds, giving $n = 5 \times 200 = 1000$ paired samples per comparison. The reference method is the textual-gradient prompt (GPT-4o-mini); Δacc is its accuracy minus the comparator's, so a positive value means the textual-gradient prompt is more accurate. Within each (dataset, budget) family of six comparisons we control the family-wise error rate with the Holm–Bonferroni correction; reported significance uses the Holm-adjusted p-values ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$; unmarked entries are not significant) (see Table B.17).

The tests support a budget-dependent reading of the results. In the extreme low-data regime on Swissmetro (B=10), the textual-gradient prompt is significantly more accurate than every comparator, including the minimal and hand-crafted prompts and all supervised/tabular

Table A.14
Minimal (zero-shot) prompts.

Task	Minimal prompt
Swissmetro mode choice	Your task is to predict a person's travel mode choice among Train, Car, and Swissmetro from the information provided.
London LPMC mode choice	Your task is to predict a person's travel mode choice among Walk, Cycle, Public transport, and Drive from the information provided.
NHTS trip-purpose prediction	Your task is to infer the purpose of a person's trip among working, shopping, social, and others from the information provided.

Table A.15
Expert hand-crafted prompt examples.

Task	Prompt
Swissmetro mode choice	Your task is to predict a person's travel mode choice from Train, Car, and Swissmetro based on travel time, travel cost information of each mode, and the person's attributes. The travel time and cost for each mode is expressed as the following dictionary format: {Travel time: {Train: 94, Car: 56, Swissmetro: 36}, Travel cost: {Train: 62, Car: 36, Swissmetro: 71}} Swissmetro has the lowest travel time, Choosing it will save 36% travel time compared to Car and save 62% travel time compared to Train. Car has the lowest travel cost. Choosing it will save 49% travel cost compared to Swissmetro and save 42% travel cost compared to Train. The person is a regular Train user. He/She does not own the Train annual pass. Please infer the most likely travel mode that the person will choose. Organize your answer in a JSON object with two keys: "prediction" (the predicted travel mode) and "reason" (explanation that supports your inference). Please consider the following aspects: 1. People are more likely to choose a travel mode with less travel cost and travel time, especially those with significant cost or time saving. The trade-off between time and cost can be quantified using value of time. 2. Regular Train users prefer to use Train. 3. Owners of Train annual pass are more likely to choose Train.
London LPMC mode choice	Your task is to predict a person's travel mode choice from Walk, Cycle, Public transport, and Drive based on travel time, travel cost of each mode, and the person's attributes. Travel time (min): {Walk: 19.5, Cycle: 8.4, Public transport: 12.0, Drive: 5.6}. Travel cost (GBP): {Walk: 0, Cycle: 0, Public transport: 1.5, Drive: 0.25}. Drive has the lowest travel time. Walk has the lowest travel cost. The person is 21 years old, male, and the household owns 1 car(s). The trip distance is 697 m. Please infer the most likely travel mode. Organize your answer in a JSON object with two keys: "prediction" (one of Walk, Cycle, Public transport, Drive) and "reason". Please consider the following aspects: 1. People prefer modes with less travel time and cost (the trade-off can be quantified by value of time). 2. Short trips favor walking or cycling; long trips favor driving or public transport. 3. Households owning more cars are more likely to drive; zero-car households rely on walking, cycling, or public transport.
NHTS trip-purpose prediction	Your task is to infer the purpose of a person's trip, classifying it into one of four categories: "working", "shopping", "social", and "others". You will make the inference based on the following information related to the socio-demographic information and trip information of the traveler. The person lives in a household with child. As of last week, His/her primary activity is working. The trip happens on weekdays and starts after 5pm. The trip lasts less than 10 minutes. The mode of transportation is not walking or biking. There were 2 or more people on the trip. Please also consider the following reasoning rules to support your inference: 1. Trips that happen on weekends are more likely to be social or shopping trips 2. Trips that happen before 10am are more likely to be working trips. Trips that happen after 5pm are more likely to be social trips 3. A trip with only one person is highly likely to be a working trip 4. Trips with walking or biking transportation mode are highly likely to be social trips 5. Working and social trip duration is usually longer than 25 min 6. People with child are more likely to trip purpose of "others", such as school or daycare 7. If the person's primary activity last week is working, it is more likely to be a working trip Please follow the above reasoning rules strictly and avoid violating them. Infer the most likely trip purpose. Organize your final answer in a JSON object with two keys: "prediction" (the predicted travel purpose) and "reason" (explanation that supports your inference in one paragraph).

Table A.16
Textual-gradient learned prompts.

Task	Learned prompt
Swissmetro mode choice	Your task is to analyze and predict an individual's travel mode choice among Train, Car, and Swissmetro, taking into account various influencing factors such as familiarity with each mode, the perceived trade-offs between time and cost, and any irregular usage patterns that may affect their decision-making process. Consider how these elements interact to shape their preferences and choices in travel.
London LPMC mode choice	Your task is to assess the provided information to identify an individual's likely travel mode—Walk, Cycle, Public transport, or Drive. Pay particular attention to demographic factors such as age and household car ownership, as these significantly influence travel preferences. Additionally, evaluate the impact of trip distance and the specific context surrounding these demographic factors on travel time and cost. Clearly outline how to balance the importance of travel time versus travel cost in various scenarios to develop a well-rounded understanding of the individual's travel behavior.
NHTS trip-purpose prediction	Your task is to determine the likely purpose of a person's trip, categorizing it as working, shopping, social, or other. Consider contextual factors such as the time of day, household composition, trip duration, and the number of individuals involved, as these elements can provide valuable insights into the trip's intent. Aim for clarity in distinguishing between overlapping purposes while making your inference.

Table B.17
Per-sample McNemar significance of the textual-gradient prompt versus each comparator (Δacc = textual-gradient minus comparator; positive favors the LLM). Holm-adjusted significance over $n = 1000$ paired samples (5 seeds $\times$200).

Dataset	Budget	vs. minimal	vs. hand-crafted	vs. TabPFN	vs. MNL	vs. RF	vs. NN
Swissmetro	B=10	+0.018*	+0.066***	+0.078***	+0.111***	+0.114***	+0.102***
	B=100	+0.038	+0.086***	+0.019	−0.003	+0.046*	+0.074***
London LPMC	B=10	−0.009	+0.016	+0.057	+0.027	+0.027	+0.047
	B=100	+0.006	+0.031	−0.094***	−0.105***	−0.067**	−0.052*
NHTS	B=10	−0.004	+0.009	+0.044	+0.045	+0.046	+0.060*
	B=100	−0.012	+0.001	−0.001	−0.045	−0.032	−0.013

baselines. As the labeled budget grows, this advantage narrows: at B=100 the learned prompt is statistically indistinguishable from the minimal prompt, TabPFN, and MNL on Swissmetro, though it still significantly exceeds the hand-crafted prompt and the weaker tree/network baselines. On London at B=100 the direction reverses — the tuned supervised and tabular models significantly outperform the LLM — while on NHTS almost no pairwise difference survives correction. These results justify the cautious, budget-conditioned language used in the main text: the LLM's advantage is statistically real mainly in the scarce-label regime, and conventional models become significantly preferable once enough local labels are available.

Data availability

Data will be made available on request.

References

Abulibdeh, A., 2023. Analysis of mode choice affects from the introduction of Doha Metro using machine learning and statistical analysis. Transp. Res. Interdiscip. Perspect. 20, 100852. http://dx.doi.org/10.1016/j.trip.2023.100852.

Ali, M., Fissha, Y., 2026. Propose adjustable support vector machine approach for classifying imbalanced work travel mode choice data. Transp. Res. Interdiscip. Perspect. 35, 101786. http://dx.doi.org/10.1016/j.trip.2025.101786.

Amin, M.M., Cambria, E., Schuller, B.W., 2023. Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of chatgpt. IEEE Intell. Syst. 38, 15–23. http://dx.doi.org/10.1109/MIS.2023.3254179.

Arreeras, T., Sunmud, S., Thanasupsin, K., Phonsitthangkun, S., 2025. Determinants of intercity mode choice preferences and travel behavior in a border tourism city. Transp. Res. Interdiscip. Perspect. 34, 101638. http://dx.doi.org/10.1016/j.trip.2025.101638.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q.V., Xu, Y., Fung, P., 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. URL http://arxiv.org/abs/2302.04023. arXiv:2302.04023 [cs].

Ben-Akiva, M., Lerman, S.R., 1985. Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press, Cambridge, MA.

Bierlaire, M., Axhausen, K., Abay, G., 2001. The acceptance of modal innovation: The case of swissmetro. In: Swiss Transport Research Conference.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. URL http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs].

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M.T., Zhang, Y., 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. URL http://arxiv.org/abs/2303.12712. arXiv:2303.12712 [cs].

Chen, C., He, Y., Wang, H., Chen, J., Luo, Q., 2024. Delayptc-llm: Metro passenger travel choice prediction under train delays with large language models. http://dx.doi.org/10.48550/arXiv.2410.00052, URL http://arxiv.org/abs/2410.00052. arXiv:2410.00052 [cs].

Chu, Z., Chen, J., Chen, Q., Yu, W., He, T., Wang, H., Peng, W., Liu, M., Qin, B., Liu, T., 2024. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. http://dx.doi.org/10.48550/arXiv.2309.15402, URL http://arxiv.org/abs/2309.15402. arXiv:2309.15402 [cs].

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., et al., 2022. A survey on in-context learning. arXiv preprint arXiv:2301.00234.

Fang, B., Yang, Z., Wang, S., Di, X., 2024. Travellm: Could you plan my new public transit route in face of a network disruption?. http://dx.doi.org/10.48550/arXiv.2407.14926, URL http://arxiv.org/abs/2407.14926. arXiv:2407.14926 [cs].

Feng, J., Du, Y., Zhao, J., Li, Y., 2025. Agentmove: A large language model based agentic framework for zero-shot next location prediction. http://dx.doi.org/10.48550/arXiv.2408.13986, URL http://arxiv.org/abs/2408.13986. arXiv:2408.13986 [cs].

Gong, Letian, Lin, Yan, Zhang, Xinyue, Lu, Yiwen, Han, Xuedi, Liu, Yichen, Guo, Shengnan, Lin, Youfang, Wan, Huaiyu, 2024. Mobility-llm: learning visiting intentions and travel preference from human mobility data with large language models. In: Advances in Neural Information Processing Systems. 37, Curran Associates, Inc., pp. 36185–36217. http://dx.doi.org/10.52202/079017-1141, https://proceedings.neurips.cc/paper_files/paper/2024/hash/3fb6c52aeb11e09053c16eabee74dd7b-Abstract-Conference.html.

Gurnee, W., Tegmark, M., 2023. Language models represent space and time. URL http://arxiv.org/abs/2310.02207. arXiv:2310.02207 [cs].

Hillel, T., Elshafie, M.Z.E.B., Jin, Y., 2018. Recreating passenger mode choice-sets for transport simulation: A case study of London, UK. Proc. Inst. Civ. Eng. – Smart Infrastruct. Constr. 171, 29–42. http://dx.doi.org/10.1680/jsmic.17.00018.

Hollmann, N., Müller, S., Eggensperger, K., Hutter, F., 2023. Tabpfn: A transformer that solves small tabular classification problems in a second. In: International Conference on Learning Representations. ICLR.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S.B., Schirrmeister, R.T., Hutter, F., 2025. Accurate predictions on small data with a tabular foundation model. Nature 637, 319–326.

Huang, F., Kwak, H., An, J., 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In: Companion Proceedings of the ACM Web Conference 2023. ACM, Austin TX USA, pp. 294–297. http://dx.doi.org/10.1145/3543873.3587368, URL https://dl.acm.org/doi/10.1145/3543873.3587368.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T., 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst. 43, 1–55. http://dx.doi.org/10.1145/3703155, arXiv:2311.05232 [cs].

Kamkar, H., Saidi, S., Ansari Esfeh, M., 2025. Understanding travel behavior and mode choice prediction for university commuters: insights from discrete choice models and machine learning. Transp. Res. Interdiscip. Perspect. 34, 101754. http://dx.doi.org/10.1016/j.trip.2025.101754.

Kobayashi, A., Takeda, N., Yamazaki, Y., Kamisaka, D., 2023. Modeling and generating human mobility trajectories using transformer with day encoding. In: Proceedings of the 1st International Workshop on the Human Mobility Prediction Challenge. pp. 7–10.

Kuzman, T., Mozetič, I., Ljubešić, N., 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. http://dx.doi.org/10.48550/arXiv.2303.03953, URL http://arxiv.org/abs/2303.03953. arXiv:2303.03953 [cs].

Li, S., Feng, J., Chi, J., Hu, X., Zhao, X., Xu, F., 2024a. Limp: Large language model enhanced intent-aware mobility prediction. http://dx.doi.org/10.48550/arXiv.2408.12832, URL http://arxiv.org/abs/2408.12832. arXiv:2408.12832 [cs].

Li, X., Huang, F., Lv, J., Xiao, Z., Li, G., Yue, Y., 2024b. Be more real: Travel diary generation using llm agents and individual profiles. http://dx.doi.org/10.48550/arXiv.2407.18932, URL http://arxiv.org/abs/2407.18932. arXiv:2407.18932 [cs].

Liang, Y., Liu, Y., Wang, X., Zhao, Z., 2023. Exploring large language models for human mobility prediction under public events. URL http://arxiv.org/abs/2311.17351. arXiv:2311.17351 [cs].

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., Ge, B., 2023. Summary of chatgpt-related research and perspective towards the future of large language models. Meta-Radiology 1, 100017. http://dx.doi.org/10.1016/j.metrad.2023.100017.

Liu, T., Li, M., Yin, Y., 2024. Can large language models capture human travel behavior? evidence and insights on mode choice. http://dx.doi.org/10.2139/ssrn.4937575, URL https://papers,ssrn.com/abstract=4937575.

Luo, Y., Cao, Z., Jin, X., Liu, K., Yin, L., 2024. Deciphering human mobility: Inferring semantics of trajectories with large language models. In: 2024 25th IEEE International Conference on Mobile Data Management. MDM, pp. 289–294. http://dx.doi.org/10.1109/MDM61037.2024.00060, URL https://ieeexplore.ieee.org/abstract/document/10591679.

Mo, B., Shen, Y., Zhao, J., 2018. Impact of built environment on first-and last-mile travel mode choice. Transp. Res. Rec. 2672, 40–51.

Mo, B., Wang, Q.Y., Moody, J., Shen, Y., Zhao, J., 2021. Impacts of subjective evaluations and inertia from existing travel modes on adoption of autonomous mobility-on-demand. Transp. Res. Part C: Emerg. Technol. 130, 103281.

OpenAI, 2024a. Gpt-4 technical report. http://dx.doi.org/10.48550/arXiv.2303.08774, URL http://arxiv.org/abs/2303.08774.arXiv:2303.08774 [cs].

OpenAI, 2024b. Text-embedding-3-small. https://openai.com/blog/new-embedding-models-and-api-updates/.

Pryzant, R., Iter, D., Li, J., Lee, Y.T., Zhu, C., Zeng, M., 2023. Automatic prompt optimization with "gradient descent" and beam search. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 7957–7968.

Qin, Z., Zhang, P., Wang, L., Ma, Z., 2025. Lingotrip: Spatiotemporal context prompt driven large language model for individual trip prediction. J. Public Transp. 27, 100117. http://dx.doi.org/10.1016/j.jpubtr.2025.100117.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1, 9.

Rae, J.W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al., 2021. Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446.

Shao, C., Xu, F., Fan, B., Ding, J., Yuan, Y., Wang, M., Li, Y., 2024. Chain-of-planned-behaviour workflow elicits few-shot mobility generation in llms. http://dx.doi.org/10.48550/arXiv.2402.09836, URL http://arxiv.org/abs/2402.09836. arXiv:2402.09836 [cs].

Singh, H., Verma, N., Wang, Y., Bharadwaj, M., Fashandi, H., Ferreira, K., Lee, C., 2024. Personal large language model agents: A case study on tailored travel planning. In: Dernoncourt, F., Preo¸tiuc Pietro, D., Shimorina, A. (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. Association for Computational Linguistics, Miami, Florida, US, pp. 486–514. http://dx.doi.org/10.18653/v1/2024.emnlp-industry.37, URL https://aclanthology.org/2024.emnlp-industry.37/.

Srisurin, P., Ahmad, I., Ali, N., Khan, R.S., Phuksuksakul, N., Hussain, Q., Suparp, S., 2026. A comparative study of predicting travel mode choice of school children using explainable machine learning techniques. Transp. Res. Interdiscip. Perspect. 37, 102035. http://dx.doi.org/10.1016/j.trip.2026.102035.

Tang, Y., Wang, Z., Qu, A., Yan, Y., Wu, Z., Zhuang, D., Kai, J., Hou, K., Guo, X., Zheng, H., Luo, T., Zhao, J., Zhao, Z., Ma, W., 2024. Itinera: Integrating spatial optimization with large language models for open-domain urban itinerary planning. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. pp. 1413–1432. http://dx.doi.org/10.18653/v1/2024.emnlp-industry.104, URL http://arxiv.org/abs/2402.07204. arXiv:2402.07204 [cs].

Team, G., 2024. Gemini: A family of highly capable multimodal models. http://dx.doi.org/10.48550/arXiv.2312.11805, URL http://arxiv.org/abs/2312.11805. arXiv:2312.11805 [cs].

Teams, L., 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G., 2023a. Llama: Open and efficient foundation language models. http://dx.doi.org/10.48550/arXiv.2302.13971, URL http://arxiv.org/abs/2302.13971. arXiv:2302.13971 [cs].

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023b. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Wang, L., Duan, P., He, Z., Lyu, C., Chen, X., Zheng, N., Yao, L., Ma, Z., 2024b. Ai-driven day-to-day route choice. http://dx.doi.org/10.48550/arXiv.2412.03338, URL http://arxiv.org/abs/2412.03338. arXiv:2412.03338 [cs].

Wang, X., Fang, M., Zeng, Z., Cheng, T., 2023b. Where would i go next? large language models as human mobility predictors. URL http://arxiv.org/abs/2308.15197. arXiv:2308.15197 [physics].

Wang, J., Jiang, R., Yang, C., Wu, Z., Onizuka, M., Shibasaki, R., Koshizuka, N., Xiao, C., 2024a. Large language models as urban residents: An llm agent framework for personal mobility generation. URL http://arxiv.org/abs/2402.14744. arXiv:2402.14744 [cs].

Wang, S., Mo, B., Zheng, Y., Hess, S., Zhao, J., 2024c. Comparing hundreds of machine learning and discrete choice models for travel demand modeling: An empirical benchmark. Transp. Res. Part B: Methodol. 190, 103061.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P., 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Toronto, Canada, pp. 2609–2634. http://dx.doi.org/10.18653/v1/2023.acl-long.147, URL https://aclanthology.org/2023.acl-long.147,

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al., 2022a. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al., 2022b. Chain-of-thought prompting elicits reasoning in large language models. Adv. Neural Inf. Process. Syst. 35, 24824–24837.

Wu, X., He, H., Wang, Y., Wang, Q., 2024. Pretrained mobility transformer: A foundation model for human mobility. arXiv preprint arXiv:2406.02578.

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., Su, Y., 2024. Travelplanner: A benchmark for real-world planning with language agents. arXiv preprint arXiv:2402.01622.

Xie, C., Zou, D., 2024. A human-like reasoning framework for multi-phases planning task with large language models. arXiv preprint arXiv:2405.18208.

Xue, H., Salim, F.D., 2022. Promptcast: A new prompt-based learning paradigm for time series forecasting. URL https://arxiv.org/abs/2210.08964v4.

Xue, H., Voutharoja, B.P., Salim, F.D., 2022. Leveraging language foundation models for human mobility forecasting. In: Proceedings of the 30th International Conference on Advances in Geographic Information Systems. pp. 1–9.

Yuksekgonul, M., Bianchi, F., Boen, J., Liu, S., Lu, P., Guestrin, C., Zou, J., 2024. Textgrad: Automatic "differentiation" via text. arXiv preprint arXiv:2406.07496.

Zhai, X., Tian, H., Li, L., Zhao, T., 2024. Enhancing travel choice modeling with large language models: A prompt-learning approach. arXiv preprint arXiv:2406.13558.

Zhang, B., Ding, D., Jing, L., 2023. How would stance detection techniques evolve after the launch of chatgpt?. http://dx.doi.org/10.48550/arXiv.2212.14548, URL http://arxiv.org/abs/2212.14548. arXiv:2212.14548 [cs].

Zhang, Z., Sun, Y., Wang, Z., Nie, Y., Ma, X., Sun, P., Li, R., 2024. Large language models for mobility in transportation systems: A survey on forecasting tasks. arXiv preprint arXiv:2405.02357.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.Y., Wen, J.R., 2023. A survey of large language models. URL http://arxiv.org/abs/2303.18223. arXiv:2303.18223 [cs].

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., Chi, E., 2023. Least-to-most prompting enables complex reasoning in large language models. URL http://arxiv.org/abs/2205.10625. arXiv:2205.10625 [cs].