DRIVESKEL-LM: LLM-BASED DRIVING ROUTE GENERATION VIA ROAD-NAME-AWARE SKELETONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work in transit planning shows that language models can generate structured routes from discrete network tokens. Extending this to driving is challenging because urban road networks are denser and more highly branched, and driving routes can contain long intersection sequences: decoding every intersection lengthens autoregressive generation, so an early error can cause missed road-change decisions, endpoint attachment failures, or disconnected routes. We introduce DriveSkel-LM, a road-name-aware framework for reliable driving route planning. Given a request, DriveSkel-LM predicts a compact skeleton of the origin, destination, and globally unique intersections at road-name changes, with aligned road names, preserving route-level decisions while omitting unchanged-road intersections. We adapt Qwen3-4B through intersection tokens and continued pre-training on anonymized navigation routes, road connectivity, and POI descriptions. Uniquely specified segments are reconstructed deterministically without graph search, while A^* is used only for locally recoverable gaps. On temporally separated Beijing data, we evaluate DriveSkel-LM on optimal routing, road-name-constrained routing, and en-route search; on optimal routing it produces more reliable routes than full intersection-level generation, demonstrating the potential of LLM-based driving route planning.

1 INTRODUCTION

Driving route planning relies on detailed road graphs, candidate search, and task-specific ranking; although effective, such systems require extensive map infrastructure and specialized engineering and provide only limited native language understanding for customized requests, whereas navigation-route records offer supervision for learning high-level routing decisions.

Token-based route generation has recently been demonstrated for public transit planning. TransitLM generates transit routes from a query and origin-destination information without map queries at inference time (Guo et al., 2026), producing line and station-ID sequences that let the route be reconstructed from the prediction alone. Transit networks are also constrained by fixed operating lines, where a line name with boarding and alighting stations usually identifies a segment. This raises the question of whether the same approach can support driving route planning.

Driving cannot be addressed by simply replacing station tokens with intersection tokens. Urban road networks are denser and more highly branched, especially in large metropolitan areas. The most direct adaptation generates every intersection, including many on the same road, creating a long autoregressive sequence in which an early error can shift later predictions, miss a route-changing intersection, prevent destination attachment, or produce consecutive intersections that are not locally connected—leaving a seemingly correct route unusable.

These limitations motivate a shorter representation that preserves the route decisions needed for recovery (Figure 1). Road names alone are too coarse, since a single road can contain many intersections and entry or exit points, while assigning a unique token to every road segment resolves this ambiguity at the cost of a much larger vocabulary and less sharing across routes. We instead pair globally unique intersection tokens with road names: the intersections specify route-decision boundaries, and the aligned road name identifies the road traversed between them.

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.

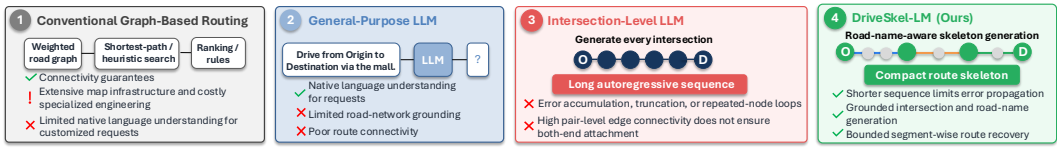

Figure 1: Comparison of driving route planning paradigms.

We present DriveSkel-LM (Figure 2), a framework that generates a compact, road-name-aware intersection skeleton: the origin, destination, and intersections where the road name changes, with aligned road names, omitting unchanged-road intersections while retaining what deterministic recovery needs. When adjacent skeleton intersections form a locally recoverable gap, DriveSkel-LM invokes A^* (Hart et al., 1968) only for that interval, separating high-level route decision generation from the limited graph search needed to resolve local gaps. Unlike the map-free decoding of TransitLM, DriveSkel-LM's recovery still reads the static road graph; what it removes from the online step is the per-intersection autoregressive generation, not the use of map data.

We instantiate DriveSkel-LM with Qwen3-4B (Yang et al., 2025), augment its vocabulary with globally unique intersection tokens, and continue pre-training on anonymized navigation-route records, road connectivity, and POI information from Beijing, then fine-tune separate models for the three tasks. Evaluation uses data collected after the pre-training period, providing temporal separation between training and testing.

Our contributions are:

- We introduce a road-name-aware intersection skeleton that adapts token-based route generation to dense and highly branched driving networks.
- We develop a route reconstruction procedure that deterministically recovers uniquely specified segments and uses A^* only for locally recoverable gaps.
- We evaluate DriveSkel-LM on three driving planning tasks. On optimal routing, it produces more reliable routes than full intersection-level generation, supporting the potential of LLM-based driving route planning.

2 Related Work

2.1 Map-Based Route Planning and Route Recommendation

Conventional driving systems generate feasible route candidates by searching a weighted road graph. Classical shortest-path and heuristic search algorithms provide connectivity guarantees, while hierarchical preprocessing accelerates large-scale routing (Dijkstra, 2022; Hart et al., 1968; Sanders & Schultes, 2006; Geisberger et al., 2008; Goldberg & Harrelson, 2005; Goldberg et al., 2006; Geisberger et al., 2012; Dibbelt et al., 2014; Bast et al., 2016). These candidates are then ranked using route attributes and user preferences. Comprehensive Comparison Network (CCN) improves this ranking stage by reasoning over the non-overlapping segments of candidate route pairs (Guo et al., 2025). It therefore selects from routes supplied by a routing engine rather than generating one; DriveSkel-LM addresses a complementary task, predicting road-name-change intersections and road names from a request and reconstructing the route, with graph search reserved for unresolved segments.

2.2 Data-Driven Mobility Modeling and Route Learning

Data-driven mobility research learns from historical trajectories and travel records for route recommendation, mobility prediction, trajectory generation, and travel-time estimation (Yuan et al., 2010; Feng et al., 2018; Wang et al., 2018a; 2025a; Newson & Krumm, 2009; Lou et al., 2009; Wang et al., 2018b; Lian et al., 2014; Cheng et al., 2013; Kipf & Welling, 2017; Veličković et al., 2018). Recent foundation and representation learning methods further broaden this paradigm across mobility trajectories and road networks (Han et al., 2025; Long et al., 2025; Zhao et al., 2025; Zhou

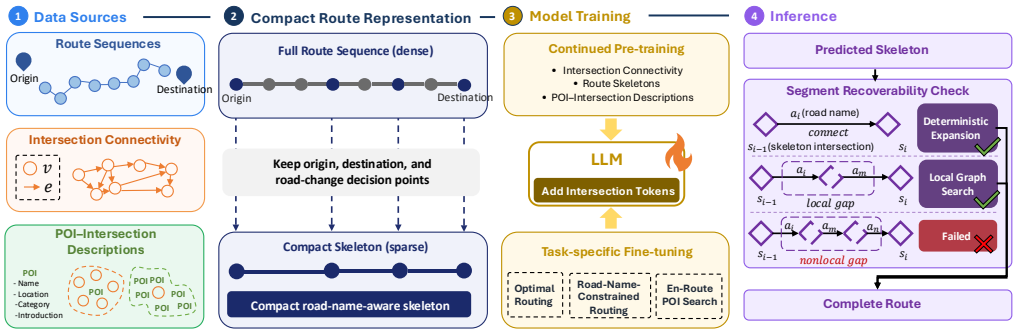

Figure 2: Overview of DriveSkel-LM. (1) Route sequences, intersection connectivity, and POI–intersection descriptions form the corpus. (2) Full routes are compressed into skeletons. (3) CPT and task-specific SFT support three routing tasks. (4) At inference, predicted skeleton segments are checked and recovered into a complete route.

et al., 2024; 2025; Wang et al., 2025b). These methods span mobility prediction, travel-time estimation, trajectory generation, and graph-search-based routing (e.g., landmark-graph routing in T-Drive and learned-cost shortest-path search in GTG), but their objectives differ from generating a structured driving route conditioned on a natural-language request and origin-destination information. DriveSkel-LM instead learns from navigation-route records and predicts the road-change decisions that define a route, targeting route planning rather than trajectory prediction, travel-time estimation, or route scoring.

2.3 Language Models for Travel and Route Planning

Large language models are increasingly applied to travel planning through tool use and external information retrieval (Xie et al., 2024; Shao et al., 2026; Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023; Ouyang et al., 2022; Wei et al., 2022; Yao et al., 2023b;a). TransitLM is the closest prior setting for language-model-based route generation (Guo et al., 2026), but its map-free decoding suits transit networks with fixed operating lines, whereas road networks are denser and more highly branched. DriveSkel-LM instead generates road-name-change intersections and road names as a compact skeleton, then uses local A^* only for unresolved intervals where the skeleton does not uniquely determine the route.

3 Method

3.1 Task Formulation and Design Objective

We model the road network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ is a globally unique intersection and each edge $e \in \mathcal{E}$ is a traversable road connection with name $n(e)$. Given a request $x = (q, o, d, c)$—a natural-language query q, origin and destination coordinates o and d, and optional task-specific constraints c—the goal is an executable route $\mathbf{r} = (v_0, v_1, \ldots, v_N)$ connecting o to d with $(v_{j-1}, v_j) \in \mathcal{E}$ for all j. The endpoints v_0 and v_N are generated access nodes and need not coincide exactly with o and d.

Directly generating $\mathbf{r}$ requires predicting every intersection, including many consecutive ones on the same road. DriveSkel-LM instead predicts a road-name-aware skeleton $\mathbf{s} = (s_0, s_1, \ldots, s_M)$ with aligned road names $\mathbf{a} = (a_1, \ldots, a_M)$, where each s_i is a route endpoint or an intersection where the road name changes and a_i identifies the road traversed from s_{i-1} to s_i. The pair $(\mathbf{s}, \mathbf{a})$ retains the route-level decisions needed to recover $\mathbf{r}$: we predict it from x, then recover uniquely specified segments deterministically and use local graph search only for locally recoverable gaps.

Table 1: Optimal-routing comparison between full-intersection generation ($\mathrm{CPT_F} + \mathrm{SFT_F}$) and DriveSkel-LM. †The full-sequence model has no skeleton intervals, so its Route Connectivity is the pair-level edge connectivity of consecutive predicted intersections rather than DriveSkel-LM's interval-level value; metric definitions in Section 4.2.

Metric	$\mathrm{CPT_F} + \mathrm{SFT_F}$	DriveSkel-LM
RC ↑	99.3%†	89.6%
BEA ↑	7.6%	95.2%
RLR ≈ 1	2.926	1.206

3.2 Road-Name-Aware Skeletonization

Generating a full intersection sequence mirrors TransitLM's complete-sequence decoding (Guo et al., 2026), where the prediction specifies a route without graph search. For driving, however, the reliability problems of such long sequences on dense, highly branched networks described in the introduction motivate a shorter representation that retains route-changing decisions while omitting unchanged-road intersections.

Table 1 quantifies this failure mode on optimal routing. Subscripts F and S denote the full-intersection and road-name-change skeleton representations used for continued pre-training (CPT) and fine-tuning (SFT); the $\mathrm{CPT_F} + \mathrm{SFT_F}$ column decodes every intersection, whereas DriveSkel-LM ($\mathrm{CPT_S} + \mathrm{SFT_S}$) is our full method (Section 4). Decoding every intersection keeps high pair-level edge connectivity but attaches both endpoints in only 7.6% of cases—reaching the origin region in 93.2% yet the destination in only 7.8%. The two Route Connectivity values use different definitions and are *not* directly comparable: the full-sequence model's higher value does not indicate a better route, since a route is usable only when also correctly attached at both ends. This is consistent with error accumulation over long autoregressive sequences, though we do not yet isolate it from other failure modes such as output-length truncation, repeated-intersection loops, or underused stopping conditions. Such attachment is far too low to yield a usable route, so we adopt the skeleton representation, which keeps Both-End Attachment above 95% and reduces Route Length Ratio from 2.926 to about 1.21 while preserving the decisions needed to reach the destination.

For a route $\mathbf{r}$, let $n_j = n((v_{j-1}, v_j))$ denote the road name of its jth edge. We define the retained index set as

$$\mathcal{B}(\mathbf{r}) = \{0, N\} \cup \{j \in \{1, \dots, N-1\} \mid n_j \neq n_{j+1}\}. \tag{1}$$

Let $0 = b_0 < b_1 < \cdots < b_M = N$ enumerate $\mathcal{B}(\mathbf{r})$. The skeleton and its aligned road-name sequence are

$$\mathbf{s} = (v_{b_0}, \dots, v_{b_M}), \qquad a_i = n_{b_{i-1}+1}, \quad i = 1, \dots, M. \tag{2}$$

Thus, a_i specifies the road traversed from s_{i-1} to s_i, while intermediate intersections on that road are omitted. This construction produces the supervision target used for skeleton generation.

For a uniquely specified segment, the boundary intersections and aligned road name determine it without graph search, so DriveSkel-LM recovers the route by deterministically expanding each such segment; segments not uniquely resolved by the static road information are classified by local recoverability and, if needed, filled by A^* (Section 3.4) rather than by re-planning the whole route.

As motivated in the introduction, pairing globally unique intersection tokens with road names is a middle ground between coarse road names and a per-segment vocabulary: it retains intersection-level route decisions while keeping the generated sequence compact.

3.3 Token-Adapted, Task-Conditioned Skeleton Generation

We build DriveSkel-LM on Qwen3-4B (Yang et al., 2025) and extend its tokenizer with 49,082 globally unique intersection tokens, letting the model generate route decisions as discrete network entities rather than coordinates or free-form descriptions. We then continue pre-training on anonymized navigation-route records converted to the skeleton representation, together with intersection knowledge and POI descriptions (Gururangan et al., 2020), grounding intersection tokens in road connectivity, route-level patterns, and POI-derived location semantics. The same representation is used for task-specific fine-tuning.

For each task, we initialize from the same pre-trained model and fine-tune a separate model conditioned on the request x, which autoregressively generates the compact representation $y = (\mathbf{s}, \mathbf{a})$ defined in Section 3.1. The query and constraints in x set the planning objective while the output representation stays shared across tasks. Given the tokenized output $y = (y_1, \ldots, y_T)$, we optimize the standard causal language-model objective

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{t=1}^{T} \log p_\theta(y_t \mid x, y_{<t}). \tag{3}$$

Here θ denotes the trainable model parameters, T is the output length, and p_θ is the conditional token distribution.

Training corpus. The CPT corpus comprises three parts: navigation-route records converted to the skeleton representation, POI descriptions, and intersection-connectivity descriptions, which ground intersection tokens in route-level patterns, location semantics, and road connectivity. The intersection and POI descriptions are cross-linked: each intersection record lists its nearby POIs, and each POI record lists its nearby intersections. The SFT data cover three tasks—optimal routing, road-name-constrained routing, and en-route search—each formatted as a system prompt, a user prompt, and a JSON label. Appendix A gives representative examples of both.

3.4 Deterministic Route Recovery and Local Fallback

Given a predicted skeleton $(\mathbf{s}, \mathbf{a})$, DriveSkel-LM recovers the route one segment at a time (Figure 2). It first checks whether each triplet (s_{i-1}, a_i, s_i) uniquely specifies a directed route segment in the road graph; if so, deterministic lookup restores the intersections omitted by skeletonization without graph search. The recovered segments are concatenated in skeleton order to form the complete route.

A predicted skeleton segment may not be uniquely resolved by the road graph. Let $\mathcal{U} \subseteq \{1, \ldots, M\}$ denote the indices i for which the triplet (s_{i-1}, a_i, s_i) uniquely identifies a directed route segment in $\mathcal{G}$. For every remaining segment, we consider only directed paths that contain its predicted road name. Let $\pi : u \rightsquigarrow v$ denote a directed path from u to v in $\mathcal{G}$, and let e denote an edge on π. We define the road-name-conditioned minimum number of distinct road names as

$$\kappa_a(u, v) = \min_{\substack{\pi: u \rightsquigarrow v \\ a \in \{n(e) \mid e \in \pi\}}} \left| \{n(e) \mid e \in \pi\} \right|. \tag{4}$$

If no directed path from u to v contains a, we set $\kappa_a(u, v) = \infty$. Let $\mathcal{J} = \{1, \ldots, M\} \setminus \mathcal{U}$ be the unresolved segment indices. We partition them as

$$\begin{aligned} \mathcal{P} &= \{i \in \mathcal{J} \mid \kappa_{a_i}(s_{i-1}, s_i) \leq 2\}, \\ \mathcal{F} &= \{i \in \mathcal{J} \mid \kappa_{a_i}(s_{i-1}, s_i) > 2\}. \end{aligned} \tag{5}$$

Together, $\mathcal{U}$, $\mathcal{P}$, and $\mathcal{F}$ form a partition of $\{1, \ldots, M\}$. Here $\mathcal{U}$ is defined by uniqueness rather than by a road-name count, and $\kappa_{a_i}(s_{i-1}, s_i) \leq 2$ is only a necessary condition for $\mathcal{P}$, so the mapping to the three recovery cases is typical rather than exact. The $\mathcal{P}$ case (a *local gap*, or break point) rests on an empirical regularity rather than a guarantee: it holds when each named road induces a simple chain and any two named roads meet at most once with directed reachability, and occasional counterexamples exist (a predicted road may be followed by either of two distinct second roads that share the same two names), in which case $\mathcal{A}^*$ returns one feasible connection instead of certifying uniqueness. The recovered segment $\hat{\mathbf{r}}_i$ is

$$\hat{\mathbf{r}}_i = \begin{cases} \mathcal{R}(s_{i-1}, a_i, s_i), & i \in \mathcal{U}, \\ \mathcal{A}^*(s_{i-1}, a_i, s_i), & i \in \mathcal{P}, \\ \perp, & i \in \mathcal{F}, \end{cases} \tag{6}$$

where $\mathcal{R}$ denotes deterministic segment expansion, $\perp$ denotes failed recovery, and $\mathcal{A}^*$ applies A^* to a local-gap interval under an explicit road-name budget: it searches only directed paths from s_{i-1} to s_i that contain a_i and use at most two distinct road names, and returns a minimum-cost one, so the connection satisfies the same $\kappa \leq 2$ condition that admitted the segment to $\mathcal{P}$ rather than drifting onto a shorter multi-name path. The search is confined to the two predicted skeleton intersections,

with all other decisions retained, so graph search is reserved for local-gap intervals rather than replanning the whole route. More fundamentally, when a segment is not uniquely specified—e.g., a fork inside a single named road makes two distinct paths share the triplet (s_{i-1}, a_i, s_i), since neither fork node is a road-name change—recovery returns one feasible path satisfying the constraint rather than certifying it reproduces the reference route; this is one reason Route IoU stays moderate.

The local fallback is deliberately not extended to every unresolved prediction: a segment in $\mathcal{F}$ returns $\perp$ and the route is marked disconnected. This boundary distinguishes an omitted but locally recoverable transition from one exceeding the scope of local recovery, and prevents the fallback from masking failures in skeleton generation. Throughout, *local* refers to locality in the number of distinct road names (at most two), not to spatial distance or to the number of omitted decisions or intersections.

Intuitively, the three cases track how many distinct road names lie between two consecutive skeleton intersections: one name connects them uniquely ($\mathcal{U}$), two form a local gap that is usually uniquely recoverable ($\mathcal{P}$), and three or more form a nonlocal gap that generally is not ($\mathcal{F}$).

End-to-end inference. At test time a request $x = (q, o, d, c)$ is rendered into the task-specific prompt (Appendix A.2), and the model autoregressively emits the JSON object holding the aligned road names $\mathbf{a}$ and skeleton $\mathbf{s}$, which we parse into $(\mathbf{s}, \mathbf{a})$ and recover segment by segment as above. Because only endpoints and road-name-change intersections are emitted, this sequence is far shorter than a full-intersection decoding of the same route—the proximate source of the attachment and length gains in Section 4.

4 Experiments

4.1 Experimental Setup

Data and temporal split. All experiments use one Beijing static road-graph snapshot for training, recovery, and evaluation. The continued pre-training corpus has 3,191,100 examples from a single daily partition: 2,433,217 de-identified navigation-route records, 245,410 intersection-connectivity descriptions, and 512,473 high-popularity POI descriptions. Per task, we sample 10,000 training and 1,000 test examples from temporally ordered, non-overlapping daily partitions after that partition.

Implementation and evaluation protocol. DriveSkel-LM uses Qwen3-4B with the tokenizer extended by 49,082 globally unique intersection tokens. Training is two-stage (Figure 2): continued pre-training on the corpus above, then supervised fine-tuning on each task's 10,000 examples in the same skeleton representation; unless stated otherwise we fine-tune one model per task from the same checkpoint. At inference the skeleton is expanded against the static road graph by the recovery procedure of Section 3.4, invoking A^* only for local-gap intervals. Every metric compares the recovered route with the realized label; all test partitions are strictly later than the training ones, so they do not overlap in time, though we do not de-duplicate identical routes across days, so this does not guarantee unseen-route generalization.

4.2 Tasks and Evaluation Metrics

Tasks. We evaluate three tasks sharing the same skeleton representation and recovery procedure. Optimal routing generates a route between the requested origin and destination without extra constraints; its label is the route the user actually executed, so "optimal" denotes this realized reference, not a provable optimum. Road-name-constrained routing takes or avoids a user-named road; en-route search requires passing a requested POI or one of the requested category.

Evaluation metrics. We compare the generated skeleton and the filled route (recovered via Section 3.4) against the complete label route. All tasks share five metrics; constrained tasks add a constraint-satisfaction measure. Route Length Ratio and Route IoU average only over examples whose recovery succeeds on every segment (no $\perp$), whereas Route Connectivity, Both-End Attachment, and Skeleton Recovery cover the full test set; the quality metrics should thus be read with Route Connectivity.

Table 2: DriveSkel-LM on the three tasks—Opt (optimal routing), Road (road-name-constrained routing), POI (en-route search)—at the full CPT corpus with single-task fine-tuning. Constraint-satisfaction columns: RNC = road-name consistency and Comp = route compliance (Road); POI-N = exact-POI-name and POI-C = POI-category consistency (POI); Opt imposes no constraint (–). Other abbreviations as in Section 4.2; rates in %.

Task	BEA↑	RC↑	RLR≈ 1	IoU↑	SR↑	LG↓	RNC↑	Comp↑	POI-N↑	POI-C↑
Opt	95.2	89.6	1.206	17.4	24.2	17.8	–	–	–	–
Road	95.9	92.8	1.191	23.5	31.7	26.5	98.2	65.1	–	–
POI	95.4	91.3	1.184	28.3	36.9	25.2	–	–	21.3	96.1

Route Connectivity (RC). For two adjacent skeleton intersections, if driving from the earlier to the later along the switched road name requires three or more distinct road names (or no directed path retains that name), the interval is a *nonlocal gap* (disconnection jump). With $\mathcal{F}^{(k)}$, $\mathcal{P}^{(k)}$ the nonlocal- and local-gap interval sets of Section 3.4 and M_k the number of skeleton intervals of example k, the pair-level Nonlocal-Gap Rate (NLG) is

$$G_{\text{nonlocal}} = \frac{\sum_k |\mathcal{F}^{(k)}|}{\sum_k M_k}, \tag{7}$$

and **Route Connectivity** is $1 - G_{\text{nonlocal}}$, the fraction of intervals that stay within local recovery and reconnect continuously. We also report the Local-Gap Rate (LG) $G_{\text{local}} = \sum_k |\mathcal{P}^{(k)}| / \sum_k M_k$, the fraction not uniquely resolved but locally recoverable through at most two road names.

Both-End Attachment (BEA). The fraction of examples whose requested origin and destination are each within 500 m of the first and last generated-skeleton intersection. This proximity measure does not certify legal vehicle access; we report the two ends separately only when analyzing endpoint failures.

Route Length Ratio (RLR). Letting $\widetilde{\mathbf{r}}^{(k)}$ be the filled route and $\mathbf{r}^{(k)}$ the complete label route, we report

$$R_L = \frac{\sum_k L(\widetilde{\mathbf{r}}^{(k)})}{\sum_k L(\mathbf{r}^{(k)})}, \tag{8}$$

where $L(\cdot)$ is route length; R_L is a single ratio of summed filled to summed label length (not a mean of per-example ratios), and a value near 1 means the routes neither detour nor collapse relative to the labels.

Route IoU (IoU). Per example, the shared mileage of the filled and label routes over their union, averaged over examples; it measures geometric overlap with the reference.

Skeleton Recovery (SR). The fraction of the label skeleton (its road-name-change intersections) covered by the generated skeleton. This node-level recall checks neither intersection order nor extra generated intersections, so it should not be read alone as decision accuracy.

Task-specific constraint satisfaction. For road-name-constrained routing we report the *road-name consistency rate* (RNC; fraction whose generated road names match the requested road) and, among those, the *route compliance rate* (Comp; fraction whose recovered route is connected, attached at both ends, and matches the request direction—traversing the named road when requested, avoiding it when excluded), a conditional rate over road-name-matched examples. For en-route search we report the POI-name (POI-N), POI-coordinate, and POI-category (POI-C) consistency rates and the mean distance from the responded POI to the skeleton.

4.3 Main Results

As motivated in Table 1, the compact skeleton is essential for endpoint attachment; Table 2 reports DriveSkel-LM at the full CPT corpus with single-task fine-tuning per task.

In Table 2 each model is fine-tuned and evaluated on a single task (CPT is skeleton S at the full corpus). Across all three, DriveSkel-LM yields well attached, largely connected routes: BEA 95.2–95.9%, RC 89.6–92.8%, and RLR 1.184–1.206, i.e. filled routes run about 18–21% longer than the labels. Two patterns recur. First, the constrained tasks beat optimal routing on geometry-sensitive

Table 3: General-purpose LLMs on the relaxed road-name-summary task (only the ordered road-name summary; no intersection tokens or route recovery). Set = road-name-set validity; Seq = road-name-sequence validity (both for all three tasks); Constr = road-constraint satisfaction; POI/Categ = exact- and category-POI hit rates. All in %; higher is better.

	Opt		Road			POI			
Model	Set	Seq	Set	Seq	Constr	Set	Seq	POI	Categ
Doubao-Seed-2.0-Pro	17.56	10.53	21.95	15.48	91.75	12.15	8.51	0.00	95.88
Qwen3.8-Max	20.31	11.96	22.19	15.11	94.74	13.98	9.61	0.00	93.75
Gemini-3.1-Pro	25.35	16.75	32.29	22.24	92.78	22.35	15.58	7.22	95.88
DeepSeek-V4-Pro	14.30	9.00	17.96	12.50	90.32	10.50	7.06	3.09	82.47
Claude-Opus-4.8	13.47	8.34	18.04	12.17	92.78	7.97	5.46	0.00	94.85
GPT-5.5	25.37	17.01	31.78	23.91	91.75	21.56	14.79	0.00	93.81

Table 4: CPT representation-type ablation on optimal routing (F: full-intersection sequence; S: road-name-change skeleton); both rows use skeleton SFT.

CPT	BEA↑	RC↑	RLR≈ 1	IoU↑	SR↑	LG↓	NLG↓
F	95.3	88.1	1.217	19.3	26.4	16.0	11.9
S	95.2	89.6	1.206	17.4	24.2	17.8	10.4

metrics (RC 92.8%/91.3% vs. 89.6%; IoU 23.5%/28.3% vs. 17.4%; SR 31.7%/36.9% vs. 24.2%), plausibly because an explicit road-name or POI-category constraint narrows candidates and anchors generation—echoing the CPT-volume ablation, where the road-name task degrades least. Second, RLR stays in a narrow band, so the three tasks incur similar extra mileage; separating recovery from prediction error needs an oracle-skeleton comparison, left for future work. Route overlap remains moderate, and constraints are partly satisfied (route compliance 65.1%; same-category POI 96.1%).

4.4 Comparison with General-Purpose LLMs

We evaluate six general-purpose LLMs on a strictly simpler variant of the three tasks (Table 3): instead of generating intersection tokens and recovering a full route, each model only outputs the ordered road-name summary, which largely reduces to recognizing names already in the query. Even so, none approaches a valid route: the best road-name-set validity is 25.37% (GPT-5.5) on optimal routing and 32.29% (Gemini-3.1-Pro) on road-name-constrained routing, and the best ordered sequence only 17.01% and 23.91%. On en-route search the exact POI is essentially never hit (0–7.22%); the apparently high same-category rate (82–96%) mostly reflects echoing the queried category, not grounding a specific POI.

These figures are not a like-for-like ranking: the LLMs are scored on whole-sequence road-name validity, DriveSkel-LM on interval-level connectivity and endpoint proximity, so the percentages measure different events and support only the narrower claim that general-purpose LLMs without domain adaptation struggle to produce a valid, ordered road-name sequence. On its own metrics DriveSkel-LM far exceeds all six models (Tables 2–6); a fair, same-protocol comparison is left for future work.

4.5 Ablation Study

We isolate three training-data factors: whether the CPT representation matches the generation target, how much CPT grounds the intersection vocabulary, and whether SFT is per-task or shared. Abbreviations follow Section 4.2; rates are in %.

CPT data type. Both share the skeleton SFT target and differ only in the CPT representation (Table 4), isolating a *distribution-matching* effect: pre-training on the same sparse, decision-point representation SFT expects grounds intersection tokens in the "road-name-change" semantics the generator must emit, lowering the Nonlocal-Gap Rate (11.9%→10.4%) and raising Route Connectivity to 89.6%. Full-intersection CPT teaches dense transitions that do not transfer to the compact target; its slight Route IoU edge (19.3% vs. 17.4%) reflects a recoverability–overlap trade-off, not a memorization effect we isolate.

Table 5: CPT data-volume ablation: full (100%) vs. reduced (40%) CPT corpus, with matched single-task SFT and evaluation per task. RNC/Comp = road-name consistency and route compliance (Road); POI-N/POI-C = exact-POI-name and POI-category consistency (POI); – = not applicable.

Amt	Task	BEA↑	RC↑	RLR≈ 1	IoU↑	SR↑	LG↓	NLG↓	RNC↑	Comp↑	POI-N↑	POI-C↑
100%	Opt	95.2	89.6	1.206	17.4	24.2	17.8	10.4	–	–	–	–
100%	Road	95.9	92.8	1.191	23.5	31.7	26.5	7.2	98.2	65.1	–	–
100%	POI	95.4	91.3	1.184	28.3	36.9	25.2	8.7	–	–	21.3	96.1
40%	Opt	95.2	86.7	1.254	15.5	22.0	5.6	13.3	–	–	–	–
40%	Road	94.8	89.4	1.242	23.8	31.9	4.4	10.6	98.3	65.3	–	–
40%	POI	94.9	86.4	1.203	25.7	34.1	5.7	13.6	–	–	22.7	96.7

Table 6: SFT task-composition ablation at the full (100%) CPT corpus: single-task models (one per task) vs. one multi-task model (*All*) trained on all three and evaluated on each. RNC/Comp = road-name consistency and route compliance (Road); POI-N/POI-C = exact-POI-name and POI-category consistency (POI); – = not applicable.

SFT	Eval	BEA↑	RC↑	RLR≈ 1	IoU↑	SR↑	LG↓	NLG↓	RNC↑	Comp↑	POI-N↑	POI-C↑
Single	Opt	95.2	89.6	1.206	17.4	24.2	17.8	10.4	–	–	–	–
Single	Road	95.9	92.8	1.191	23.5	31.7	26.5	7.2	98.2	65.1	–	–
Single	POI	95.4	91.3	1.184	28.3	36.9	25.2	8.7	–	–	21.3	96.1
All	Opt	98.3	93.4	1.156	22.8	31.0	17.5	6.6	–	–	–	–
All	Road	98.6	93.6	1.174	26.5	35.6	9.1	6.4	89.7	64.2	–	–
All	POI	98.7	91.4	1.156	28.8	38.3	8.6	8.6	–	–	11.7	94.5

CPT data volume. Reducing the CPT corpus to 40% (Table 5) changes gap *severity* rather than uniformly lowering quality: since the roughly 49k intersection tokens are sparsely observed, 40% CPT under-trains many tokens and their adjacency, collapsing the Local-Gap Rate (optimal routing 17.8%→5.6%) while the nonlocal-gap share grows (10.4%→13.3%), driving Route Connectivity down (89.6%→86.7%; en-route search 91.3%→86.4%). Because the two models yield different skeletons, these shifts are not the same gaps migrating between categories. Two aspects stay robust: Both-End Attachment holds near 95% (endpoints are cued by prompt coordinates, not pre-training coverage), and the road-name task degrades least (route compliance 65.1% vs. 65.3%), plausibly because its explicit constraint anchors the sequence.

SFT task composition. Single- and multi-task (*All*) models are both trained at the full CPT corpus (Table 6). Since all three tasks share one core skill—emitting a correct road-name-change skeleton over globally unique intersection tokens—multi-task SFT supplies roughly three times more skeleton evidence. It lifts RC, BEA, RLR, and SR on every task (optimal routing: RC 89.6%→93.4%, BEA 95.2%→98.3%), at a cost confined to strict per-task precision (road-name consistency 98.2%→89.7%, exact-POI 21.3%→11.7%).

A qualitative case study of two representative generated routes is provided in Appendix B.

5 Conclusion

We introduced DriveSkel-LM, a road-name-aware framework for driving route planning with language models. Rather than decoding every intersection, it generates a compact skeleton of globally unique intersections and aligned road names; uniquely specified segments are reconstructed without graph search, with local A^* reserved for recoverable gaps. On temporally separated Beijing data, this substantially improves endpoint attachment and route-length quality over full-intersection generation, and one shared representation supports unconstrained routing, road-name constraints, and en-route search. Route IoU and Skeleton Recovery remain moderate, and connectivity and attachment are measured at the interval and endpoint level rather than as a whole-route joint success rate: routes are well attached and largely connected but do not yet reproduce the reference geometry closely. These results support the potential of LLM-based driving route planning; a whole-route success metric, tighter geometric fidelity, broader coverage, dynamic conditions, and stronger constraint satisfaction are left for future work.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting and language revision. The authors take responsibility for the technical content, experimental design, reported results, and final manuscript.

ETHICS STATEMENT

This study uses de-identified historical navigation-route records, static road connectivity, and POI descriptions. We report only aggregate experimental results and do not analyze or seek to infer individual mobility behavior.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce our experiments. The paper specifies the backbone language model, the expanded intersection-token vocabulary, the training protocol (continued pre-training followed by task-specific supervised fine-tuning), the corpus composition, and the temporally separated per-task train/test splits. It formalizes the road-name-aware skeletonization and the deterministic route-recovery procedure, and gives a formal definition of every evaluation metric. The appendix further provides the exact CPT and SFT data formats—system prompt, user prompt, and JSON label—with representative examples for each task, together with qualitative case-study outputs. The navigation-route, road-connectivity, and POI data are de-identified real-world records that cannot be released owing to privacy and proprietary constraints; the complete data formats, examples, and training protocol described above are provided to enable replication on comparable datasets.

REFERENCES

Hannah Bast, Daniel Delling, Andrew V Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. Route planning in transportation networks. In *Algorithm Engineering: Selected Results and Surveys*, pp. 19–80. Springer, 2016.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2605–2611, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

Julian Dibbelt, Ben Strasser, and Dorothea Wagner. Customizable contraction hierarchies. In *International Symposium on Experimental Algorithms*, pp. 271–282. Springer, 2014.

Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pp. 287–290. 2022.

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.

Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *International workshop on experimental and efficient algorithms*, pp. 319–333. Springer, 2008.

Robert Geisberger, Peter Sanders, Dominik Schultes, and Christian Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, 2012.

Andrew V Goldberg and Chris Harrelson. Computing the shortest path: a^* search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 156–165, 2005.

Andrew V Goldberg, Haim Kaplan, and Renato F Werneck. Reach for a^*: Efficient point-to-point shortest path algorithms. In *Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 129–143. SIAM, 2006.

Hanyu Guo, Chao Chen, Longfei Xu, Chengzhang Wang, Kaikui Liu, and Xiangxiang Chu. Towards full candidate interaction: A comprehensive comparison network for better route recommendation. *arXiv preprint arXiv:2508.08745*, 2025.

Hanyu Guo, Jiedong Yang, Chao Chen, Longfei Xu, Kaikui Liu, and Xiangxiang Chu. Transitlm: A large-scale dataset and benchmark for map-free transit route generation. *arXiv preprint arXiv:2605.22355*, 2026.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Chonghua Han, Yuan Yuan, Jingtao Ding, Jie Feng, Fanjin Meng, and Yong Li. MoveGPT: Scaling mobility foundation models with spatially-aware mixture of experts. *arXiv preprint arXiv:2505.18670*, 2025.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 831–840, 2014.

Qingyue Long, Yuan Yuan, and Yong Li. A universal model for human mobility prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 894–905, 2025.

Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pp. 352–361, 2009.

Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pp. 336–343, 2009.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Peter Sanders and Dominik Schultes. Engineering highway hierarchies. In *European Symposium on Algorithms*, pp. 804–816. Springer, 2006.

Jie-Jing Shao, Bo-Wen Zhang, Xiao-Wen Yang, Baizhi Chen, Siyu Han, Jinghao Pang, Wen-Da Wei, Guohao Cai, Zhenhua Dong, Lan-Zhe Guo, et al. ChinaTravel: An open-ended travel planning benchmark with compositional constraint validation for language agents. In *International Conference on Learning Representations*, volume 2026, pp. 96008–96060, 2026.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Jingyuan Wang, Yujing Lin, and Yudong Li. GTG: Generalizable trajectory generation model for urban mobility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 834–842, 2025a.

Xiangheng Wang, Ziquan Fang, Chenglong Huang, Danlei Hu, Lu Chen, and Yunjun Gao. GTR: A general, multi-view, and dynamic framework for trajectory representation learning. In *Forty-second International Conference on Machine Learning*, 2025b.

Zheng Wang, Kun Fu, and Jieping Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 858–866, 2018b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-Drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pp. 99–108, 2010.

Jie Zhao, Chao Chen, Yuanshao Zhu, Mingyu Deng, and Yuxuan Liang. UniTR: A unified framework for joint representation learning of trajectories and road networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13348–13356, 2025.

Haicang Zhou, Weiming Huang, Yile Chen, Tiantian He, Gao Cong, and Yew-Soon Ong. Road network representation learning with the third law of geography. In *Advances in Neural Information Processing Systems*, volume 37, pp. 11789–11813, 2024.

Silin Zhou, Shuo Shang, Lisi Chen, Peng Han, and Christian S Jensen. Grid and road expressions are complementary for trajectory representation learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2135–2146, 2025.

A TRAINING DATA EXAMPLES

This appendix gives representative examples of the continued-pre-training (CPT) and supervised fine-tuning (SFT) data. For readability, the original Chinese records are translated into English; intersection IDs (e.g., `drive-XXXXXXXX`), coordinates, and JSON field names are kept unchanged. The CPT records cover three semantic types—intersection connectivity, POI descriptions, and navigation routes, the last shown in both skeleton and full-intersection form—and the SFT data cover three tasks, each shown with its system prompt, user prompt, and label.

A.1 CONTINUED-PRE-TRAINING DATA

Intersection knowledge.

drive-00040595: intersection of Deze West Street, West Ring Middle Road, and West Ring Middle Road Side Road; no traffic light; city: Beijing; coordinates: 116.508028,39.777056. Nearby POIs: Beijing Vocational University of Science and Technology - Express Service Center; YTO Express Oriental Finance Bookkeeping and Business Registration; Beijing Vocational University of Science and Technology, School of Basic Studies; Ruyi Eyewear Store; Jushan Xiaolai Car Beauty Store; Fengchao Express Locker; Yizhuang Mige Laboratory; Meituan Food Delivery; Chengqi Convenience Store; Beijing Vocational University of Science and Technology, School of Integrated Circuits; Hanhuang Laundry and Care (Yizhuang Branch); Beijing Vocational University of Science and Technology Gymnasium; Beijing Vocational University of Science and Technology Student Canteen No. 2; Beijing Vocational University of Science and Technology, School of Economics and Management; SpecialU Art Center; Beijing Electronics; People's Rule-of-Law Online Public-Opinion Research Center; Yiliu Art Framing Studio; Mama Station; Youhui Self-Service Billiards and Board Games; Beijing Vocational University of Science and Technology, School of Art and Design; Kunlei Guitar and Eloquence Art Center; NAURA Express Locker; Beijing Vocational University of Science and Technology, School of Telecommunication Engineering, Teaching Building No. 2; Jubaolong Fresh Supermarket; Cainiao Station; Aitian Cut Fruit (Yizhuang Branch); Zhongjin Yunwang Canteen; Fengchao Locker; Beijing Hongxin Decoration; Beijing Vocational University of Science and Technology, School of Art, Training Building; Aochi Sports Center; Benese Skincare & Styling (Haojing International Building 3 Branch); Beijing Vocational University of Science and Technology, School of Mechanical and Electrical Engineering; Haojing Xingsheng Convenience Supermarket; Huayou Shaobo Table Tennis Club; Haoxin Technology Computer Assembly, Repair, and Recycling Service Center; Staple Food Kitchen; Yizhuang Town Legal Service Center; Beijing Vocational University of Science and Technology Teaching Building; Youju Youzhu (Haojing International Branch); Jinlinbao Express; Beijing Vocational College of Electronic Science and Technology, Building F Food-Delivery Locker; Hefei Banyitong Beijing Marketing Service Center; Beijing Vocational University of Science and Technology, School of Bioengineering; Tongqu Sand Play Area; Lianhuasheng Elegant Recreation Club; China Sports Lottery (Taiheyuan Wumart Supermarket Branch); BOE Core Competence Building; NAURA; Beijing Vocational University of Science and Technology Administration Building; 21st Century Real Estate (Haojing International Branch); Shuma Zhuangyuan Basketball Hall; Graphic and Text Quick-Printing Advertising; Beijing Vocational College of Electronic Science and Technology Express Locker; Xiaoguan Tea Teahouse (Yizhuang AVIC Plaza Branch); Public Rental Housing Leasing Service Center; Yinanxin; Lin Xiansen ID Photo (Beijing Vocational University of Science and Technology Student Service Center Branch); Fengchao Smart Locker (Foxconn Phase III South Gate, Locker No. 1); Public Toilet; Luquan Cemetery; China Household Electric Appliance Research Institute (Boxing 8th Road); Xiaopang Master Studio; Beijing Vocational University of Science and Technology Express Station; CH V-KOOL Invisible Car Cover, Heat-Insulation Film, and Color-Change Film; China Rolls-Royce Club Rolls Reception Hall; Hongxing Erguotou Tobacco, Liquor, and Tea Supermarket; Pinduoduo Station; Fanhua Nail and Eyelash; Lanxu Dance Space; Yizhuang Electric Power Research Institute Swimming Pool; Mingxing Table Tennis Club; Aichezhiwu Automobile Service Co., Ltd.; 7FRESH Supermarket (Beijing Zhongdian Jinyang); HH Rhythmic Gymnastics Club; Zhihuixing Rental; Qiwu Shiguang Art Center; Fengchao Express Locker; 35DayGYM Comprehensive Training Hall; Beijing Vocational University of Science and Technology Bathhouse (Boxing 8th Road); Huamu China Lanzhou Hand-Pulled Noodles; SMIC; Fengchao Smart Express Locker; FreshCube Smart Fresh-Drink Robot (Integrated-Circuit Industry-Education Integration Base Station); Tengyue Sports

Badminton; Yitongchang; Cainiao Station (Haojing International Branch); Yezi Makeup Room; Meile Cute Pet Store (Yizhuang Branch); Fengchao Express Locker (BOE Core Competence Building Branch); Maoyun Qingshang Music Education (Yizhuang Campus); Fengchao Express Locker (AVIC International Plaza Branch); Beijing Vocational University of Science and Technology Building G; Shijueya Styling; Beijing Vocational University of Science and Technology; Shuma Zhuangyuan Restaurant; Beijing Wentao Kung Fu Studio; Yujie Badminton; Beijing Vocational University of Science and Technology - Fitness Center; Active Health Research Institute; Aoyu Sports Space (Shuma Zhuangyuan); Fengchao Express Locker; Beijing Vocational University of Science and Technology, School of Art and Design, Lecture Hall; Yunda Express (Haojing International Building 2 Branch); Emergency Shelter; Foxconn Phase III Recruitment Center; Yarui Qianyi Beauty and Wellness (Haojing International Branch); 7FRESH Warehouse Delivery Station; Tianlei Combat Sports; Ningxia Tan Sheep (Boxing 9th Road Branch). Road network between adjacent intersections: between this intersection and drive-00040620 lies Deze West Street (urban secondary arterial, 91 m); between this intersection and drive-00040608 lies West Ring Middle Road (urban primary arterial, 79 m).

POI knowledge.

Bobo Fishing Park; category: fishing park; city: Beijing; coordinates: 116.923131,40.395824. Nearby roads: G45 Daguang Expressway; Right Levee Road; Jingshen Road; Shuangshiling Road; Left Levee Road; Stone Levee Road. Nearby intersections: drive-00010433; drive-00010432; drive-00010434.

Route description (skeleton only).

Route plan from my location to Zhongmensi Street. Planning city: Beijing. Start coordinate: 116.144707,39.903182; end coordinate: 116.089838,39.933785. Follow this route: depart from the start (coordinate 116.144707,39.903182), arrive at intersection drive-00017551 (coordinate 116.144032,39.903093, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 2960 m, arrive at intersection drive-00017409 (coordinate 116.124051,39.923424, has traffic light), turn onto Pushan Road (urban primary arterial) and travel 861 m, arrive at intersection drive-00017411 (coordinate 116.114243,39.921205, has traffic light), turn onto Xinqiao Street (urban primary arterial) and travel 927 m, arrive at intersection drive-00017252 (coordinate 116.108967,39.928425, has traffic light), turn onto Gaojiayuan Road (urban primary arterial) and travel 1301 m, arrive at intersection drive-00017122 (coordinate 116.097884,39.934226, has traffic light), turn onto South Dayu Road (urban primary arterial) and travel 471 m, arrive at intersection drive-00017150 (coordinate 116.092290,39.933905, has traffic light), turn onto South Zhongmensi Road (urban primary arterial) and travel 155 m, arrive at intersection drive-00017148 (coordinate 116.090743,39.933078, has traffic light), turn onto South Kuangjian Street (urban secondary arterial) and travel 106 m, arrive at intersection drive-00017160 (coordinate 116.089954,39.933838, has traffic light), and arrive at the destination Zhongmensi Street (coordinate 116.089838,39.933785). The current plan has a total time of 1087.0 s, a total distance of 6843.0 m, a total toll of 0.0 yuan, and 17.0 traffic lights.

Route description (full intersection sequence).

Route plan from my location to Zhongmensi Street. Planning city: Beijing. Start coordinate: 116.144707,39.903182; end coordinate: 116.089838,39.933785. Follow this route: depart from the start (coordinate 116.144707,39.903182), arrive at intersection drive-00017551 (coordinate 116.144032,39.903093, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 258.0 m, arrive at intersection drive-00017567 (coordinate 116.142849,39.905316, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 232.0 m, arrive at intersection drive-00017566 (coordinate 116.141682,39.907501, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 186.0 m, arrive at intersection drive-00017568 (coordinate 116.140680,39.909258, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 984.0 m, arrive at intersection drive-00017451 (coordinate 116.135088,39.917010, has traffic light), turn onto Yudai Street (urban secondary arterial) and travel 1300.0 m, arrive at intersection drive-00017409 (coordinate 116.124051,39.923424, has traffic light), turn onto Pushan Road (urban primary arterial) and travel 319.0 m, arrive at intersection drive-00017422 (coordinate 116.120498,39.922224, has traffic light), turn onto Pushan Road (urban pri-

mary arterial) and travel 503.0 m, arrive at intersection drive-00017443 (coordinate 116.114840,39.920926, no traffic light), turn onto Pushan Road (urban primary arterial) and travel 39.0 m, arrive at intersection drive-00017411 (coordinate 116.114243,39.921205, has traffic light), turn onto Xinqiao Street (urban primary arterial) and travel 927.0 m, arrive at intersection drive-00017252 (coordinate 116.108967,39.928425, has traffic light), turn onto Gaojiayuan Road (urban primary arterial) and travel 253.0 m, arrive at intersection drive-00017253 (coordinate 116.105878,39.928101, has traffic light), turn onto Gaojiayuan Road (urban primary arterial) and travel 644.0 m, arrive at intersection drive-00017153 (coordinate 116.099514,39.930697, has traffic light), turn onto Gaojiayuan Road (urban primary arterial) and travel 404.0 m, arrive at intersection drive-00017122 (coordinate 116.097884,39.934226, has traffic light), turn onto South Dayu Road (urban primary arterial) and travel 471.0 m, arrive at intersection drive-00017150 (coordinate 116.092290,39.933905, has traffic light), turn onto South Zhongmensi Road (urban primary arterial) and travel 155.0 m, arrive at intersection drive-00017148 (coordinate 116.090743,39.933078, has traffic light), turn onto South Kuangjian Street (urban secondary arterial) and travel 106.0 m, arrive at intersection drive-00017160 (coordinate 116.089954,39.933838, has traffic light), and arrive at the destination Zhongmensi Street (coordinate 116.089838,39.933785). The current plan has a total time of 1087.0 s, a total distance of 6843.0 m, a total toll of 0.0 yuan, and 17.0 traffic lights.

A.2 Supervised Fine-Tuning Data

Optimal routing. System prompt.

```
You are an expert in driving route planning, familiar with the driving
network.

[Task] Based on the user's start and end coordinates and nearby
intersection IDs, plan a comprehensively optimal driving route.

[Input]
- query: the user's natural-language request (the user wants a
  comprehensively optimal route, e.g., "Give me the best route.")
- start: start longitude/latitude coordinate, formatted as
  "longitude,latitude"
- start_intersection: intersection ID near the start, formatted as
  "drive-XXX"
- end: end longitude/latitude coordinate, formatted as "longitude,latitude"
- end_intersection: intersection ID near the end, formatted as "drive-XXX"
- city: the city, e.g., "Beijing", "Shanghai", "Shenzhen", "Chengdu"

[Output] JSON with the following fields:
1. endpoints_seq: the first and last intersection IDs of the route,
   e.g., ["drive-XXX", "drive-XXX"], in order from the start side to
   the end side
2. road_name_seq: a road-name description sequence one-to-one with
   skeleton_seq; the start is "Enter XX Road", intermediate entries
   are "From XX Road onto YY Road", and the end is "Along XX Road to
   the destination"
3. skeleton_seq: the sequence of skeleton intersection IDs; must
   include the first and last intersections, with intermediate
   intersections selected at road-name change points (kept when the
   road entering and the road leaving an intersection have different
   names)
4. eta: estimated travel time in seconds
5. length: total route length in meters
6. light: number of traffic lights passed
7. toll: toll amount in yuan
```

User prompt.

```
{"query": "Plan the best route for me.",
 "start": "116.363792,40.076926",
 "start_intersection": "drive-00015872",
 "end": "116.231034,40.220952",
 "end_intersection": "drive-00003238",
 "city": "Beijing"}
```


Label.


```
{
  "endpoints_seq": ["drive-00015872", "drive-00003238"],
  "road_name_seq": [
    "Enter Kexing Road",
    "From Kexing Road onto West Huangping Road",
    "From West Huangping Road onto Beiqing Road",
    "From Beiqing Road onto Beiqing Expressway",
    "From Beiqing Expressway onto Beiqing Expressway exit",
    "From Beiqing Expressway exit onto G7 Jingxin Expressway",
    "From G7 Jingxin Expressway onto Baiquanzhuang First Bridge",
    "From Baiquanzhuang First Bridge onto Shuinan Road",
    "From Shuinan Road onto Hongtai Road",
    "From Hongtai Road onto Xinshui Road",
    "From Xinshui Road onto Baibai Road",
    "From Baibai Road onto West South-Ring Road",
    "From West South-Ring Road onto West Ring Road",
    "From West Ring Road onto Zhengfu Street",
    "Along Zhengfu Street to the destination"
  ],
  "skeleton_seq": [
    "drive-00015872", "drive-00005126", "drive-00004948",
    "drive-00005144", "drive-00004747", "drive-00004743",
    "drive-00003565", "drive-00003511", "drive-00003509",
    "drive-00003211", "drive-00003214", "drive-00003225",
    "drive-00003243", "drive-00003233", "drive-00003238"
  ],
  "eta": 2434,
  "length": 30109,
  "light": 12,
  "toll": 5
}
```


Road-name-constrained routing. System prompt.

```
You are an expert in driving route planning, familiar with the driving
network.

[Task] Based on the user's road-avoidance request, start and end
coordinates, and nearby intersection IDs, plan a driving route that
satisfies the request.

[Input]
- query: the user's natural-language request, containing a specific
  road name and the intent to avoid it
- start: start longitude/latitude coordinate, formatted as
  "longitude,latitude"
- start_intersection: intersection ID near the start, formatted as
  "drive-XXX"
- end: end longitude/latitude coordinate, formatted as "longitude,latitude"
- end_intersection: intersection ID near the end, formatted as "drive-XXX"
```

```
- city: the city, e.g., "Beijing", "Shanghai", "Shenzhen", "Chengdu"
- road_name: the user's road request, a string, e.g., "want to take
  the 4th Ring Road", "do not want to take the 3rd Ring Road"

[Requirement] The planned route must completely avoid the
user-specified road.

[Output] JSON with the following fields:
1. road_name: the user's road request, a string, identical to the
   input, e.g., "want to take the 4th Ring Road"
2. endpoints_seq: the first and last intersection IDs of the route,
   e.g., ["drive-XXX", "drive-XXX"], in order from the start side to
   the end side
3. road_name_seq: a road-name description sequence one-to-one with
   skeleton_seq; the start is "Enter XX Road", intermediate entries
   are "From XX Road onto YY Road", and the end is "Along XX Road to
   the destination"
4. skeleton_seq: the sequence of skeleton intersection IDs; must
   include the first and last intersections, with intermediate
   intersections selected at road-name change points (kept when the
   road entering and the road leaving an intersection have different
   names)
```

User prompt.

```
{"query": "Plan a route that avoids Guang'anmen Inner Street.",
 "start": "116.363214,39.886570",
 "start_intersection": "drive-00031466",
 "end": "116.300323,39.861759",
 "end_intersection": "drive-00031623",
 "city": "Beijing",
 "road_name": "Do not take Guang'anmen Inner Street"}
```

Label.

```
{
  "road_name": "Do not take Guang'anmen Inner Street",
  "endpoints_seq": ["drive-00031466", "drive-00031623"],
  "road_name_seq": [
    "Enter Niujie 4th Alley",
    "From Niujie onto You'anmen Inner Street",
    "From You'anmen Inner Street onto You'anmen Outer Street",
    "From You'anmen Outer Street onto You'anmen Bridge",
    "From You'anmen Bridge onto South 2nd Ring Road",
    "From South 2nd Ring Road onto Caihuying Bridge",
    "From Caihuying Bridge onto Lize Road",
    "From Lize Road onto North Fengtai Road",
    "From North Fengtai Road onto North Fengtai Road exit",
    "From North Fengtai Road exit onto North Fengtai Road service road",
    "From North Fengtai Road service road onto Niwa Road",
    "Along Niwa Road to the destination"
  ],
  "skeleton_seq": [
    "drive-00031466", "drive-00031413", "drive-00031723",
    "drive-00031727", "drive-00031900", "drive-00031849",
    "drive-00031844", "drive-00031650", "drive-00031655",
    "drive-00031605", "drive-00031591", "drive-00031623"
  ]
}
```

En-route search. System prompt.

```
You are an expert in driving route planning, familiar with the driving
network.

[Task] Based on the user's start and end coordinates, nearby
intersection IDs, and an en-route request, select a place along the
route that matches the user's category requirement and plan a driving
route passing through it.

[Input]
- query: the user's natural-language request (the user only states the
  type of place to pass, not a specific name, e.g., "find a
  convenience store along the way." / "fill up gas along the way.")
- start: start longitude/latitude coordinate, formatted as
  "longitude,latitude"
- start_intersection: intersection ID near the start, formatted as
  "drive-XXX"
- end: end longitude/latitude coordinate, formatted as "longitude,latitude"
- end_intersection: intersection ID near the end, formatted as "drive-XXX"
- city: the city, e.g., "Beijing", "Shanghai", "Shenzhen", "Chengdu"

[Output] JSON with the following fields:
1. poi: the name of the selected en-route place (decided by you; must
   match the requested type and lie along the route)
2. endpoints_seq: the first and last intersection IDs of the route,
   e.g., ["drive-XXX", "drive-XXX"], in order from the start side to
   the end side
3. road_name_seq: a road-name description sequence one-to-one with
   skeleton_seq; the start is "Enter XX Road", intermediate entries
   are "From XX Road onto YY Road", and the end is "Along XX Road to
   the destination"
4. skeleton_seq: the sequence of skeleton intersection IDs; must
   include the first and last intersections, with intermediate
   intersections selected at road-name change points (kept when the
   road entering and the road leaving an intersection have different
   names)
```

User prompt.

```
{"query": "Find me a coffee shop along the way.",
 "start": "116.524970,39.904640",
 "start_intersection": "drive-00028356",
 "end": "116.442532,39.920778",
 "end_intersection": "drive-00027214",
 "city": "Beijing"}
```

Label.

```
{
  "poi": "NNCAFE Nuonuo Coffee (Chaoyang Jianguomen Branch)",
  "endpoints_seq": ["drive-00028356", "drive-00027214"],
  "road_name_seq": [
    "Enter South Huihe Street",
    "From North Huihe Street onto Tonghui Riverside Ramp Bridge",
    "From Tonghui Riverside Ramp Bridge onto Jianguo Road",
    "From Jianguo Road onto Jingtong Expressway entrance",
    "From Jingtong Expressway entrance onto Jingtong Expressway",
    "From Jingtong Expressway onto Jingtong Expressway exit",
    "From Jingtong Expressway exit onto Jianguo Road",
```

```
    "From Jianguo Road onto Jianguomen Outer Street",
    "From Jianguomen Outer Street onto Jianguomen Outer Street service road",
    "From Jianguomen Outer Street service road onto Dongdaqiao Road",
    "From Dongdaqiao Road onto North Ritan Road",
    "From North Ritan Road onto West Fangcaodi Street",
    "From West Fangcaodi Street onto Nanyingfang Hutong",
    "Along Nanyingfang Hutong to the destination"
  ],
  "skeleton_seq": [
    "drive-00028356", "drive-00028304", "drive-00028299",
    "drive-00028327", "drive-00028287", "drive-00028190",
    "drive-00028064", "drive-00028070", "drive-00028058",
    "drive-00027432", "drive-00027222", "drive-00027251",
    "drive-00027249", "drive-00027214"
  ]
}
```


B CASE STUDY

Figure 3 shows two representative routes generated by DriveSkel-LM. In each case a compact road-name-aware skeleton is recovered into a complete route by deterministic expansion, with local A^* reserved for recoverable gaps. Both routes are reasonable—endpoints correctly attached, paths connected and following plausible roads, and lengths close to the references.

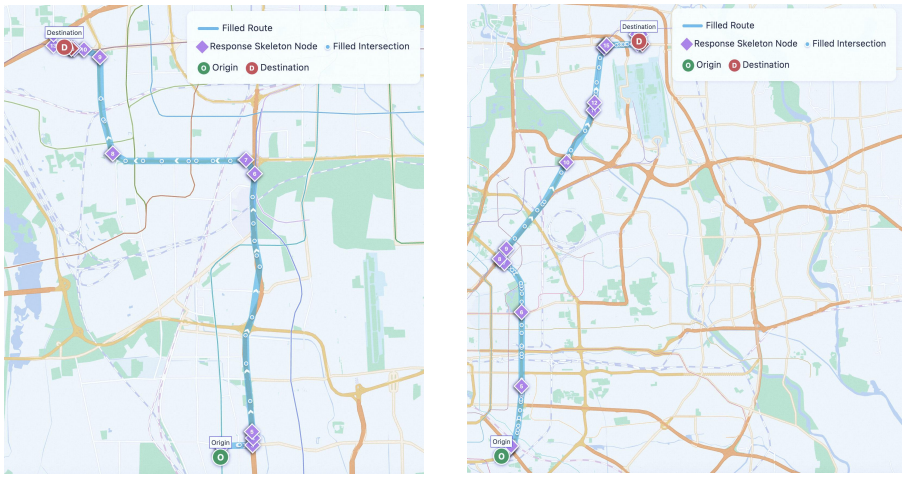

Figure 3: Two representative routes generated by DriveSkel-LM.