# PATHTRAJ: PROMPT-AWARE TWO-STAGE HIERARCHICAL TRAJECTORY GENERATION FROM LANGUAGE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Synthetic trajectory generation offers a practical means of obtaining human mobility data when access to real records is limited by privacy concerns, collection costs, and proprietary restrictions. However, existing language-conditioned methods typically map a global condition directly to trajectories. As a result, they often overlook intermediate travel requirements, degrade on long journeys and under ambiguous instructions, and fail to preserve fine-grained motion dynamics. To address these limitations, we propose **PathTraj**, a prompt-aware two-stage hierarchical framework that decomposes generation into route planning and trajectory synthesis. In the planning stage, PathTraj grounds the instruction in the road network to infer feasible origin and destination segments, then decodes a connected sequence of segments under graph constraints and converts it into continuous geometric guidance. In the synthesis stage, a diffusion model generates coordinate-level trajectories conditioned on the planned route and the instruction, while a Motion Representation Extractor derives local motion features from intermediate denoising states to guide realistic speed and acceleration dynamics. Experiments on two real-world taxi trajectory datasets show that PathTraj outperforms existing baselines in instruction faithfulness and motion realism, and maintains this advantage across travel distances and levels of instruction detail. The code is available at `https://anonymous.4open.science/r/PathTraj-EC63`.

## 1 INTRODUCTION

Human mobility trajectories record how people move through urban environments and support a wide range of applications, such as traffic management, urban planning, and location-based services (Luca et al., 2021; Zheng, 2015; Barbosa et al., 2018; Zheng et al., 2014). However, obtaining large-scale trajectory data that reflects real travel behavior remains challenging due to privacy concerns, collection costs, and proprietary restrictions (De Montjoye et al., 2013; Du et al., 2023; Gursoy et al., 2018). Trajectory generation offers a practical alternative by synthesizing realistic mobility records that support model development and evaluation without exposing raw travel data (Cao & Li, 2021). Beyond reproducing aggregate mobility patterns, many applications require trajectories that realize user-specified journeys, such as a morning commute from a residential area to a business district. Therefore, modern scenarios call for controllable generation methods that translate high-level travel requirements into trajectories that are both faithful to the intended trip and realistic in their spatial and temporal dynamics.

Deep generative models, particularly diffusion models, have advanced trajectory synthesis. DiffTraj (Zhu et al., 2023) demonstrated that diffusion models can capture complex mobility distributions and generate high-fidelity GPS trajectories, while ControlTraj (Zhu et al., 2024) introduced road-topology guidance to generate trajectories along specified paths. Language conditioning has also been explored for trajectory prediction and traffic simulation (Bae et al., 2024; Chang et al., 2025), and InsTraj (Zhu et al., 2026) extended it to urban trajectory generation from natural-language travel intentions. However, translating these intentions into coherent road-level plans and realizing them as trajectories with realistic local dynamics remains challenging. Three limitations are particularly relevant to language-conditioned generation. (i) Global language conditioning may overlook requirements beyond the endpoints, such as route directness and road preferences. (ii) Maintaining spatial coherence becomes more difficult over longer journeys or when instructions provide limited



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

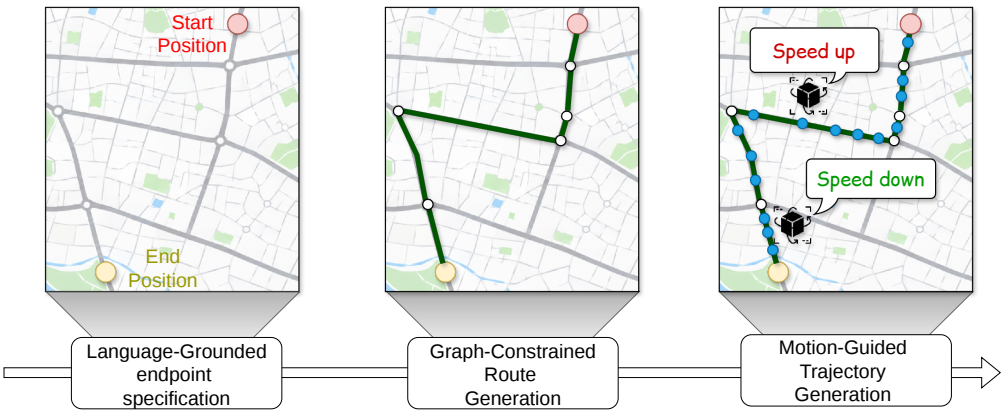


Figure 1: Overview of the proposed PathTraj framework. It first grounds natural-language instructions into feasible endpoints, then constructs graph-constrained routes, and finally employs motion-guided diffusion to generate coordinate-level trajectories.

spatial detail. (iii) An emphasis on overall geometry and spatial distributions can leave local speed variations and acceleration patterns insufficiently modeled.

To summarize, these limitations point to three requirements for language-conditioned trajectory generation: (i) **Faithfulness**: Travel requirements should be translated into explicit spatial plans that preserve their relationships and order. Such plans should cover not only the origin and destination but also intermediate route preferences, so that the generated trajectory realizes the complete journey described by the user. (ii) **Consistency**: Generation quality should remain stable across trajectory lengths and levels of instruction clarity. Long trajectories should stay anchored to a coherent spatial structure, and underspecified instructions should still yield plausible, road-valid trips. (iii) **Realism**: Beyond plausible route geometry, generated trajectories should capture fine-grained motion characteristics. In particular, their speed variations and acceleration patterns should be consistent with observed travel behavior.

To meet these requirements, we introduce **PathTraj**, a **p**rompt-**a**ware **t**wo-stage **h**ierarchical framework for **traj**ectory generation that bridges language understanding and trajectory generation through explicit route planning. As illustrated in **Figure 1**, PathTraj decomposes generation into a planning stage and a synthesis stage. The planning stage first uses language-grounded endpoint specification to infer a feasible origin–destination pair on the road network. Conditioned on these endpoints and the instruction, graph-constrained route planning decodes a connected sequence of road segments and converts it into continuous geometric guidance. In the synthesis stage, motion-guided trajectory generation uses a diffusion model to produce coordinate-level trajectories conditioned on the planned route and the instruction. A Motion Representation Extractor derives local motion features from intermediate denoising states, allowing motion guidance to adapt as the trajectory is refined. Consequently, explicit language grounding and route planning support instruction faithfulness and spatial coherence, while adaptive motion guidance promotes realistic dynamics.

The main contributions of this work are summarized as follows:

- We propose PathTraj, a prompt-aware hierarchical framework for language-conditioned trajectory generation. Instead of mapping a global condition directly to coordinates, PathTraj first plans an explicit route on the road network and then synthesizes trajectories under its guidance.
- We design three complementary modules: language-grounded endpoint specification, graph-constrained route planning, and motion-guided trajectory generation, which respectively infer feasible endpoints, construct connected routes, and synthesize motion-aligned trajectories.
- Experiments on two real-world taxi trajectory datasets show that PathTraj outperforms existing baselines across travel distances and levels of instruction clarity, and ablation studies confirm the contribution of each component.

## 2 Related Work

**Trajectory Generation.** Trajectory generation has evolved alongside advances in generative modeling (Luca et al., 2021; Zheng et al., 2014). Early statistical models captured regular mobility patterns but struggled to represent nonlinear spatio-temporal dependencies (Jiang et al., 2016). Recurrent networks subsequently improved the sequential modeling of mobility (Feng et al., 2018). GANs (Liu et al., 2018; Rao et al., 2020; Cao & Li, 2021; Choi et al., 2021; Henke et al., 2023) and VAEs (Chen et al., 2021) then made synthesized trajectories more realistic and diverse. More recently, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have become a leading paradigm owing to their stable training and strong distribution-learning capacity, and DiffTraj (Zhu et al., 2023) demonstrated their effectiveness for GPS trajectory synthesis. Subsequent work has shifted from unconditional synthesis toward controllable generation, using road networks, trip-level attributes, and mobility priors as structured conditions (Wei et al., 2024; Zhu et al., 2024; Cao et al., 2025a; Long et al., 2025; Zhang et al., 2025). Language conditioning further allows users to express high-level travel intentions. It has been explored in trajectory prediction (Bae et al., 2024), traffic simulation (Chang et al., 2025), and urban trajectory generation (Zhu et al., 2026). However, these methods typically inject a global condition into a single generation process. They lack an explicit intermediate representation that links instructions to road-level routes.

**Conditional Diffusion Generation.** Conditional diffusion models have progressed from class labels to rich semantic and structural control (Cao et al., 2025b). In text-to-image generation, pretrained language encoders (Radford et al., 2021; Saharia et al., 2022), cross-attention conditioning (Rombach et al., 2022), and classifier-free guidance (Ho & Salimans, 2022) align outputs with complex prompts. Spatial conditions such as edge maps, depth, and poses further enable precise control over output structure (Zhang et al., 2023; Mou et al., 2024). Diffusion Transformers (Peebles & Xie, 2023) offer a scalable backbone that has been widely adopted for conditional generation. This paradigm has since been extended to other modalities, including video (Singer et al., 2022) and speech (Liu et al., 2023). Together, these studies suggest that combining semantic conditions with explicit structural guidance gives more control than semantic conditioning alone. PathTraj applies this insight to trajectories. Rather than asking users to supply structural conditions, it derives an explicit route from the language instruction and uses this route, together with motion representations, to guide diffusion.

## 3 Definitions and Problem Setup

**Definition 1 (Route).** Given a road network $\mathcal{G}$, a route $\mathbf{r} = (s_1, \ldots, s_K)$ is an ordered sequence of connected road segments, where $s_K$ is the destination segment (Wei et al., 2024; Zhu et al., 2024). A route describes the high-level spatial structure of a trip and constrains how the journey progresses.

**Definition 2 (Trajectory).** A trajectory $\mathbf{x} = (p_1, \ldots, p_n)$ is an ordered sequence of GPS observations with uniform time intervals, where $p_i = (\mathrm{lat}_i, \mathrm{lon}_i)$. Unlike a route, which represents discrete road-level connectivity, a trajectory captures fine-grained movement along the trip, from which motion characteristics such as speed and acceleration can be derived.

**Definition 3 (Travel Instruction).** A travel instruction $I$ is a natural-language description of an intended journey that specifies where, why, and how the user wishes to travel. Its content is organized around 16 semantic attributes covering trip intent, trip profile, route strategy and behavior, and road context (See **Appendix A.2** for more details).

**Language-Conditioned Trajectory Generation** Given a travel instruction $I$ and a road network $\mathcal{G}$, instruction-guided trajectory generation aims to learn a conditional distribution $p_\theta(\mathbf{x} \mid I, \mathcal{G})$ , from which a trajectory $\tilde{\mathbf{x}} \sim p_\theta(\mathbf{x} \mid I, \mathcal{G})$ can be generated. Beyond preserving conventional spatio-temporal fidelity, $\tilde{\mathbf{x}}$ should satisfy the three requirements outlined in Section 1. First, it should be **semantically faithful** to the constraints expressed in $I$. Second, it should maintain **generation consistency**, following a coherent and road-valid route on $\mathcal{G}$ regardless of trip length or instruction clarity. Third, it should exhibit **motion realism** at the coordinate level, preserving plausible local movement patterns.

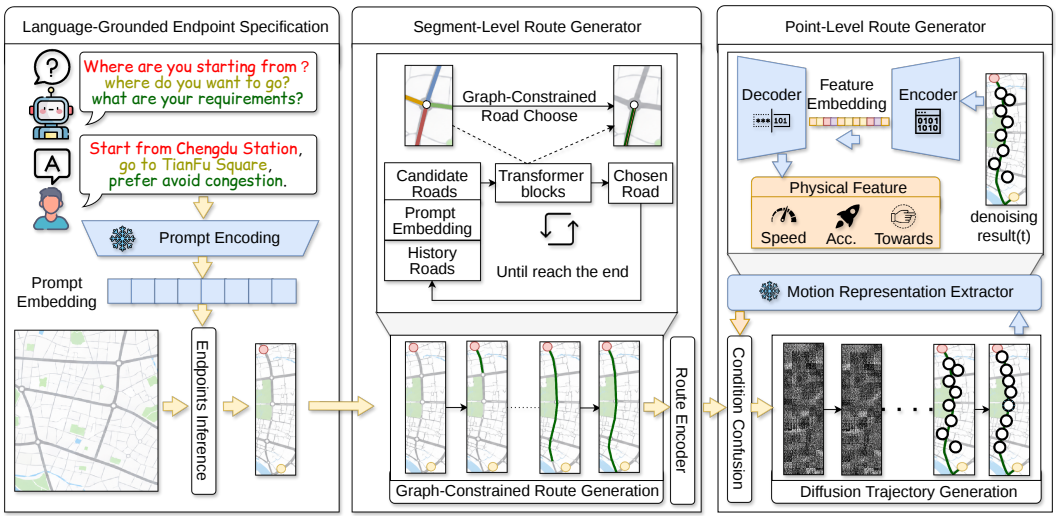


Figure 2: Detailed architecture of PathTraj. An LLM encodes travel instructions to infer reachable endpoints, while a graph-constrained planner constructs connected routes using road topology and destination distances. The planned routes are converted into continuous geometric guidance, where a pretrained motion predictor provides adaptive motion representations during denoising.

# 4 METHOD

As illustrated in **Figure 2**, PathTraj decomposes language-conditioned trajectory generation into a planning stage and a synthesis stage. In the planning stage, PathTraj grounds the instruction in the road network to infer a feasible origin-destination pair. It then decodes a connected route between the two endpoints and converts it into continuous geometric guidance. In the synthesis stage, a diffusion model generates coordinate-level trajectories along the planned route and progressively incorporates local motion representations during denoising.

## 4.1 LANGUAGE-GROUNDED ENDPOINT SPECIFICATION

Natural-language instructions may specify trip endpoints explicitly through segment identifiers or implicitly through functional and spatial cues, such as "from a residential area to a business district". Such heterogeneous descriptions cannot directly initialize route construction. Therefore, we encode the instruction into contextual representations and ground its endpoint semantics in concrete roads.

**Prompt Encoding.** Travel instructions provide semantic cues for endpoint selection and route planning by describing locations, facilities, and travel requirements. Interpreting these cues requires capturing their contextual meanings and semantic relationships to accurately represent the intended journey. To this end, we use the pretrained LLM Qwen3.5-9B (Yang et al., 2025) to encode each instruction and extract contextual token representations:

$$\mathbf{Z}_I = \mathrm{LLM}_{\mathrm{enc}}(I), \quad \mathbf{Z}_I \in \mathbb{R}^{T\times 4096}. \tag{1}$$

Here, $I$ denotes a natural-language instruction consisting of $T$ tokens, and each row of $\mathbf{Z}_I$ corresponds to the contextual representation of one token. These representations capture both token content and semantic relationships within the instruction, providing a basis for endpoint inference and subsequent route generation.

**Endpoint Inference.** To associate endpoint semantics with road segments, we introduce two learnable role queries (Vaswani et al., 2017) that attend to $\mathbf{Z}_I$ and extract origin- and destination-related features, respectively. Each feature is matched against a shared segment embedding table $\mathbf{H}$, yielding distributions $\boldsymbol{\pi}^{\mathrm{o}}$ and $\boldsymbol{\pi}^{\mathrm{d}}$ over road segments. Because the same embeddings are reused during route decoding, endpoint inference and route construction operate on a common road representation. A destination with high probability is not necessarily reachable from the selected origin on the directed road network. We therefore constrain destination selection by graph reachability. Let $\mathcal{R}^{+}(s)$

denote the set of segments, excluding $s$ itself, that can be reached from $s$ through a directed path in $\mathcal{G}$. For an origin $s$ with $\mathcal{R}^+(s) \neq \varnothing$, the destination distribution is renormalized over its reachable set:

$$\widetilde{\pi}^{\mathrm{d}}_j(s) = \begin{cases} \dfrac{\pi^{\mathrm{d}}_j}{\sum_{k\in\mathcal{R}^+(s)} \pi^{\mathrm{d}}_k}, & j \in \mathcal{R}^+(s), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here, $j$ and $k$ index road segments, and membership in $\mathcal{R}^+(s)$ refers to the corresponding segments. During inference, we select the highest-probability origin $\hat{s}_1$ among segments with non-empty reachable sets, and then the highest-probability destination $\hat{s}_{\mathrm{d}}$ under $\widetilde{\boldsymbol{\pi}}^{\mathrm{d}}(\hat{s}_1)$. The pair $(\hat{s}_1, \hat{s}_{\mathrm{d}})$ specifies the initial and target segments for route decoding, and a successfully decoded route $\mathbf{r} = (s_1, \ldots, s_K)$ satisfies $s_1 = \hat{s}_1$ and $s_K = \hat{s}_{\mathrm{d}}$.

### 4.2 Graph-Constrained Route Planning

The inferred endpoints specify where a trip begins and ends, but not how the two segments are connected through the road network. This module decodes a connected route between them and converts it into continuous geometric guidance for trajectory synthesis.

**Destination-Aware Route Decoding.** Route construction must preserve road connectivity while making progress toward the destination. Starting from $s_1 = \hat{s}_1$, the decoder extends the route one segment at a time. At step $k$, only the outgoing neighbors $\mathcal{N}^+(s_k)$ are considered as candidates, so every decoded route is connected by construction. Connectivity alone does not determine which neighbor to follow. To steer decoding toward the destination while respecting the instruction, we provide the decoder with the distance from each segment in the current prefix $\mathbf{r}_{\le k} = (s_1, \ldots, s_k)$ to the destination:

$$\delta_j = \mathrm{dist}_{\mathcal{G}}(s_j, \hat{s}_{\mathrm{d}}), \quad j = 1, \ldots, k, \tag{3}$$

where $\mathrm{dist}_{\mathcal{G}}(\cdot,\cdot)$ denotes the shortest-path distance on $\mathcal{G}$. The decoder $f_\theta$ combines the prefix, the distance sequence $\boldsymbol{\delta}_{\le k}$, and the instruction representations $\mathbf{Z}_I$ to score each candidate, and the scores are normalized within the candidate set:

$$\pi^{\mathrm{route}}_k(s') = \frac{\exp f_\theta(\mathbf{r}_{\le k}, \mathbf{Z}_I, \boldsymbol{\delta}_{\le k}, \mathbf{h}_{s'})}{\sum_{s''\in\mathcal{N}^+(s_k)} \exp f_\theta(\mathbf{r}_{\le k}, \mathbf{Z}_I, \boldsymbol{\delta}_{\le k}, \mathbf{h}_{s''})}, \quad s' \in \mathcal{N}^+(s_k), \tag{4}$$

where $\mathbf{h}_{s'}$ is the embedding of candidate $s'$ from the shared table $\mathbf{H}$. The next segment is selected greedily as $s_{k+1} = \arg\max_{s'} \pi^{\mathrm{route}}_k(s')$. Decoding terminates when the destination is reached ($s_{k+1} = \hat{s}_{\mathrm{d}}$), when no valid successor exists, or when the route exceeds a maximum length $K_{\max}$.

**Continuous Route Representation.** The decoded route is discrete, whereas diffusion operates on continuous coordinates. We therefore concatenate the geometries of the decoded segments in traversal order and parameterize the resulting polyline by normalized cumulative distance, yielding the continuous route representation $\bar{\mathbf{r}}$. This representation retains the spatial layout, orientation, and progression of the planned route, and serves as the geometric condition for trajectory synthesis.

### 4.3 Motion-Guided Trajectory Generation

The continuous route $\bar{\mathbf{r}}$ specifies where a trajectory should travel, but not how the vehicle moves along it. A single route admits many coordinate trajectories that differ in local spatial details and motion patterns. We therefore model this conditional distribution with route-guided diffusion, which follows the planned spatial structure while preserving diversity. During denoising, adaptive motion representations provide additional local guidance as the trajectory becomes progressively structured.

**Route-Guided Diffusion.** Let $\mathbf{x}_0 \in \mathbb{R}^{L\times 2}$ denote a real trajectory. The forward process perturbs it with Gaussian noise (Ho et al., 2020):

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

where $\bar{\alpha}_t$ is the cumulative signal coefficient at timestep $t$ and $\boldsymbol{\epsilon} \in \mathbb{R}^{L\times 2}$. The reverse process is parameterized by a noise predictor conditioned on the planned route, the instruction, and the current motion representations:

$$\hat{\boldsymbol{\epsilon}}_t = \epsilon_\theta(\mathbf{x}_t, t, \bar{\mathbf{r}}, \mathbf{Z}_t, \mathbf{Z}_I)\,. \tag{6}$$

Here, $\bar{\mathbf{r}}$ provides geometric guidance along the planned path, $\mathbf{Z}_I$ encodes the global semantic requirements of the instruction, and $\mathbf{Z}_t$ captures the local motion dynamics of the current state. The network is trained with the standard denoising objective

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\left[\left\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_t\right\|_2^2\right]. \tag{7}$$

At inference, generation starts from pure Gaussian noise and applies the reverse update iteratively. The route and instruction conditions guide every denoising step. Motion conditioning, in contrast, is activated only when $t \leq \tau$, because trajectories at higher noise levels retain too little positional structure for reliable motion estimation. Once activated, $\mathbf{Z}_t$ is re-extracted from the latest $\mathbf{x}_t$ at each step, so that motion guidance adapts to the progressively refined geometry.

**Motion Representation Extraction.** Motion attributes such as speed and heading depend on displacements between adjacent points, which diffusion noise obscures. We therefore design a noise-tolerant extractor conditioned on the planned route and the diffusion timestep. It consists of two streams. The trajectory stream models temporal variations in the noisy positions $\mathbf{x}_t$, and the route stream relates each position to the planned path $\bar{\mathbf{r}}$. An encoder fuses both streams with $t$, adapting feature extraction to the current noise level:

$$\mathbf{H}_t = E_\phi(\mathbf{x}_t, \bar{\mathbf{r}}, t). \tag{8}$$

A motion decoder then maps $\mathbf{H}_t$ to a latent sequence aligned with the trajectory points, which serves as the motion condition in Eq. equation 6. To ensure that this sequence encodes motion-relevant information, an auxiliary head $P_\omega$ predicts per-point motion attributes from it:

$$\mathbf{Z}_t = D_\psi(\mathbf{H}_t, t) \in \mathbb{R}^{L\times d}, \qquad \hat{\mathbf{m}}_t = P_\omega(\mathbf{Z}_t) \in \mathbb{R}^{L\times 4}, \tag{9}$$

where $d$ is the latent dimension and each row of $\hat{\mathbf{m}}_t$ contains speed, acceleration, and the sine and cosine of heading. Encoding heading by its sine and cosine avoids the angular discontinuity at $\pm\pi$. The targets $\mathbf{m}$ are computed from the clean trajectory $\mathbf{x}_0$ by finite differences over the uniform sampling interval. The extractor is pretrained by minimizing $\mathcal{L}_{\text{motion}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\left[\|\hat{\mathbf{m}}_t - \mathbf{m}\|_2^2\right]$ and kept frozen during diffusion training. This supervision encourages $\mathbf{Z}_t$ to capture changes in motion magnitude and direction across adjacent points.

## 5 Experiments

We evaluate PathTraj on real-world DiDi trajectory datasets from Chengdu and Xi'an through baseline comparisons (Zhu et al., 2023; 2024; 2026) and component ablations, examining faithfulness, consistency, and realism (Zhao et al., 2021; Phan-Minh et al., 2020; Gupta et al., 2018; Salzmann et al., 2020). PathTraj achieves better instruction compliance, higher spatio-temporal fidelity, and more realistic motion dynamics than the baselines. These performance advantages remain consistent across trajectory distance levels under natural-language instructions of varying granularity. Ablation studies further indicate that route planning strengthens semantic grounding and spatial structure, while motion guidance improves distributional fidelity and local motion realism.

### 5.1 Faithfulness Analysis

Faithfulness is assessed by examining whether generated trajectories follow the instructions, using semantic agreement and endpoint matching as direct measures. Because the travel requirements expressed in these instructions also influence route selection, comparisons of spatial density, origin–destination patterns, and travel-distance distributions against real data provide complementary evidence about the resulting journeys.

**Instruction compliance. Table 1** compares instruction compliance in Chengdu and Xi'an. PathTraj leads all four metrics in both cities, indicating stronger agreement with instruction semantics, reconstructed attributes, and endpoint requirements. Among the baselines, InsTraj leads TIS in Chengdu, while ControlTraj achieves the highest PRR in both cities. Their lower endpoint scores highlight the difficulty of translating language into spatial decisions. PathTraj addresses this gap through endpoint inference and route planning before trajectory generation. The planned route then guides diffusion, helping preserve these spatial decisions in the generated trajectories. The improvements in PRR, FCS, and FCD are consistent with this approach to grounding travel instructions.

Table 1: Instruction faithfulness comparison of PathTraj and generative baselines

| Model | Chengdu | | | | Xi'an | | | |
|---|---|---|---|---|---|---|---|---|
| | PRR (↑) | TIS (↑) | FCS (↑) | FCD (↑) | PRR (↑) | TIS (↑) | FCS (↑) | FCD (↑) |
| InsTraj | 0.7114 | 0.4813 | 0.6473 | 0.5827 | 0.7349 | 0.4795 | 0.6043 | 0.5463 |
| DiffTraj | 0.7066 | 0.4674 | 0.6087 | 0.5337 | 0.7069 | 0.4794 | 0.6210 | 0.5677 |
| ControlTraj | 0.7329 | 0.4799 | 0.6443 | 0.5793 | 0.7601 | 0.4842 | 0.6620 | 0.6040 |
| **PathTraj (ours)** | **0.7886** | **0.4936** | **0.8447** | **0.7850** | **0.7914** | **0.4903** | **0.8253** | **0.8063** |

**Distributional fidelity.** **Table 2** compares fidelity across spatial distributions, origin–destination patterns, and travel distances. PathTraj achieves the lowest Density, Trip, and Length JSD in both cities, with Length JSD slightly lower than that of the most competitive baseline, ControlTraj. These results indicate that its generated trajectories preserve the spatial characteristics, trip structure, and travel-distance distributions of real journeys. Together with the instruction-based evaluation, these results show that improved semantic agreement is accompanied by faithful reproduction of real travel patterns.

Table 2: Distributional fidelity comparison of PathTraj and generative baselines.

| Model | Chengdu | | | Xi'an | | |
|---|---|---|---|---|---|---|
| | Density (↓) | Trip (↓) | Length (↓) | Density (↓) | Trip (↓) | Length (↓) |
| InsTraj | 0.0558 | 0.2350 | 0.0140 | 0.0577 | 0.2734 | 0.0120 |
| DiffTraj | 0.0550 | 0.2348 | 0.0038 | 0.0656 | 0.2842 | 0.0069 |
| ControlTraj | 0.0466 | 0.2322 | 0.0038 | 0.0536 | 0.2506 | 0.0065 |
| **PathTraj (ours)** | **0.0239** | **0.1665** | **0.0036** | **0.0364** | **0.2351** | **0.0064** |

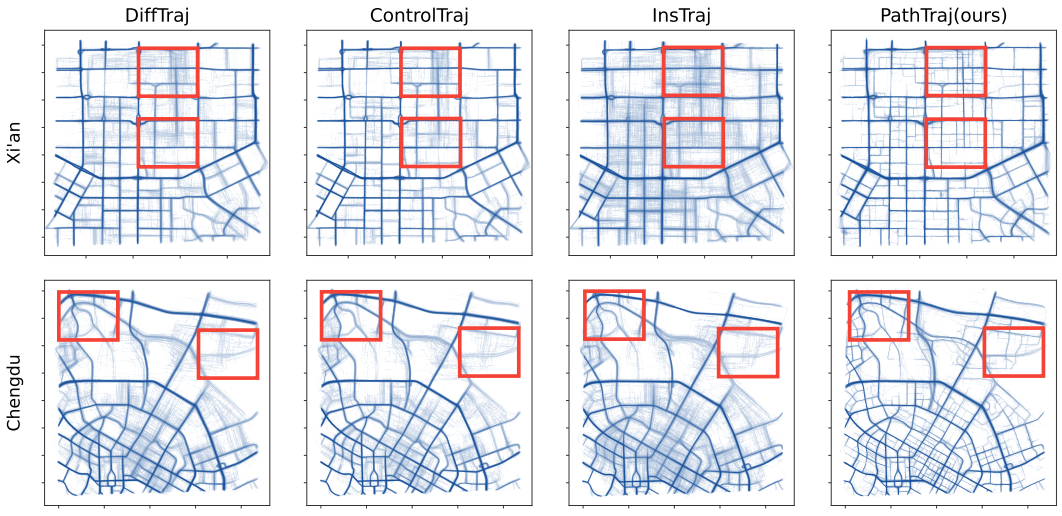


Figure 3: Comparison of generated trajectory distributions in Xi'an (top) and Chengdu (bottom). Red boxes highlight differences in local road corridors and intersection structures.

**Visualization analysis.** The qualitative comparison in **Figure 3** further illustrates how different models reproduce urban spatial structure. All methods capture the major road corridors, but the highlighted regions reveal differences in local trajectory organization. DiffTraj, ControlTraj, and InsTraj exhibit more diffuse traces around secondary streets, whereas PathTraj produces clearer local corridors and more distinct intersections in both cities.

### 5.2 Consistency Analysis

Consistent trajectory generation requires reliable performance across travel distances and levels of instruction detail. The following analysis evaluates this consistency across three distance groups and five prompt granularities. These comparisons examine whether PathTraj maintains its advantages over the baselines in spatial fidelity and instruction compliance as travel distance and semantic guidance vary. Detailed numerical results are provided in **Appendix D.1**.

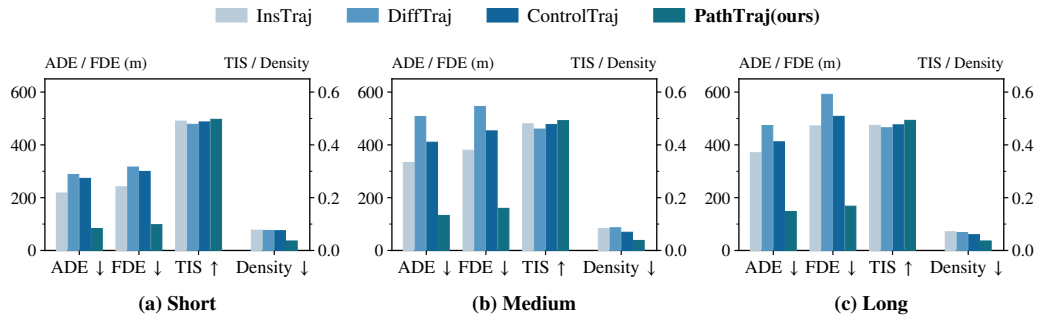


Figure 4: Performance comparison across (a) short, (b) medium, and (c) long distance trajectories.

**Distance Consistency. Figure 4** compares generation performance across short, medium, and long distance trajectories. PathTraj achieves the best results on all four metrics in every distance group, demonstrating its ability to maintain consistent performance advantages as travel distance increases. Although displacement errors increase with travel distance for all models, the increase is smaller for PathTraj than for the baselines. Its semantic similarity and spatial fidelity also remain relatively stable, showing that the model preserves instruction semantics and spatial distributions even over longer journeys. These results support the role of route guidance in maintaining generation consistency. By conditioning point-level generation on planned route geometry and cross-segment continuity, route-guided diffusion provides a stable spatial reference that helps limit deviations while preserving the intended journey.

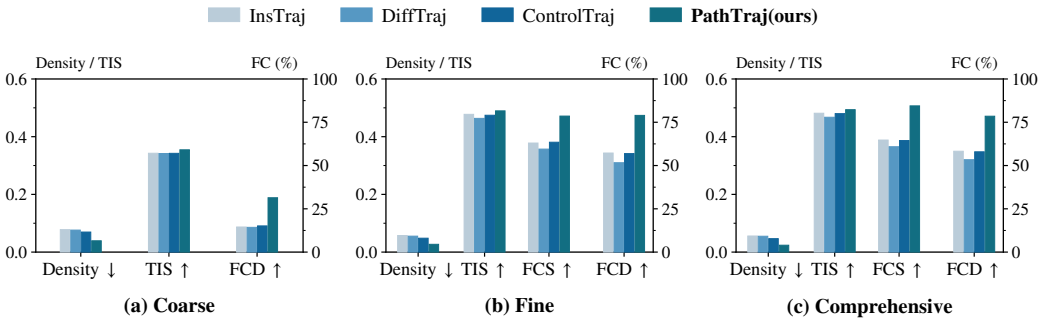


Figure 5: Performance comparison on prompt comprehension

**Prompt Comprehension Consistency.** To examine consistency under varying instruction detail, the evaluation considers five prompt granularities: Coarse, Medium, Fine, Detailed, and Comprehensive. As shown in **Figure 5**, PathTraj achieves the best results across all prompt granularities, demonstrating consistent advantages in instruction compliance and spatial distribution matching. The advantages persist even under coarse prompts, where limited semantic detail makes instruction matching more difficult for all models. This pattern is consistent with the role of explicit endpoint inference and route planning in translating available language cues into spatial decisions that guide subsequent trajectory generation.

### 5.3 Realism Analysis

**Table 3** compares motion realism and positional accuracy in Chengdu and Xi'an. PathTraj achieves the best results on all four metrics in both cities. Compared with the strongest baselines, PathTraj reduces discrepancies in both speed and acceleration distributions, indicating closer agreement with observed movement dynamics. The accompanying reductions in ADE and FDE show that this improvement in motion realism is achieved alongside greater positional accuracy. These gains are consistent with PathTraj's combination of route-guided diffusion and motion representation extraction. The former provides spatial guidance through the planned route, while the latter aligns local motion characteristics with real trajectories.

Table 3: Trajectory generation performance comparison in Chengdu and Xi'an.

| Model | Chengdu | | | | Xi'an | | | |
|---|---|---|---|---|---|---|---|---|
| | ADE (↓) | FDE (↓) | Spd (↓) | Acc. (↓) | ADE (↓) | FDE (↓) | Spd (↓) | Acc. (↓) |
| InsTraj | 429.34 | 364.36 | 0.432 | 0.205 | 500.12 | 522.10 | 0.400 | 0.181 |
| DiffTraj | 537.49 | 484.48 | 0.507 | 0.233 | 482.64 | 481.37 | 0.467 | 0.234 |
| ControlTraj | 464.29 | 420.29 | 0.493 | 0.223 | 485.10 | 444.40 | 0.457 | 0.223 |
| **PathTraj (ours)** | **312.80** | **141.75** | **0.190** | **0.097** | **325.28** | **211.95** | **0.160** | **0.072** |

### 5.4 Ablation Analysis

We conduct three ablations to evaluate the contributions of route planning, motion guidance, and diffusion. As shown in **Table 4**, with complete results provided in **Appendix D.2**, removing each component degrades overall performance, confirming their complementary contributions.

Table 4: Ablation results of PathTraj in Chengdu and Xi'an.

| Model | Chengdu | | | Xi'an | | |
|---|---|---|---|---|---|---|
| | Density(↓) | Trip(↓) | Length(↓) | Density(↓) | Trip(↓) | Length(↓) |
| w/o route planner | 0.1153 | 0.3918 | 0.0124 | 0.0653 | 0.2865 | 0.0076 |
| w/o motion guidance | 0.0348 | 0.2260 | 0.0093 | 0.0443 | 0.2650 | 0.0114 |
| w/o diffusion | 0.0482 | 0.4842 | 0.0284 | 0.0558 | 0.5923 | 0.0229 |
| **PathTraj** | **0.0239** | **0.1665** | **0.0036** | **0.0364** | **0.2351** | **0.0064** |

Specifically, route planning improves spatial and trip distributions, while motion guidance enhances local movement modeling. Removing route planning causes the largest deterioration in Density JSD, highlighting its importance for preserving spatial structure. Removing diffusion produces the largest degradation in Trip and Length JSD, demonstrating its contribution to reproducing real travel patterns. The full model achieves the lowest divergence across all three metrics in both cities.

## 6 Conclusion

This paper introduces PathTraj, a prompt-aware hierarchical framework for generating realistic and controllable trajectories from natural-language instructions. PathTraj integrates language-grounded endpoint specification, graph-constrained route planning, and motion-guided trajectory generation into a unified framework. This hierarchical design translates travel instructions into explicit spatial plans and guides coordinate-level generation while incorporating local movement dynamics. Extensive experiments on real-world trajectory datasets demonstrate improvements in instruction faithfulness, generation consistency, and motion realism. The results further demonstrate consistent performance across varying trajectory lengths and instruction granularities, while ablation studies validate the contributions of the key components. These findings highlight PathTraj's potential for synthesizing realistic mobility trajectories that satisfy diverse travel requirements.

## REFERENCES


Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Can language beat numerical regression? language-based multimodal trajectory prediction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 753–766. IEEE, 2024.

Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R. James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018. doi: 10.1016/j.physrep.2018.01.001. URL https://doi.org/10.1016/j.physrep.2018.01.001.

Chu Cao and Mo Li. Generating mobility trajectories with retained data utility. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2610–2620, 2021. doi: 10.1145/3447548.3467158. URL https://doi.org/10.1145/3447548.3467158.

Ji Cao, Tongya Zheng, Qinghong Guo, Yu Wang, Junshu Dai, Shunyu Liu, Jie Yang, Jie Song, and Mingli Song. Holistic semantic representation for navigational trajectory generation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025a.

Pu Cao, Feng Zhou, Qing Song, and Lu Yang. Controllable generation with text-to-image diffusion models: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Wei-Jer Chang, Wei Zhan, Masayoshi Tomizuka, Manmohan Chandraker, and Francesco Pittaluga. Langtraj: Diffusion model and dataset for language-conditioned trajectory simulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26622–26631. IEEE, 2025.

Xinyu Chen, Jiajie Xu, Rui Zhou, Wei Chen, Junhua Fang, and Chengfei Liu. TrajVAE: A variational autoencoder model for trajectory generation. *Neurocomputing*, 428:332–339, 2021. doi: 10.1016/j.neucom.2020.03.120. URL https://doi.org/10.1016/j.neucom.2020.03.120.

Seongjin Choi, Jiwon Kim, and Hwasoo Yeo. TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning. *Transportation Research Part C: Emerging Technologies*, 128:103091, 2021. doi: 10.1016/j.trc.2021.103091. URL https://doi.org/10.1016/j.trc.2021.103091.

Yves-Alexandre De Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3(1):1376, 2013. doi: 10.1038/srep01376. URL https://doi.org/10.1038/srep01376.

Yuntao Du, Yujia Hu, Zhikun Zhang, Ziquan Fang, Lu Chen, Baihua Zheng, and Yunjun Gao. LDP-Trace: Locally differentially private trajectory synthesis. *Proceedings of the VLDB Endowment*, 16(8):1897–1909, 2023. doi: 10.14778/3594512.3594520. URL https://doi.org/10.14778/3594512.3594520.

Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pp. 1459–1468, 2018.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264. IEEE, 2018.

Mehmet Emre Gursoy, Ling Liu, Stacey Truex, Lei Yu, and Wenqi Wei. Utility-aware synthesis of differentially private and attack-resilient location traces. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 196–211, 2018. doi: 10.1145/3243734.3243741. URL https://doi.org/10.1145/3243734.3243741.

Nils Henke, Shimon Wonsak, Prasenjit Mitra, Michael Nolting, and Nicolas Tempelmeier. CondTraj-GAN: Conditional sequential GAN for generating synthetic vehicle trajectories. In *Advances in Knowledge Discovery and Data Mining*, pp. 79–91, 2023. doi: 10.1007/978-3-031-33377-4_7. URL https://doi.org/10.1007/978-3-031-33377-4_7.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016. doi: 10.1073/pnas.1524261113. URL https://doi.org/10.1073/pnas.1524261113.

Xia Liu, Hanzhou Chen, and Clio Andris. trajgans: Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper). In *Location privacy and security workshop*, pp. 1–7, 2018.

Zhijun Liu, Yiwei Guo, and Kai Yu. Diffvoice: Text-to-speech with latent diffusion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Qingyue Long, Can Rong, Huandong Wang, and Yong Li. One fits all: General mobility trajectory modeling via masked conditional diffusion. *arXiv preprint arXiv:2501.13347*, 2025.

Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44, 2021.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14062–14071. IEEE, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. In *11th International Conference on Geographic Information Science (GIScience 2021) – Part I*, volume 177 of *Leibniz International Proceedings in Informatics*, pp. 12:1–12:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.GIScience.2021.I.12. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.GIScience.2021.I.12.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022. doi: 10.52202/068431-2643. URL https://arxiv.org/abs/2205.11487.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European conference on computer vision*, pp. 683–700. Springer, 2020.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, volume 37, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-RNTraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):7940–7953, 2024. doi: 10.1109/TKDE.2024.3460051. URL https://doi.org/10.1109/TKDE.2024.3460051.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

Yuheng Zhang, Yuan Yuan, Jingtao Ding, Jian Yuan, and Yong Li. Noise matters: Diffusion model-based urban mobility generation with collaborative noise priors. In *Proceedings of the ACM on Web Conference 2025*, pp. 5352–5363, 2025.

Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on robot learning*, pp. 895–904. PMLR, 2021.

Yu Zheng. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology*, 6(3):1–41, 2015. doi: 10.1145/2743025. URL https://doi.org/10.1145/2743025.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–55, 2014. doi: 10.1145/2629592. URL https://doi.org/10.1145/2629592.

Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. DiffTraj: Generating GPS trajectory with diffusion probabilistic model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 65168–65188, 2023. doi: 10.52202/075280-2843. URL https://doi.org/10.52202/075280-2843.

Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. ControlTraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4676–4687, 2024. doi: 10.1145/3637528.3671866. URL https://doi.org/10.1145/3637528.3671866.

Yuanshao Zhu, Yuxuan Liang, Xiangyu Zhao, Liang Han, Xinwei Fang, Xun Zhou, Xuetao Wei, and James Jianqiao Yu. Instraj: Instructing diffusion models with travel intentions to generate real-world trajectories. *arXiv preprint arXiv:2604.04106*, 2026.

# A Data and Instruction Construction

This section describes how trajectories, road networks, and points of interest (POIs) are combined to construct trajectory–instruction pairs. We first introduce the trajectory data and urban context, then explain semantic feature extraction and the construction of prompts at five clarity levels.

## A.1 Trajectory Datasets and Preprocessing

We conduct experiments on two large-scale public taxi trajectory datasets, Taxi-Chengdu and Taxi-Xi'an. Preprocessing removes short or sparsely sampled trajectories and segments individual trips. Each trip is map-matched to the road network to obtain an ordered road-segment sequence. We then resample each coordinate trajectory into $L = 200$ points using linear interpolation; the resulting model input is indexed by $L$.

## A.2 Road Networks and Urban-Function Profiling

**Road networks.** We obtain road-network data for Chengdu and Xi'an from OpenStreetMap (OSM). The resulting directed networks contain 7,598 and 6,019 road segments, respectively. Each segment is associated with its endpoint topology, length, and legal transition relations, providing the graph structure used for map matching and route generation.

Table 5: Road-network and POI statistics used for urban-function profiling.

| City | Directed segments | POI records | Buffer radius |
|---|---|---|---|
| Chengdu | 7,598 | 168,914 | 100 m |
| Xi'an | 6,019 | 134,305 | 100 m |

**POI-based functional labels.** We obtain 168,914 POI records for Chengdu and 134,305 for Xi'an from Amap. Each record contains geographic coordinates and hierarchical service categories. To associate urban functions with the road network, we collect POIs within a 100 m metric buffer around each road geometry. Address and passage metadata are excluded from functional labeling, and the remaining categories are aggregated into urban-function labels. Following the use of POI distributions to characterize urban regions, these labels provide local semantic context for endpoint inference and route analysis. **Table 5** summarizes the available city data, and **Figure 6** visualizes the resulting road-segment functions.

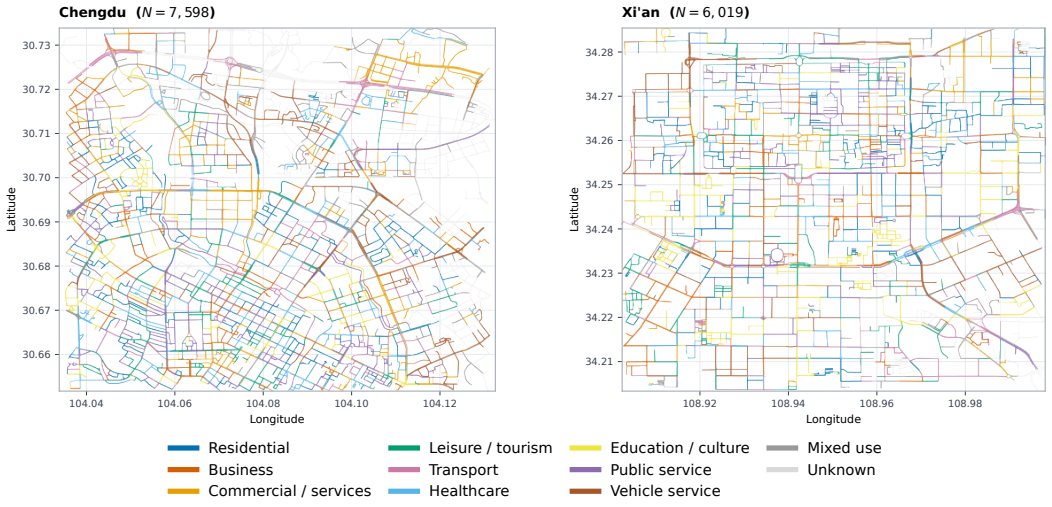


Figure 6: Urban-function labels of road segments in Chengdu (left) and Xi'an (right), derived from POIs within 100 m of road geometries. Colors indicate functional categories.

Table 6: The 16 trajectory-derived semantic attributes and their definitions. Departure period is used as contextual information for purpose inference and prompt rendering.

| Attribute | Calculation or Definition |
|---|---|
| **Trip Intent** | |
| Starting Road Segment | The first valid segment $s_1$ of the repaired route. |
| Destination Road Segment | The last valid segment $s_K$ of the repaired route. |
| Starting Area Type | Dominant urban-function label derived from the POI distribution associated with $s_1$. |
| Destination Area Type | Dominant urban-function label derived from the POI distribution associated with $s_K$. |
| Trip Purpose | Inferred from endpoint functions, departure period, and POI characteristics. |
| **Trip Profile** | |
| Trip Duration | Temporal span of the trip, categorized using the training-set duration tertiles. |
| Trip Distance | Travel distance of the trip, categorized using the training-set distance tertiles. |
| Overall Travel Direction | Direction of the endpoint displacement $(x_{\text{dest}} - x_{\text{start}},\, y_{\text{dest}} - y_{\text{start}})$. |
| **Route Strategy and Behavior** | |
| Directness Preference | Derived from $r_{\text{detour}} = \frac{\sum_{k=1}^{K} \ell(s_k)}{d_{\text{shortest}}}$, comparing route length with the shortest-path distance between its endpoints. |
| Detour Tolerance | Categorized using the same detour ratio $r_{\text{detour}}$. |
| Backtracking Tendency | $b = \frac{1}{K-1} \sum_{k=1}^{K-1} \mathbf{1}[g_{k+1} > g_k]$ for $K > 1$, measuring the fraction of transitions that increase the distance to the destination. |
| Route Windingness | Derived from bearing changes between consecutive route segments. |
| Pace Consistency | $v_i = \frac{(u_{i+1} - u_i)D}{\tau_{i+1} - \tau_i}$ and $CV_v = \frac{\sigma(v)}{\mu(v)}$, using the variation in travel speed. |
| **Road Context** | |
| Preferred Road Type | Inferred by aggregating the connectivity characteristics of traversed road segments. |
| Starting Junction Type | Determined from the out-degree of $s_1$. |
| Ending Junction Type | Determined from the in-degree of $s_K$. |

### A.3 Semantic Feature Extraction

Using the map-matched route and its associated road and POI context, we derive 16 structured attributes in four semantic categories. These attributes summarize the trip's endpoints, travel profile, route behavior, and road context, connecting observed movement with interpretable travel requirements specifies their calculations. The departure period provides additional context for purpose inference and prompt rendering; it is not counted as a separate attribute in the 16-field summary.

**Trip Intent.** We check and repair the map-matched segment sequence against the road graph and use its first and last valid segments as the trip endpoints. The POI distributions associated with these segments determine the Starting Area Type and Destination Area Type. Together with the departure period and endpoint relationship, these functions support inference of Trip Purpose. This category therefore contains the two endpoint identifiers, their two area types, and the inferred purpose.

**Trip Profile.** Trip Duration and Trip Distance are obtained from trajectory metadata and assigned to short, medium, or long categories using training-set tertiles. Overall Travel Direction is determined from the geographic displacement between the endpoints, with a dominance threshold separating cardinal and diagonal directions.

**Route Strategy and Behavior.** We compare the repaired route length with the shortest-path distance to derive Directness Preference and Detour Tolerance. Changes in the remaining distance to the destination characterize Backtracking Tendency, while bearing changes between consecutive segments characterize Route Windingness. Pace Consistency is determined from the coefficient of variation of speeds derived from route progress and timestamps.

**Road Context.** We use segment connectivity statistics along the route to infer Preferred Road Type. The starting segment's out-degree and destination segment's in-degree determine their junc-

tion types, categorized as dead end, through road, simple junction, or complex junction. Preferred Road Type is grouped here together with the two endpoint junction attributes.

For the feature definitions, let $\mathbf{r} = (s_1, \ldots, s_K)$ denote a route containing $K$ segments, with terminal destination segment $s_K$, and let $\ell(s)$ denote segment length. Let $g_k$ be the distance from $s_k$ to the terminal segment $s_K$. For speed estimation, $u_i \in [0, 1]$ denotes normalized cumulative route progress at timestamp $\tau_i$, and $D$ is the total distance corresponding to that progress parameterization. Here, $K$ counts road segments, whereas $L$ counts resampled coordinate points.

### A.4 Natural-Language Prompt Construction

Because the trajectory datasets lack corresponding textual instructions, we use Qwen3.5-9B to convert the extracted semantic attributes into natural-language conditioning prompts. The model is instructed to combine related attributes into coherent descriptions, omit unavailable values, and preserve the supplied semantics without adding unsupported information.

To examine how instruction detail affects trajectory generation, we construct prompts at five cumulative clarity levels. Coarse prompts specify only the destination's urban function, while Medium prompts additionally describe the starting function. Fine prompts add explicit endpoint identifiers, trip purpose, departure context, and trip-profile attributes. Detailed prompts further introduce route preferences and behavioral requirements, including directness, detours, backtracking, windingness, pace consistency, and road preference. Comprehensive prompts complete the description with the starting and ending junction types. **Table 7** presents all five levels for the same trip, illustrating how additional attributes progressively refine the travel requirements.

Table 7: Prompt examples at different clarity levels.

| Clarity | Prompt Instruction |
|---|---|
| **Coarse** | Travel to a business district. |
| **Medium** | Go from a residential area to a business district. |
| **Fine** | Please drive from road segment 120 to road segment 904, starting in a residential area and heading to a business district. Make this a commute-like trip during the morning rush. Use a medium-distance route with a moderate travel time and an overall southeast direction. |
| **Detailed** | Please drive from road segment 120 to road segment 904, starting in a residential area and heading to a business district. Make this a commute-like trip during the morning rush. Use a medium-distance route with a moderate travel time and an overall southeast direction. Choose a reasonably direct route, allow a mild detour, avoid unnecessary backtracking, follow a gently winding path, maintain a generally steady pace, and favor main roads. |
| **Comprehensive** | Please drive from road segment 120 to road segment 904, starting in a residential area and heading to a business district. Make this a commute-like trip during the morning rush. Use a medium-distance route with a moderate travel time and an overall southeast direction. Choose a reasonably direct route, allow a mild detour, avoid unnecessary backtracking, follow a gently winding path, maintain a generally steady pace, and favor main roads. Begin on a through road and finish at a simple junction. |

## B Details of the PathTraj Framework

This section complements PathTraj by clarifying the information passed between the three modules of the framework. **Figure 7** presents language-grounded endpoint specification, graph-constrained route generation, and motion-guided trajectory generation, including the motion representation and auxiliary prediction branches.

The principal configurations are summarized in **Table 8**. Qwen3.5-9B is used as the frozen instruction encoder, and its contextual token representations $\mathbf{Z}_I$ are supplied to the endpoint and route

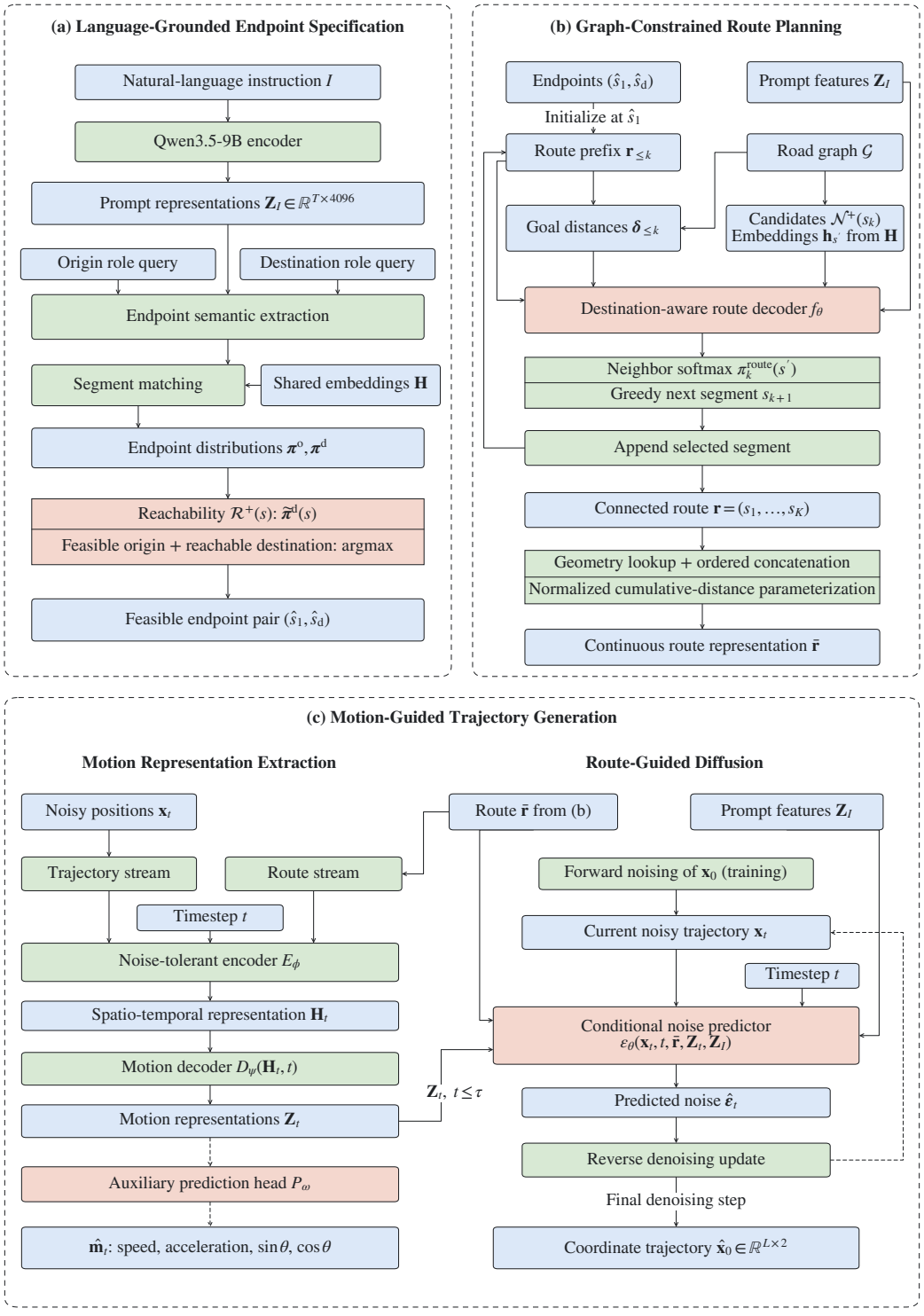


Figure 7: Architecture of PathTraj. (a) Language-grounded endpoint specification identifies a feasible start–destination pair. (b) Graph-constrained route generation produces a connected segment sequence and continuous geometric guidance. (c) Motion-guided trajectory generation combines route, prompt, and latent motion conditions during denoising; an auxiliary head supervises motion representation learning.

Table 8: Compact implementation configuration for PathTraj.

| Component | Parameter | Value |
|---|---|---|
| Instruction encoder | Model | Qwen3.5-9B |
| | Parameters | Frozen |
| | Token hidden size | 4096 |
| Endpoint inference | Role queries | Two learned queries |
| | Prompt projection size | 256 |
| | Segment embedding size | 64 |
| Route planner | Architecture | Transformer decoder |
| | Layers | 4 |
| | Attention heads | 8 |
| | Hidden size | 256 |
| | Conditioning | Destination distance and prompt representations |
| | Successor selection | Highest-probability legal successor |
| Motion module | Hidden size | 256 |
| | Input streams | Trajectory, route, and joint streams |
| | Auxiliary targets | Speed, acceleration, and heading components |
| Diffusion training | Diffusion steps | 500 |
| | Optimizer | AdamW |
| | Learning rate | $2 \times 10^{-4}$ |
| | Batch size | 256 |
| | Epochs | 200 |
| Motion pretraining | Optimizer | AdamW |
| | Learning rate | $2 \times 10^{-4}$ |
| | Batch size | 256 |
| | Epochs | 50 |
| | Update frequency | Every 1–5 denoising steps |
| Inference | Sampler | Route-conditioned DDIM |
| | Denoising steps | 100 |
| | Motion updates | During reverse denoising |

modules. Two learned role queries produce start and destination distributions over the shared road-segment embeddings. Reachability masking is applied before selecting the highest-probability feasible endpoint at each stage.

The route decoder scores legal outgoing neighbors using the route prefix, $\mathbf{Z}_I$, and destination-distance features. The selected segments are concatenated in traversal order and parameterized by cumulative distance to form the continuous route representation $\bar{\mathbf{r}}$. The coordinate denoiser is conditioned on $\bar{\mathbf{r}}$, $\mathbf{Z}_I$, and the trajectory-aligned motion representation $\mathbf{Z}_t$. The motion module encodes the noisy trajectory, route anchors, and diffusion timestep; its auxiliary head predicts speed, acceleration, and heading components.

Training proceeds by fitting the route module, pretraining the motion module, and then optimizing the coordinate denoiser with the route and motion conditions. During inference, endpoint selection is followed by graph-constrained route decoding, route-geometry construction, and conditioned diffusion sampling.

# C Experimental Setup and Evaluation

This section brings together baseline implementations, metric definitions, and training settings. Dataset and prompt construction details are provided in **Appendix A**.

### C.1 Baseline Implementation Details

This section describes the architectural configurations and implementation details of the baseline models. For a consistent comparison, all models are trained on the same trajectory–instruction pairs and use token-level instruction embeddings extracted from Qwen3.5-9B. DiffTraj and ControlTraj are modified to incorporate these semantic representations, while InsTraj integrates them through joint attention.

**DiffTraj.** We follow the original implementation, which employs a U-Net architecture with a ResNet backbone. To incorporate instruction guidance, we replace its original attribute conditioning with token-level instruction embeddings extracted from Qwen3.5-9B. These embeddings are projected into the model's conditioning space and integrated through its adaptive normalization mechanism.

**ControlTraj.** We adopt the original GeoUNet architecture and modify its conditioning mechanism to support natural-language instructions. Specifically, we replace the RoadMAE-derived road-segment embeddings with token-level instruction embeddings from the same Qwen3.5-9B encoder. The embeddings are projected into the required conditioning dimension and supplied as keys and values to GeoUNet's geographic cross-attention layers.

**InsTraj.** We implement the multimodal diffusion Transformer using six MM-DiT blocks with an embedding dimension of 384 and six attention heads. Instruction embeddings extracted from Qwen3.5-9B are projected into instruction tokens and concatenated with trajectory patch tokens. Joint self-attention enables interactions between the two modalities throughout the denoising process.

### C.2 Evaluation Metrics

We evaluate PathTraj in terms of distributional fidelity, instruction faithfulness, and local motion realism. For distributional comparisons, we use Jensen–Shannon divergence (JSD). Given the empirical distributions $P$ and $Q$ of real and generated trajectories, respectively, JSD is defined as

$$\mathrm{JSD}(P, Q) = \frac{1}{2}\,\mathrm{KL}(P\|M) + \frac{1}{2}\,\mathrm{KL}(Q\|M), \qquad M = \frac{P+Q}{2}. \tag{10}$$

Lower JSD values indicate closer agreement between generated and real trajectory distributions. The individual metrics are defined as follows:

**Density.** This metric measures the spatial distribution of trajectory endpoints. We divide each city into a $64 \times 64$ geographic grid, count the endpoints within each cell, and compute the JSD between the normalized real and generated distributions.

**Trip.** This metric measures the joint origin–destination distribution. Using the same $64 \times 64$ grid, we represent each trip as an ordered pair of origin and destination cells. We compute the JSD between the real and generated distributions over these cell pairs, excluding trips with invalid or unavailable endpoints.

**Length.** This metric evaluates whether generated trajectories reproduce the travel-distance distribution of real trips. We calculate each trajectory's total distance by summing the Haversine distances between consecutive observations. We then construct distance histograms for real and generated trajectories and compute their JSD.

**Speed and Acceleration.** These metrics assess the distributional realism of local motion. We compute speed and acceleration from ordered coordinates and timestamps after removing invalid transitions. For each motion attribute, we construct real and generated histograms and compute their JSD.

**TIS.** This metric measures semantic agreement between generated trajectories and their conditioning instructions. We compute the TrajCLIP similarity for each valid trajectory–instruction pair and report the mean. Higher values indicate stronger semantic alignment.

**PRR.** This metric measures how well generated trajectories satisfy the semantic fields specified in their instructions. After map matching, we reconstruct these fields from each generated trajectory

and compare them with the requested values:

$$\mathrm{PRR} = \frac{\sum_j \sum_{f \in \mathcal{F}_j} \mathbf{1}\left[\widehat{f}_j = f_j\right]}{\sum_j |\mathcal{F}_j|}, \tag{11}$$

where $\mathcal{F}_j$ contains the requested fields that are available and unmasked for instance $j$, and $\widehat{f}_j$ denotes the reconstructed value. PRR pools field counts across conditions and is reported as a real-valued proportion in $[0, 1]$; higher values indicate stronger field-level instruction agreement.

**FCS and FCD.** These metrics measure whether generated origins and destinations satisfy the urban-function requirements specified in the instructions. For endpoint type $e \in \{\mathrm{start}, \mathrm{dest}\}$, we compute

$$\mathrm{FC}_e = \frac{1}{|\mathcal{V}_e|} \sum_{j \in \mathcal{V}_e} \mathbf{1}[\widehat{y}_{j,e} = y_{j,e}], \tag{12}$$

where $\mathcal{V}_e$ is the valid evaluation set, and $\widehat{y}_{j,e}$ and $y_{j,e}$ denote the generated endpoint's urban-function label and the requested label, respectively. FCS and FCD correspond to $\mathrm{FC}_{\mathrm{start}}$ and $\mathrm{FC}_{\mathrm{dest}}$. Higher values indicate stronger endpoint compliance.

# D Supplementary Experimental Results

This section presents the detailed numerical results supporting the consistency analysis in the main text. Component ablations and qualitative visualizations are organized as separate subsections to distinguish their experimental purposes.

## D.1 Consistency Analysis

The results complement **Section 5.2** by comparing three travel-distance groups and five prompt clarity levels in both cities. The distance comparison examines spatial accuracy and semantic alignment over increasingly long journeys, while the clarity comparison examines performance under varying amounts of semantic guidance. Here, consistency refers to sustained advantages over the baselines across these settings, rather than identical scores across groups.

**Distance Consistency. Table 9** supplements **Figure 4** with results for each city and distance group. PathTraj achieves the lowest ADE, FDE, and Density and the highest TIS in every group. Its displacement errors increase on longer trips, but its advantage over the baselines persists, while TIS and Density vary within comparatively narrow ranges. These findings are consistent with route guidance providing a spatial reference that helps maintain semantic alignment and distributional fidelity over longer journeys.

**Prompt Comprehension Consistency. Table 10** provides the numerical results complementing **Figure 5**. PathTraj consistently achieves leading performance across prompt clarity levels in both cities. Its advantages under Coarse and Medium prompts suggest that endpoint inference and route planning help translate limited semantic cues into spatial guidance. At Fine, explicit endpoint identifiers become available and compliance scores increase substantially relative to Medium. Further detail does not improve every metric monotonically, as richer prompts introduce additional requirements. The sustained advantage across clarity levels therefore supports the model's consistency under varying instruction detail.

## D.2 Ablation Studies

All ablation comparisons are conducted with Comprehensive-level prompts, so every evaluated semantic field is available under the same instruction condition.

**Instruction faithfulness. Table 11** shows that the complete PathTraj configuration provides the strongest agreement with the requested semantic fields across both cities. Removing route planning or diffusion causes larger declines in PRR and endpoint compliance, indicating that both explicit route structure and denoising are important for translating instruction semantics into a realizable path. Removing motion guidance produces a smaller but consistent reduction in semantic compliance. The corresponding TIS changes are comparatively modest, suggesting that global text–trajectory similarity is less sensitive than the field-level and endpoint measures.

Table 9: Performance comparison across trajectory distance levels in Chengdu and Xi'an.

| City | Dist. level | Model | ADE (↓) | FDE (↓) | TIS (↑) | Density (↓) |
|---|---|---|---|---|---|---|
| Chengdu | Short | InsTraj | 248.84 | 241.38 | 0.4896 | 0.0766 |
| | | DiffTraj | 330.91 | 316.10 | 0.4781 | 0.0756 |
| | | ControlTraj | 270.46 | 299.46 | 0.4866 | 0.0747 |
| | | **PathTraj (ours)** | **156.10** | **97.77** | **0.4965** | **0.0361** |
| | Medium | InsTraj | 322.69 | 379.80 | 0.4805 | 0.0828 |
| | | DiffTraj | 540.99 | 545.78 | 0.4596 | 0.0864 |
| | | ControlTraj | 436.06 | 453.11 | 0.4775 | 0.0695 |
| | | **PathTraj (ours)** | **286.89** | **159.62** | **0.4921** | **0.0378** |
| | Long | InsTraj | 716.49 | 471.90 | 0.4738 | 0.0709 |
| | | DiffTraj | 740.57 | 591.56 | 0.4649 | 0.0676 |
| | | ControlTraj | 686.34 | 508.30 | 0.4759 | 0.0600 |
| | | **PathTraj (ours)** | **495.41** | **167.86** | **0.4926** | **0.0355** |
| Xi'an | Short | InsTraj | 327.07 | 355.49 | 0.4866 | 0.1067 |
| | | DiffTraj | 327.55 | 354.01 | 0.4799 | 0.0975 |
| | | ControlTraj | 371.61 | 314.49 | 0.4849 | 0.0835 |
| | | **PathTraj (ours)** | **210.16** | **112.98** | **0.4916** | **0.0539** |
| | Medium | InsTraj | 460.65 | 530.65 | 0.4789 | 0.0963 |
| | | DiffTraj | 441.92 | 497.67 | 0.4792 | 0.0832 |
| | | ControlTraj | 446.94 | 440.10 | 0.4817 | 0.0700 |
| | | **PathTraj (ours)** | **296.97** | **254.88** | **0.4878** | **0.0465** |
| | Long | InsTraj | 712.63 | 680.17 | 0.4729 | 0.0868 |
| | | DiffTraj | 678.46 | 592.43 | 0.4790 | 0.0694 |
| | | ControlTraj | 636.76 | 578.60 | 0.4861 | 0.0583 |
| | | **PathTraj (ours)** | **468.71** | **267.99** | **0.4915** | **0.0499** |

**Fine-motion realism.** The complete configuration also achieves the closest agreement with the real speed and acceleration distributions. Removing motion guidance worsens both discrepancies, supporting the role of trajectory-aligned motion representations in modeling local movement changes. Removing the route planner likewise degrades fine-motion realism, showing that global geometric context benefits local kinematics. The largest deterioration follows removal of diffusion, consistent across both motion statistics and both cities, which emphasizes that denoising is necessary for converting route and motion conditions into smooth coordinate sequences.

Table 10: Performance under different prompt clarity levels in Chengdu and Xi'an. Best results at each clarity level are highlighted in bold.

| City | Clarity | Model | Density (↓) | Trip (↓) | Length (↓) | PRR (↑) | TIS (↑) | FCS (↑) | FCD (↑) |
|---|---|---|---|---|---|---|---|---|---|
| Chengdu | Coarse | InsTraj | 0.0778 | 0.3461 | 0.0166 | 0.1450 | 0.3426 | N/A | 0.1450 |
| | | DiffTraj | 0.0763 | 0.3431 | 0.0111 | 0.1430 | 0.3417 | N/A | 0.1430 |
| | | ControlTraj | 0.0695 | 0.3352 | 0.0118 | 0.1513 | 0.3423 | N/A | 0.1513 |
| | | **PathTraj** | **0.0392** | **0.2680** | **0.0110** | **0.3140** | **0.3545** | N/A | **0.3140** |
| | Medium | InsTraj | 0.0744 | 0.3083 | 0.0181 | 0.1465 | 0.3554 | 0.1520 | 0.1373 |
| | | DiffTraj | 0.0701 | 0.3117 | 0.0102 | 0.1519 | 0.3558 | 0.1560 | 0.1427 |
| | | ControlTraj | 0.0627 | 0.2983 | **0.0097** | 0.1526 | 0.3583 | 0.1497 | 0.1357 |
| | | **PathTraj** | **0.0345** | **0.2285** | **0.0097** | **0.2980** | **0.3720** | **0.3220** | **0.3060** |
| | Fine | InsTraj | 0.0574 | 0.2437 | 0.0149 | 0.7064 | 0.4775 | 0.6300 | 0.5723 |
| | | DiffTraj | 0.0553 | 0.2405 | 0.0036 | 0.6888 | 0.4635 | 0.5950 | 0.5163 |
| | | ControlTraj | 0.0482 | 0.2303 | 0.0038 | 0.7185 | 0.4743 | 0.6347 | 0.5690 |
| | | **PathTraj** | **0.0270** | **0.1859** | **0.0033** | **0.8434** | **0.4895** | **0.7857** | **0.7897** |
| | Detailed | InsTraj | 0.0565 | 0.2432 | 0.0148 | 0.6529 | 0.4806 | 0.6427 | 0.5797 |
| | | DiffTraj | 0.0551 | 0.2395 | 0.0041 | 0.6561 | 0.4669 | 0.6077 | 0.5393 |
| | | ControlTraj | 0.0479 | 0.2337 | 0.0041 | 0.6840 | 0.4793 | 0.6457 | 0.5630 |
| | | **PathTraj** | **0.0250** | **0.1714** | **0.0040** | **0.7342** | **0.4922** | **0.8423** | **0.7873** |
| | Comprehensive | InsTraj | 0.0558 | 0.2350 | 0.0140 | 0.7114 | 0.4813 | 0.6473 | 0.5827 |
| | | DiffTraj | 0.0550 | 0.2348 | 0.0038 | 0.7066 | 0.4674 | 0.6087 | 0.5337 |
| | | ControlTraj | 0.0466 | 0.2322 | 0.0038 | 0.7329 | 0.4799 | 0.6443 | 0.5793 |
| | | **PathTraj** | **0.0239** | **0.1665** | **0.0036** | **0.7886** | **0.4936** | **0.8447** | **0.7850** |
| Xi'an | Coarse | InsTraj | 0.0888 | 0.3678 | 0.0108 | 0.1353 | 0.3420 | N/A | 0.1353 |
| | | DiffTraj | 0.0759 | 0.3342 | 0.0084 | 0.1250 | 0.3423 | N/A | 0.1250 |
| | | ControlTraj | 0.0650 | 0.3081 | 0.0088 | 0.1260 | 0.3438 | N/A | 0.1260 |
| | | **PathTraj** | **0.0491** | **0.3027** | **0.0047** | **0.3102** | **0.3469** | N/A | **0.3102** |
| | Medium | InsTraj | 0.0901 | 0.3574 | 0.0114 | 0.1455 | 0.3544 | 0.1337 | 0.1320 |
| | | DiffTraj | 0.0741 | 0.3189 | 0.0105 | 0.1381 | 0.3574 | 0.1517 | 0.1360 |
| | | ControlTraj | 0.0631 | 0.2937 | 0.0101 | 0.1467 | 0.3621 | 0.1537 | 0.1327 |
| | | **PathTraj** | **0.0461** | **0.2855** | **0.0048** | **0.2531** | **0.3653** | **0.3007** | **0.3149** |
| | Fine | InsTraj | 0.0802 | 0.3113 | 0.0133 | 0.7449 | 0.4812 | 0.5983 | 0.5443 |
| | | DiffTraj | 0.0633 | 0.2806 | 0.0065 | 0.7009 | 0.4848 | 0.6340 | 0.5890 |
| | | ControlTraj | 0.0529 | 0.2491 | 0.0068 | 0.7765 | 0.4918 | 0.6803 | 0.6103 |
| | | **PathTraj** | **0.0400** | **0.2386** | **0.0051** | **0.8056** | **0.4955** | **0.7883** | **0.8323** |
| | Detailed | InsTraj | 0.0774 | 0.3001 | 0.0117 | 0.7006 | 0.4872 | 0.6157 | 0.5610 |
| | | DiffTraj | 0.0632 | 0.2744 | 0.0060 | 0.6622 | 0.4878 | 0.6403 | 0.5947 |
| | | ControlTraj | 0.0525 | 0.2487 | 0.0058 | 0.7229 | 0.4933 | 0.6790 | 0.6230 |
| | | **PathTraj** | **0.0368** | **0.2330** | **0.0054** | **0.7539** | **0.4979** | **0.8402** | **0.8334** |
| | Comprehensive | InsTraj | 0.0577 | 0.2734 | 0.0120 | 0.7349 | 0.4795 | 0.6043 | 0.5463 |
| | | DiffTraj | 0.0656 | 0.2842 | 0.0069 | 0.7069 | 0.4794 | 0.6210 | 0.5677 |
| | | ControlTraj | 0.0536 | 0.2506 | 0.0065 | 0.7601 | 0.4842 | 0.6620 | 0.6040 |
| | | **PathTraj** | **0.0364** | **0.2351** | **0.0064** | **0.7914** | **0.4903** | **0.8253** | **0.8063** |

**Ablation summary.** Taken together, the results confirm that the three components contribute complementary information. Route planning supplies global spatial and semantic constraints, motion guidance refines local dynamics, and diffusion integrates these conditions during coordinate generation. The consistent trends across Chengdu and Xi'an indicate that the improvements are not tied to a single city or evaluation aspect.

### D.3 Route-Level Compliance and Progression Analysis

Existing distributional, instruction-faithfulness, and fine-motion metrics do not directly assess whether each generated trajectory follows its intended route. We therefore conduct a route-level

Table 11: Ablation study of PathTraj in Chengdu and Xi'an.

| City | Model | Density (↓) | Trip (↓) | Length (↓) | PRR (↑) | TIS (↑) | FCS (↑) | FCD (↑) | Spd. (↓) | Acc. (↓) |
|---|---|---|---|---|---|---|---|---|---|---|
| Chengdu | w/o route planner | 0.1153 | 0.3918 | 0.0124 | 0.7021 | 0.4701 | 0.6031 | 0.5402 | 0.503 | 0.235 |
| | w/o motion guidance | 0.0348 | 0.2260 | 0.0093 | 0.7531 | 0.4933 | 0.8333 | 0.7791 | 0.220 | 0.118 |
| | w/o diffusion | 0.0482 | 0.4842 | 0.0284 | 0.7175 | 0.4851 | 0.5657 | 0.6237 | 0.555 | 0.298 |
| | **PathTraj (ours)** | **0.0239** | **0.1665** | **0.0036** | **0.7886** | **0.4936** | **0.8447** | **0.7850** | **0.190** | **0.097** |
| Xi'an | w/o route planner | 0.0653 | 0.2865 | 0.0076 | 0.7401 | 0.4796 | 0.6110 | 0.5617 | 0.464 | 0.236 |
| | w/o motion guidance | 0.0443 | 0.2650 | 0.0114 | 0.7531 | 0.4834 | 0.8214 | 0.8011 | 0.166 | 0.102 |
| | w/o diffusion | 0.0558 | 0.5923 | 0.0229 | 0.7237 | 0.4842 | 0.6710 | 0.5847 | 0.549 | 0.341 |
| | **PathTraj (ours)** | **0.0364** | **0.2351** | **0.0064** | **0.7914** | **0.4903** | **0.8253** | **0.8063** | **0.160** | **0.072** |

analysis to evaluate geometric adherence and along-route progression, providing complementary evidence for faithful and consistent trajectory generation.

**Evaluation setup.** For each paired evaluation instance, we concatenate the geometries of the ground-truth segment sequence in traversal order to form a continuous intended-route polyline. Generated and ground-truth coordinates are projected independently onto their nearest segments of this polyline. The projected positions are then converted into cumulative route progress measured in meters from the route origin. Comparisons use corresponding points in the generated and ground-truth trajectories without enforcing monotonic progress during projection. For progression-based metrics, points are retained only when both projections are within 50 m of the intended route and are unambiguous. A projection is considered ambiguous when nearly equidistant route segments imply progress values that differ by at least 20 m.

**Route compliance metrics.** Route distance measures geometric adherence by averaging the shortest distance from generated points to the intended-route polyline. Progress MAE measures along-route consistency by comparing generated and ground-truth cumulative progress at corresponding valid points. Normalized MAE divides Progress MAE by the total intended-route length, allowing comparison across trips of different lengths. Endpoint error measures the difference between the final generated and ground-truth progress values, capturing route completion accuracy beyond direct coordinate displacement. Lower values indicate better route compliance and progression consistency for all four metrics.

**Quantitative results. Table 12** reports the route-level results. PathTraj achieves the lowest Route distance, Progress MAE, Normalized MAE, and Endpoint error in both cities. These results indicate that its generated trajectories remain closer to the intended route and more accurately reproduce the ground-truth trajectory's progression along that route than the baselines.

Table 12: Route-level compliance and progression comparison in Chengdu and Xi'an.

| City | Model | Route distance (↓) | Progress (↓) | Normalized (↓) | Endpoint (↓) |
|---|---|---|---|---|---|
| Chengdu | **PathTraj** | **44.574** | **233.164** | **0.0761** | **69.772** |
| | InsTraj | 182.015 | 285.022 | 0.0917 | 195.973 |
| | DiffTraj | 265.799 | 302.513 | 0.0965 | 250.661 |
| | ControlTraj | 225.282 | 291.797 | 0.0936 | 219.845 |
| Xi'an | **PathTraj** | **75.090** | **237.316** | **0.0708** | **104.548** |
| | InsTraj | 273.103 | 320.966 | 0.0931 | 284.427 |
| | DiffTraj | 251.398 | 309.815 | 0.0899 | 281.018 |
| | ControlTraj | 221.812 | 304.483 | 0.0881 | 271.858 |

**Qualitative analysis. Figures 8** and **9** provide qualitative comparisons for Xi'an and Chengdu, respectively. The map views show that PathTraj more closely follows the intended route, while the route-progress plots show closer agreement with the ground-truth progression over the trajectory horizon. Together, the quantitative and qualitative results support stronger individual-trajectory route compliance, complementing the population-level fidelity, instruction-faithfulness, and fine-motion evaluations.

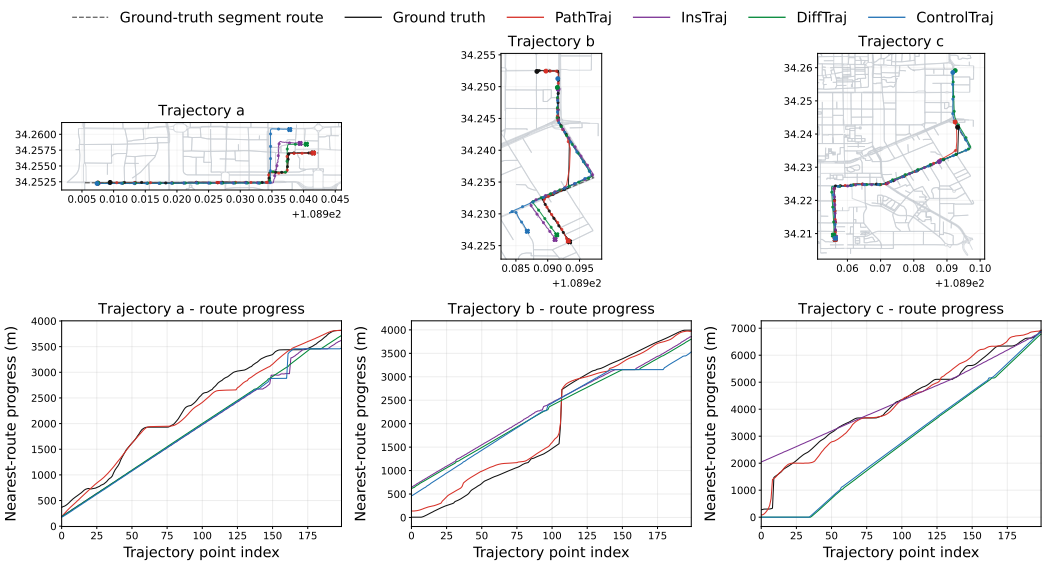


Figure 8: Qualitative route-level compliance and progression comparison in Xi'an. Each row shows a reference route and the corresponding generated trajectories.

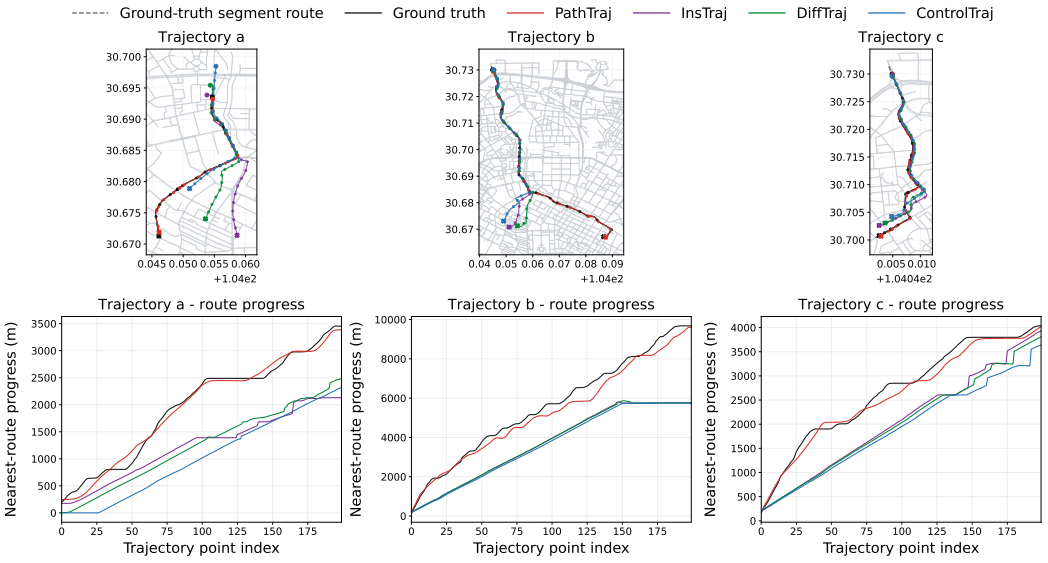


Figure 9: Qualitative route-level compliance and progression comparison in Chengdu.

## E AI USE STATEMENT

Qwen3.5-9B is an explicit component of the proposed PathTraj framework. It is used to encode natural-language travel instructions into contextual semantic representations and to render structured trajectory attributes into natural-language prompts at five cumulative clarity levels. These model outputs are generated under predefined input formats and prompt-construction rules, and are subsequently used as conditioning information for endpoint inference and trajectory generation.

During manuscript preparation, generative AI tools were also used to improve grammar, wording, and readability in selected passages. All AI-assisted revisions were manually checked by the authors to ensure that they preserve the intended methodology, experimental descriptions, numerical results, and scientific conclusions.

## F ETHICS STATEMENT

This study uses publicly available taxi trajectory benchmarks from Chengdu and Xi'an, together with road-network data from OpenStreetMap and points-of-interest data from Amap. We do not collect new human-subject data, conduct user studies, or access personally identifying information. The trajectory records are used for map matching, aggregate mobility analysis, and synthetic trajectory generation rather than for identifying or profiling individual travelers.

Because mobility traces may reveal sensitive information about travel routines and frequently visited locations, we take precautions to limit privacy risks. We do not attempt to re-identify users, infer personal identities, or release raw trajectory records beyond the terms permitted by the original data providers. The extracted semantic attributes are used at the level of road segments, urban functions, trip profiles, and route behaviors. The generated trajectories are intended for mobility research, model development, and evaluation, and should not be used for unauthorized surveillance, individual tracking, or other applications that could compromise personal privacy.

## G REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the paper describes the task formulation, the PathTraj architecture, and the training and inference procedures. **Appendix A** documents the trajectory datasets, preprocessing steps, road networks, urban-function labels, semantic feature extraction process, and natural-language prompt construction. **Appendix B** provides additional details of Language-Grounded Endpoint Specification, Graph-Constrained Route Generation, and Motion-Guided Trajectory Generation. **Appendix C** specifies the baseline configurations, evaluation metrics, and implementation settings. **Appendix D** reports detailed numerical results for distance consistency, prompt-comprehension consistency, ablation studies, and qualitative visualization. For each trajectory, we specify the map-matching procedure, the resampling length, the 16 structured semantic attributes, and the five cumulative prompt-clarity levels. The materials supporting this study are publicly available at `https://anonymous.4open.science/r/PathTraj-EC63`.