

RoadWeaver: Large-Scale Lane-Level HD Map Generation from Scratch for Autonomous Driving Simulation

Yueyuan Li, Zexi Chen, Weijie Xi, Mingyang Jiang, Songan Zhang, Hanyang Zhuang, and Ming Yang

Abstract—Autonomous driving simulation requires diverse and scalable lane-level HD maps to support long-horizon evaluation across complex road networks. Existing approaches either rely on handcrafted or reconstructed real-world maps, which limits scalability, or generate only local road structures rather than complete HD maps. We present RoadWeaver, a coarse-to-fine framework for from-scratch generation of diverse, large-scale HD maps. RoadWeaver first synthesizes a global road layout, expands it into a connected road network, and then constructs lane-level geometry with topologically consistent lane connectivity. Experimental results show that RoadWeaver achieves a 99.8% reachability, a 10.7% dead-end ratio, and an endpoint alignment error of 0.24 m. Compared with SOTA generation methods, it reduces endpoint alignment error by 94.4% while generating complete HD maps in 1.39–3.50 s. The generated maps can be directly deployed in driving simulators, providing scalable simulation environments for future closed-loop evaluation of autonomous driving systems. The training code and an out-of-the-box implementation of RoadWeaver will be released upon acceptance.

Index Terms—Simulation; Autonomous Driving; Test and Evaluation; HD Map Generation; Road Layout Generation

I. INTRODUCTION

Simulation has become an indispensable tool for the development and evaluation of autonomous driving systems (ADS) [1]. Widely adopted benchmarks, such as the CARLA Leaderboard and Bench2Drive, have highlighted the importance of evaluating ADS over long-horizon closed-loop routes rather than isolated scenario fragments [2, 3]. Such evaluations expose planning failures that accumulate over time, including inappropriate lane selections, route deviations, and traffic-rule violations that may remain hidden in short, independent scenarios. Consequently, long-horizon evaluation demands road maps that are not only large-scale and geographically diverse, but also sufficiently detailed to accurately represent lane-level geometry and connectivity.

Unlike conventional road networks designed primarily for traffic-flow analysis, high-definition (HD) maps provide rich geometric, semantic, and topological information that is essential for ADS evaluation. Among these properties, road-network topology plays a crucial role in shaping driving behavior. To better understand this relationship, we analyze trajectory dynamics, including velocity, acceleration, steering, and other motion characteristics, from multiple autonomous driving datasets [4–13]. As illustrated in Fig. 1, trajectories from different road topologies exhibit distinct behavioral distributions, indicating that topology influences driving and interaction patterns. These observations suggest that increasing the diversity of road-network topology is crucial for exposing a broader spectrum of driving behaviors and improving the structural coverage of long-horizon ADS evaluation.

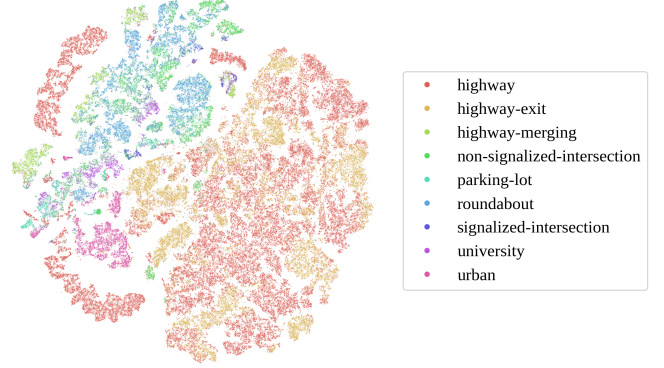


Fig. 1. t-SNE visualization of driving behavior analysis collected from multiple autonomous driving datasets, colored by scenario type.

Existing simulation maps are primarily obtained through three approaches: handcraft [2, 14], reconstructed from real-world [10, 11], and automatic generation [15, 16]. The first two approaches suffer from limited scalability: manually designed maps require substantial engineering effort, whereas reconstructed HD maps are restricted to existing geographic regions and their inherent topological characteristics. Automatic map generation offers a scalable alternative for constructing large-scale driving environments. However, existing methods still struggle to simultaneously achieve global topological coherence, lane-level geometric validity, controllable road morphology, and direct compatibility with mainstream simulation platforms. These limitations make it difficult to generate diverse road networks suitable for systematic long-horizon ADS evaluation.

To address these challenges, we propose RoadWeaver, a framework for generating diverse, large-scale lane-level HD maps from scratch. RoadWeaver constructs globally consistent road networks while preserving lane-level geometric correctness and enabling flexible control over road morphology. The generated maps can be exported directly to OpenStreetMap (OSM) and OpenDRIVE formats [17, 18], allowing seamless integration with existing autonomous driving simulators. We further integrate RoadWeaver into Tactics2D [19], providing an out-of-the-box map generation pipeline for scalable simulation and customized long-horizon ADS evaluation.

The contributions of this work are summarized as follows:

- We propose RoadWeaver, a from-scratch framework for generating diverse lane-level HD maps with valid road-network topology and lane geometry.
- RoadWeaver supports large-scale map generation by con-

TABLE I
COMPARISON OF REPRESENTATIVE METHODS FOR AUTONOMOUS DRIVING MAP PREPARATION.

Method	Diversity		Scale Large-scale	Usability		
	From Scratch	Global Topology		Lane-level	Controllable	Simulation-ready
CARLA (2017) [2]	✗	✓	✗	✓	✗	✓
Highway-env (2020) [14]	✗	△	✗	△	△	✓
HDMaGen (2021) [16]	✓	△	✗	✓	✗	✗
HDMaNet (2022) [20]	✗	✗	✗	✓	✗	✗
MapTR (2022) [21]	✗	✗	✗	✓	✗	✗
VectorMapNet (2023) [22]	✗	✗	✗	✓	✗	✗
MetaDrive (2022) [15]	✓	✓	✓	△	✓	✓
RoadGen (2024) [23]	✓	✓	△	△	✓	△
ControlMap (2026) [24]	✓	△	✗	✓	✓	✗
RoadWeaver	✓	✓	✓	✓	✓	✓

From Scratch: generate maps without existing maps or sensor observations.; Global Topology: generate globally consistent road networks;

Lane-level: support HD-map lane-level representations; Large-scale: generate large-scale maps;

Controllable: support explicit user control over map generation; Simulation-ready: directly deployable in driving simulators.

✓: fully supported; △: partially supported under restricted settings (e.g., predefined rules, local generation); ✗: not supported.

trolling the spatial extent while maintaining a target road-network density.

- RoadWeaver is simulation-ready, exporting maps directly to OSM and OpenDRIVE formats and integrating seamlessly with Tactics2D for closed-loop ADS evaluation.

II. RELATED WORKS

Existing approaches to preparing maps for autonomous driving simulation can be broadly categorized into three groups: manual construction, reconstruction from real-world observations, and automatic generation. The last category can be further divided into rule-based and data-driven methods. Table I summarizes the capabilities of representative methods.

Many driving simulators rely on maps that are manually designed and assembled. CARLA and Highway-env, for example, provide handcrafted road environments for controlled and reproducible experiments [2, 14]. Such maps are effective for benchmark development, but extending them to large road networks requires substantial engineering effort. Their structural diversity is also limited by the available map assets and manual design process.

Another line of work constructs HD maps from real-world observations. Traditional reconstruction pipelines recover lane geometry and semantic elements from vehicle-mounted or aerial sensor data [25]. Recent learning-based methods, including HDMaNet [20], MapTR [21], and VectorMapNet [22], predict vectorized map elements from multi-view images or LiDAR observations. These approaches achieve high geometric fidelity but are restricted to existing road networks. Their reliance on data collection and preprocessing also makes generating novel topologies and large-scale maps costly.

Rule-based generation methods combine road networks with predefined road elements under geometric or topological constraints. MetaDrive [15] assembles road primitives to create configurable driving environments, while RoadGen [23] connects parameterized road components subject to geometric and connectivity rules. These methods are efficient and offer direct control over map attributes. Nevertheless, the generated structures remain constrained by manually specified primitives and composition rules. Increasing map scale or topology

diversity often requires more templates and rules, and the resulting lane geometry may lack the variation as in real maps.

Data-driven methods instead learn the distribution of road structures from existing maps. HDMaGen [16] models HD maps using a hierarchical graph generative framework, and ControlMap [24] introduces spatial conditions to improve generation controllability. Compared with procedural approaches, learned models can capture richer structural patterns without explicitly encoding every road configuration. However, existing methods are mainly designed for local map synthesis or bounded map patches. They do not construct complete large-scale lane-level maps with consistent connectivity, nor do they ensure that the generated maps can be readily used in closed-loop driving simulation.

Overall, existing methods improve different aspects of map generation, but none simultaneously supports large-scale, simulation-ready HD map generation for comprehensive autonomous driving system evaluation.

III. METHOD

A. Overview

RoadWeaver adopts a coarse-to-fine generation strategy, in which a road network is constructed from global topology to lane-level geometry. As illustrated in Fig. 2, the proposed framework comprises three stages: global road skeleton generation, skeleton-guided road graph expansion, and topology-aware HD map construction. The first stage generates a sparse road skeleton that captures the principal structure of the target road network. Based on this skeleton, branch and local roads are subsequently introduced to improve geometric details. The refined road graph is converted into a HD map while maintaining valid connectivity and topology.

RoadWeaver controls map scale by adjusting the spatial extent while maintaining a target road-network density. By maintaining a target density while expanding the generation area, the number of generated road nodes can naturally increase without modifying the generation procedure.

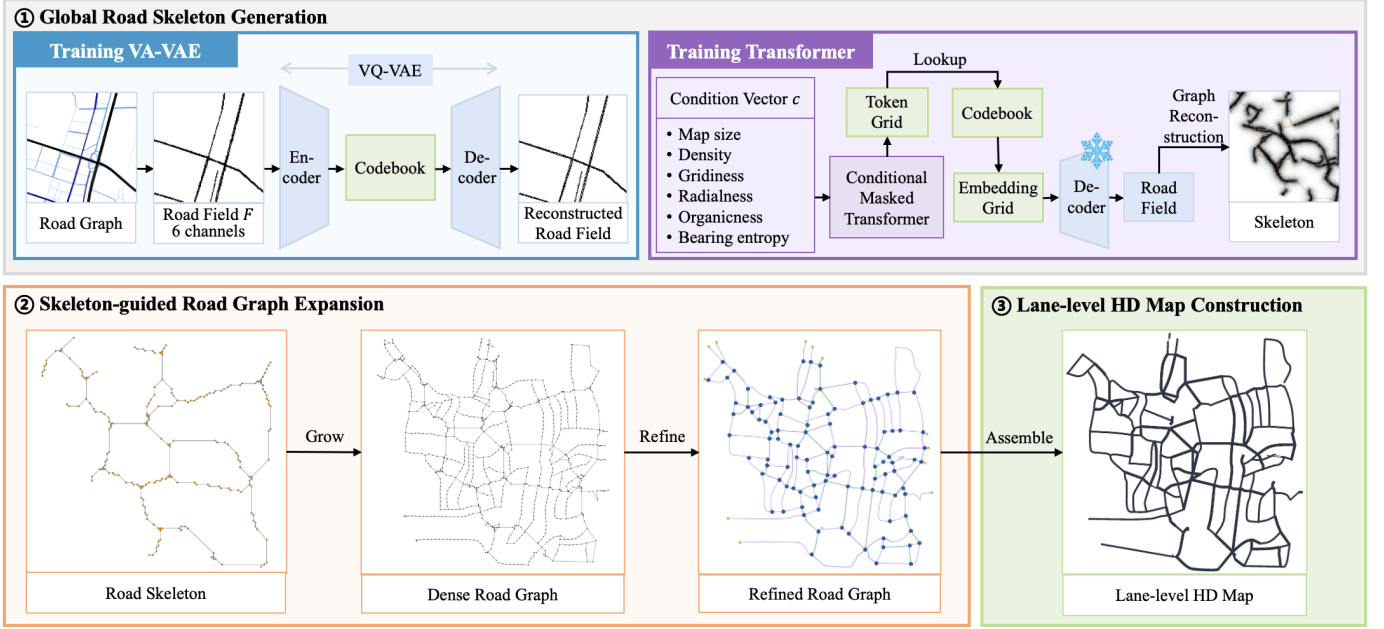


Fig. 2. An overview of the map generation pipeline of RoadWeaver.

B. Global Road Skeleton Generation

Instead of generating all road segments simultaneously, we construct a sparse road skeleton that preserves the principal topology. At this stage, geometric details are intentionally omitted, while long-range connectivity and the overall road layout are retained. This decomposition reduces the complexity of the initial generation task and enables local roads to be added in subsequent stages without substantially changing the global network organization.

Since irregular road graphs cannot be directly represented on a regular spatial grid, we rasterize each graph into a road-field tensor $\mathbf{F} \in \mathbb{R}^{H \times W \times 6}$, where H and W denote the spatial resolution. Its six channels consist of one road probability channel, two orientation channels, one junction heatmap, one endpoint heatmap, and one distance field. These channels preserve the geometric and topological cues for subsequent graph reconstruction. During training, $\mathbf{F}$ is constructed from the ground-truth road graph extracted from OSM.

Because the objective of this stage is to model the global road topology, we represent the road field in a compact discrete latent space for simplicity and efficiency. Specifically, we employ a VQ-VAE [26, 27] to encode $\mathbf{F}$ into a grid of discrete latent tokens and reconstruct the road field from the decoded tokens. This discrete representation converts continuous field prediction into categorical token prediction, retaining the dominant topological structure while suppressing geometric details. The VQ-VAE is optimized using reconstruction loss together with codebook and commitment losses.

To control the characteristics of the skeleton, we define a conditioning vector $\mathbf{c} \in \mathbb{R}^{11}$ consisting of a six-dimensional road-style code together with five structural priors: road density, gridness, radialness, organicness, and bearing entropy. A conditional masked Transformer [28] is trained to model the

distribution of latent tokens conditioned on $\mathbf{c}$. Road layouts exhibit strong long-range spatial dependencies, and the configuration of a local region depends on structures distributed across the entire map. Masked token prediction can exploit bidirectional context and refine multiple regions in parallel, enabling more globally coherent layouts and more efficient sampling than autoregressive generation. We use AdaLN conditioning [29] to inject $\mathbf{c}$ into each Transformer block.

During inference, the sampled latent token grid is decoded by the VQ-VAE to recover the road field $\mathbf{F}$. Road centerlines are then extracted from the decoded field and converted into a vectorized road skeleton through graph reconstruction.

C. Skeleton-Guided Road Graph Expansion

The generated skeleton specifies the main connectivity and large-scale organization of the road network, but does not provide sufficient local-road detail for lane-level map construction. We expand the generated skeleton using a procedural road-growth strategy [30].

Following the structure tensor formulation [31], we construct a continuous directional tensor field from tangent samples collected along the generated skeleton. For each spatial location $\mathbf{p}$, the tensor is defined as

$$\mathbf{T}(\mathbf{p}) = \frac{\sum_{i \in \mathcal{N}(\mathbf{p})} g_i \mathbf{t}_i \mathbf{t}_i^\top}{\sum_{i \in \mathcal{N}(\mathbf{p})} g_i} \quad (1)$$

$$g_i = \exp\left(-\frac{\|\mathbf{q}_i - \mathbf{p}\|_2^2}{2\sigma^2}\right). \quad (2)$$

Here, $\mathbf{T}(\mathbf{p}) \in \mathbb{R}^{2 \times 2}$ is a symmetric positive-semidefinite matrix, $\mathbf{t}_i \in \mathbb{R}^2$ is the unit tangent direction associated with the i -th sampled skeleton point $\mathbf{q}_i$, and $\mathcal{N}(\mathbf{p})$ denotes the local neighborhood of $\mathbf{p}$. The coefficient g_i represents the

spatial contribution of each sample, where σ controls the decay bandwidth of the Gaussian kernel. A separate neighborhood radius determines which samples are included in $\mathcal{N}(\mathbf{p})$.

Road growth is initialized from seed points sampled on both sides of the skeleton. Let t denote the current propagation step, where $\mathbf{p}_t \in \mathbb{R}^2$ and $\mathbf{d}_t \in \mathbb{R}^2$ denote the position and unit propagation direction of an active growth front, respectively. We define

$$\mathbf{a}(\mathbf{p}_t, \mathbf{d}_t) = \arg \max_{\mathbf{v} \in \{\pm \mathbf{e}_{\text{major}}(\mathbf{p}_t), \pm \mathbf{e}_{\text{minor}}(\mathbf{p}_t)\}} \mathbf{v}^\top \mathbf{d}_t, \quad (3)$$

where $\mathbf{e}_{\text{major}}(\mathbf{p}_t)$ and $\mathbf{e}_{\text{minor}}(\mathbf{p}_t)$ are the major and minor eigenvectors of $\mathbf{T}(\mathbf{p}_t)$, respectively, and $\mathbf{a}(\mathbf{p}_t, \mathbf{d}_t)$ denotes the eigen-direction of $\mathbf{T}(\mathbf{p}_t)$ that is most closely aligned with the current propagation direction.

The deterministic candidate direction is computed as

$$\tilde{\mathbf{d}}_{t+1} = \mathcal{C}_{\theta_{\max}} \left(\frac{w_{\text{tensor}} \mathbf{a}(\mathbf{p}_t, \mathbf{d}_t) + w_{\text{inertia}} \mathbf{d}_t}{\|w_{\text{tensor}} \mathbf{a}(\mathbf{p}_t, \mathbf{d}_t) + w_{\text{inertia}} \mathbf{d}_t\|_2} \right), \quad (4)$$

where w_{tensor} and w_{inertia} are scalar weights for tensor guidance and directional inertia, respectively. The operator $\mathcal{C}_{\theta_{\max}}$ limits the angular deviation from $\mathbf{d}_t$ to a prescribed maximum turning angle $\theta_{\max}$.

The final propagation direction $\mathbf{d}_{t+1}$ is obtained by applying a style-dependent angular perturbation $\Delta\phi$ to $\tilde{\mathbf{d}}_{t+1}$. The position of the growth front is then advanced using a first-order Euler step:

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \Delta s \mathbf{d}_{t+1}, \quad (5)$$

where Δs denotes the propagation step length.

The propagation of an active front terminates when it leaves the map boundary, reaches the prescribed maximum length or iteration limit, or approaches an existing road within the snapping radius. In the latter case, the newly generated road is attached to the existing graph.

After procedural growth, dangling endpoints are reconnected to nearby roads through A* search [32] over a cost map derived from the decoded road field $\mathbf{F}$. Regions with stronger predicted road evidence are assigned lower traversal costs, encouraging the reconnection paths to remain consistent with the spatial evidence predicted by the generative model. Residual disconnected fragments are subsequently removed through largest-connected-component filtering.

Once the principal road graph has been formed, large unoccupied regions are detected and further populated with local roads. These additional roads are generated using the same structure tensor field $\mathbf{T}$, which helps maintain directional consistency with the surrounding road network.

D. Lane-Level HD Map Construction

The final stage of RoadWeaver transforms the refined road graph into a lane-level HD map through a sequence of graph transformation and lane construction operations. As summarized in Algorithm 1, the road graph is first refined through geometric regularization, topology simplification, and junction-aware processing. Lane configurations C are then assigned to individual road segments, from which lane boundaries B and lane instances L are generated. Junction connector lanes are

generated and lane-level topology is subsequently established to construct a directed lane graph. Finally, geometric conflicts are repaired, the HD map is assembled, and a topology repair pass is applied to recover disconnected lane connections, producing the final lane-level HD map.

Algorithm 1 Lane-Level HD Map Construction Pipeline

Require: Semantic road graph $G = (V, E)$ with geometric and road attributes
Ensure: Lane-level HD map M with lane geometries and topology

- 1: $G \leftarrow \text{REFINEROADGRAPH}(G)$
- 2: $C = \{c_e\}_{e \in E} \leftarrow \text{ASSIGNLANECONFIG}(G)$
- 3: **for** each road segment $e \in E$ **do**
- 4: $B_e \leftarrow \text{GENERATELANEBOUNDARIES}(e, c_e)$
- 5: $L_e \leftarrow \text{CONSTRUCTDIRECTIONALLANES}(B_e)$
- 6: **end for**
- 7: **for** each junction J **do**
- 8: $L_J^{\text{con}} \leftarrow \text{CONSTRUCTJUNCTIONCONNECTIONS}(J, L)$
- 9: **end for**
- 10: $L \leftarrow \text{MERGELANEINSTANCES}(\{L_e\}, \{L_J^{\text{con}}\})$
- 11: $L \leftarrow \text{ADDSUCCESSORLINKS}(L)$
- 12: $L \leftarrow \text{REPAIRLANEGEOMETRY}(L)$
- 13: $M \leftarrow \text{ASSEMBLEHDMAP}(L)$
- 14: $M \leftarrow \text{REPAIRTOPOLOGY}(M)$
- 15: **return** M

IV. EXPERIMENTS

A. Experiment Setup

1) *Dataset:* We train RoadWeaver on real-world road networks collected from OSM, comprising approximately 58,000 samples from 144 cities. Each urban road-network sample covers an area of approximately $2 \text{ km} \times 2 \text{ km}$. We define road-node density as the number of road nodes per km^2 under a consistent graph extraction procedure and use it as a controllable measure of road density. During inference, the generation area can be flexibly adjusted according to the desired simulation scale. By maintaining the target road-node density, larger spatial regions naturally result in larger road networks with more nodes while preserving consistent structural characteristics.

2) *Baselines:* We compare RoadWeaver with representative rule-based and data-driven map generation methods. MetaDrive [15] and RoadGen [23] are adopted as rule-based baselines, while HDMaGen [16] is included as a data-driven baseline for local lane-level map generation.

3) *Hardware:* All experiments are conducted on a workstation equipped with an NVIDIA GeForce RTX 5090 GPU, an AMD Ryzen 7 9800X3D CPU, and 64 GB of system RAM. The reported generation time covers the complete map-generation pipeline and is averaged over multiple independent runs, excluding model-loading and warm-up overhead.

B. Generation Quality, Diversity, and Controllability

We first compare the generation quality of RoadWeaver with representative existing methods under a comparable map scale.

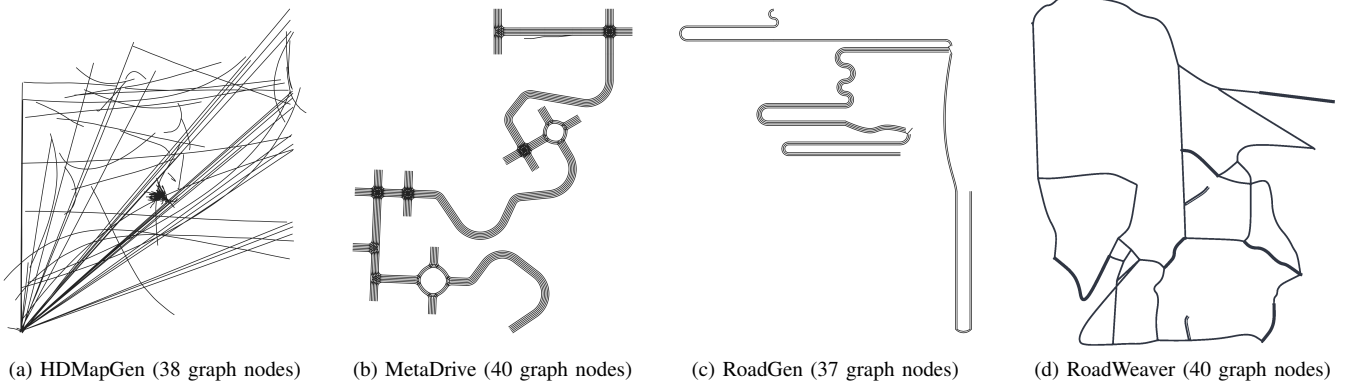


Fig. 3. Representative maps generated by different methods at a comparable map scale (35–40 graph nodes). Intersections, road endpoints, and turning transition points are counted as graph nodes.

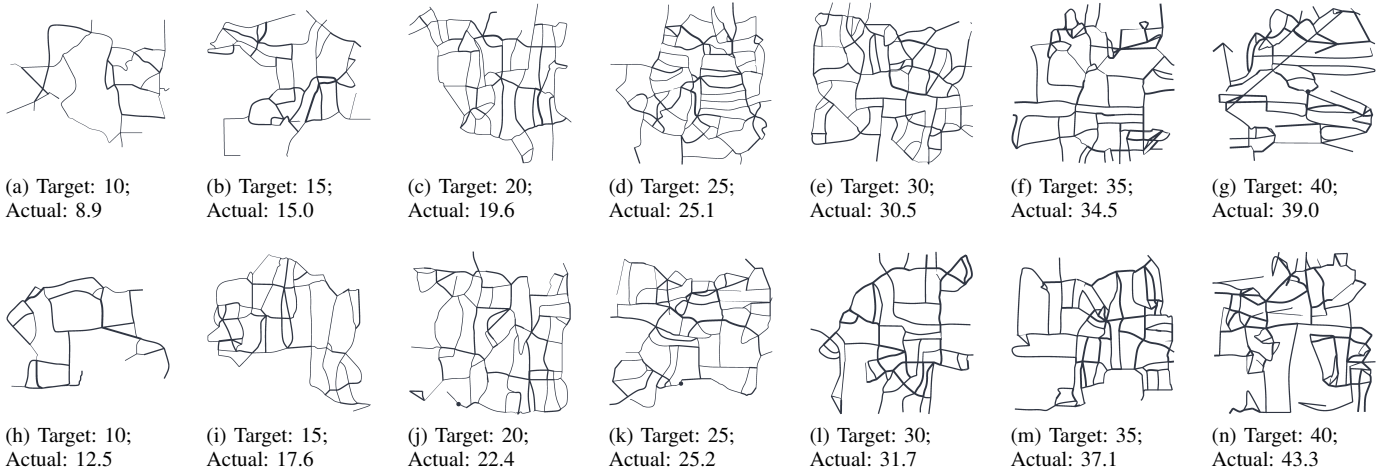


Fig. 4. Demonstration of RoadWeaver's controllability and diversity under different road-node density conditions. The road-node density is defined as the number of road nodes per km^2 .

All methods are configured to generate road graphs containing approximately 35–40 nodes.

As shown in Fig. 3, RoadWeaver generates road networks with coherent topology and geometry across different regions of the map. Specifically, RoadWeaver maintains connectivity among road segments while avoiding disconnected components and abrupt transitions observed in baseline methods. The generated networks also preserve lane-level structures and intersection configurations, demonstrating the ability to jointly model road topology and geometric layouts.

Beyond generation quality, Fig. 4 illustrates maps generated with different random seeds under the same target density. The paired examples show $2\text{ km} \times 2\text{ km}$ maps generated with target densities of 10, 15, 20, 25, 30, 35, and 40 road nodes/ km^2 . Despite having similar measured densities, these maps exhibit distinct topological organizations, intersection configurations, and local geometric patterns. These results demonstrate that RoadWeaver can generate varied road structures under the same density constraint while preserving consistent network-level statistics.

The same figure also demonstrates the controllability of RoadWeaver over road-network density. By increasing the tar-

get density, the generated maps become progressively denser while maintaining connectivity and lane-level consistency. The measured road-node densities remain close to the specified values, confirming that the density constraint can be effectively reflected in the generated networks. When the target density exceeds approximately 40 road nodes/ km^2 , the density increase becomes less significant due to the increasing topological constraints of dense road networks.

C. Topological and Geometric Validity

Following previous road-network generation and HD map modeling studies [16, 21–23], we evaluate the generated maps from topological and geometric perspectives using five metrics. Specifically, the LCC ratio and reachability assess graph connectivity, the dead-end ratio and cycle ratio characterize network structure, and the endpoint alignment error evaluates lane-level geometric consistency. The LCC ratio denotes the proportion of nodes in the largest connected component [33]. Reachability is the fraction of node pairs connected by valid paths [16, 23, 33]. The dead-end ratio quantifies the proportion of terminal nodes [33]. The cycle ratio measures the proportion of nodes belonging to at least

TABLE II
TOPOLOGICAL AND GEOMETRIC VALIDITY OF GENERATED ROAD NETWORKS.

Method	LCC (%) $\uparrow$	Reachability (%) $\uparrow$	Dead-end Ratio (%) $\downarrow$	Cycle Ratio (%) $\uparrow$	Endpoint Alignment Error (m) $\downarrow$
MetaDrive	99.5 ± 2.3	99.3 ± 3.3	43.0 ± 10.1	50.5 ± 13.8	8.17 ± 2.43
RoadGen	100.0 ± 0.0	100.0 ± 0.0	12.8 ± 6.2	0.0 ± 0.0	4.80 ± 1.29
HDMaGen	64.3 ± 24.7	56.2 ± 28.2	42.9 ± 29.4	46.5 ± 37.2	4.32 ± 0.41
RoadWeaver	99.9 ± 0.3	99.8 ± 0.7	10.7 ± 3.6	85.2 ± 6.0	0.24 ± 0.22

one cycle [33]. The endpoint alignment error is defined as the average Euclidean distance between lane endpoints that are expected to be connected [21, 22].

Table II summarizes the results. RoadWeaver achieves an LCC of 99.9% and a reachability of 99.8%, showing that most generated road segments belong to a connected road network. It also obtains the lowest dead-end ratio among all compared methods, indicating fewer disconnected road extensions.

The cycle ratio shows a clear difference among the generated graph structures. RoadWeaver reaches 85.2%, compared with 50.5% for MetaDrive, 46.5% for HDMaGen, and 0.0% for RoadGen. RoadWeaver produces road networks with more closed-loop structures, resulting in more connected road blocks and multiple routing possibilities.

RoadWeaver also achieves the smallest endpoint alignment error, with only 0.24 m, while the baselines range from 4.32 m to 8.17 m. The smaller error indicates that lane endpoints remain better aligned during graph expansion, resulting in more consistent road connections.

These results show that RoadWeaver can generate large-scale HD maps from scratch while maintaining graph connectivity and lane-level geometric consistency.

D. Scalability and Efficiency

We compare the computational efficiency of all methods with respect to the number of generated road-network nodes, which provides a unified measure since different methods define map scale differently.

The scalability of the baselines is constrained by their generation mechanisms. HDMaGen does not impose an explicit limit on scale, but as is shown in Figure 3, larger graphs provide limited quality improvement. RoadGen is limited by geometric feasibility. Once the available connection queue is exhausted, the assembly process terminates early, so valid maps rarely extend beyond roughly 60 nodes. MetaDrive scales by increasing the predefined map configuration, which leads to rapidly rising resource consumption as the map grows. In contrast, RoadWeaver scales by enlarging the map area while maintaining a bounded road density, so the number of generated nodes can grow naturally without changing the generation procedure or relying on a fixed map template.

Figure 5 shows the generation time and CPU consumption as the number of graph nodes increases. HDMaGen can increase graph size, but larger graphs provide limited quality improvement. RoadGen maintains strong geometric constraints, but its generation time increases from approximately 50 s to 370 s, making large-scale expansion impractical.

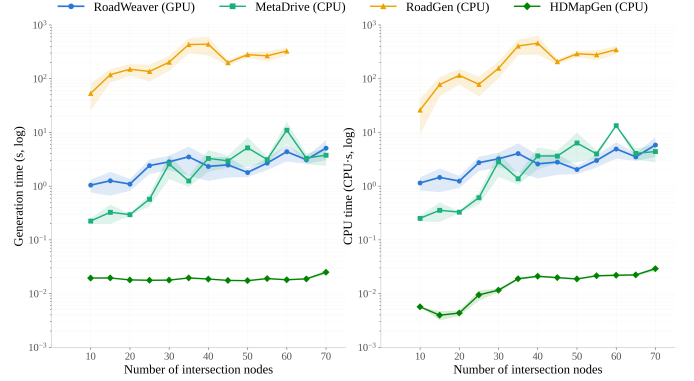


Fig. 5. Generation time and CPU consumption of different methods as the number of graph nodes increases. CPU consumption is measured in CPU-seconds, calculated as the product of the average CPU utilization and the wall-clock generation time.

Compared with MetaDrive, RoadWeaver achieves comparable runtime while offering better scalability and road-network quality. Specifically, RoadWeaver generates maps within approximately 1.39–3.50 s.

E. Simulator Deployment Validation

To validate the practical usability of the generated HD maps, we integrate RoadWeaver into Tactics2D [19]. The generated maps are compatible with OSM and OpenDRIVE representations and can be imported into Tactics2D without manual conversion. Tactics2D is a lightweight open-source simulator that supports CPU-based closed-loop autonomous driving evaluation and provides lane-level HD map representation, route planning, vehicle models, driving behaviors, and benchmark datasets. These characteristics make it an appropriate platform for verifying whether RoadWeaver can be directly deployed for autonomous driving simulation.

We randomly generate 100 lane-level HD maps covering diverse road layouts and network densities. Each generated map contains complete lane-level topology, including predecessor, successor, and neighboring lane relationships, and is directly imported into Tactics2D without any manual editing or post-processing. All generated maps are successfully imported into the simulator, achieving an import success rate of 100.0%. We further perform 10 randomly sampled routing tasks on each map, resulting in a total of 1,000 route-planning tasks. The built-in route planner successfully generates valid paths for most of the tasks, corresponding to a routing success rate of 98.7%. The average routing time is about 0.82 ± 0.17 s.

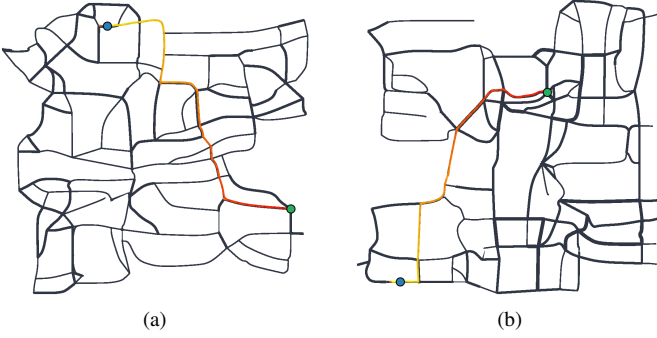


Fig. 6. Routing sample of generated maps from RoadWeaver in Tactics2D.

Representative generated maps and their corresponding routing results are shown in Fig. 6.

These results demonstrate that the generated HD maps are not only geometrically and topologically valid, but also directly deployable for downstream autonomous driving simulation and route-planning tasks without manual intervention.

V. CONCLUSION

This paper presented RoadWeaver, a coarse-to-fine framework for generating large-scale lane-level HD maps from scratch. Unlike existing methods, RoadWeaver generates simulation-ready maps while preserving road topology and lane-level geometry. Experiments show that the generated maps are diverse, scalable, and directly compatible with existing driving simulators, making them suitable for large-scale closed-loop evaluation of ADS. Future work will focus on richer traffic semantics, roadside assets, and interactive map editing, and further improve the realism of generated driving environments.

REFERENCES

- [1] Y. Li, W. Yuan, S. Zhang, W. Yan, Q. Shen, C. Wang, and M. Yang, “Choose your simulator wisely: A review on open-source simulators for autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 5, pp. 4861–4876, 2024.
- [2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [3] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan, “Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 819–844, 2024.
- [4] B. Coifman and L. Li, “A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset,” *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 2017.
- [5] R. Krajewski, J. Bock, L. Kloecker, and L. Eckstein, “The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems,” in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2118–2125.
- [6] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, “The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1929–1934.
- [7] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, “The round dataset: A drone dataset of road user trajectories at roundabouts in germany,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.
- [8] J. Bock, L. Vater, R. Krajewski, and T. Moers, “Highly accurate scenario and reference data for automated driving,” *ATZ worldwide*, vol. 123, no. 5, pp. 50–55, 2021.
- [9] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, “Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps,” *arXiv preprint arXiv:1910.03088*, 2019.
- [10] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles,” *arXiv preprint arXiv:2106.11810*, 2021.
- [11] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.
- [12] X. Shen, M. Lacayo, N. Guggilla, and F. Borrelli, “Parkpredict+: Multimodal intent and motion prediction for vehicles in parking lots with cnn and transformer,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 3999–4004.
- [13] O. Zheng, M. Abdel-Aty, L. Yue, A. Abdelraouf, Z. Wang, and N. Mahmoud, “Citysim: A drone-based vehicle trajectory dataset for safety-oriented research and digital twins,” *Transportation research record*, vol. 2678, no. 4, pp. 606–621, 2024.
- [14] E. Leurent, “An environment for autonomous driving decision-making,” <https://github.com/eleurent/highway-env>, 2018, accessed Mar. 12, 2024.
- [15] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, “Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3461–3475, 2022.
- [16] L. Mi, H. Zhao, C. Nash, X. Jin, J. Gao, C. Sun, C. Schmid, N. Shavit, Y. Chai, and D. Anguelov, “Hdmapgen: A hierarchical graph generative model of high definition maps,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4227–4236.
- [17] M. Haklay and P. Weber, “Openstreetmap: User-

- generated street maps,” *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [18] M. Dupuis, M. Strobl, and H. Grezlikowski, “Opendriven 2010 and beyond—status and future of the de facto standard for the description of road networks,” in *Proc. of the Driving Simulation Conference Europe*, 2010, pp. 231–242.
- [19] Y. Li, S. Zhang, M. Jiang, X. Chen, J. Yang, Y. Qian, C. Wang, and M. Yang, “Tactics2d: A highly modular and extensible simulator for driving decision-making,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 5, pp. 4840–4844, 2024.
- [20] Q. Li, Y. Wang, Y. Wang, and H. Zhao, “Hdmapnet: An online hd map construction and evaluation framework,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4628–4634.
- [21] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, “Maptr: Structured modeling and learning for online vectorized hd map construction,” *arXiv preprint arXiv:2208.14437*, 2022.
- [22] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, “Vectormapnet: End-to-end vectorized hd map learning,” in *International conference on machine learning*. PMLR, 2023, pp. 22 352–22 369.
- [23] F. Yang, Y. Lu, B. Chen, P. Qin, and X. Peng, “Roadgen: Generating road scenarios for autonomous vehicle testing,” *arXiv preprint arXiv:2411.19577*, 2024.
- [24] M. Farag, S. Wäldele, and Y. Yao, “Controlmap: Controllable high-definition map generation for traffic scenario simulation,” *arXiv preprint arXiv:2606.15930*, 2026.
- [25] L. Liao, W. Yan, W. Xu, M. Yang, S. Zhang, and H. E. Tseng, “Learning-based 3d reconstruction in autonomous driving: A comprehensive survey,” *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [26] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [27] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.
- [28] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, “Maskgit: Masked generative image transformer,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 11 305–11 315.
- [29] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 4172–4182.
- [30] Y. I. Parish and P. Müller, “Procedural modeling of cities,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 301–308.
- [31] J. Bigun and G. H. Granlund, “Optimal orientation detection of linear symmetry,” in *Proceedings of the IEEE First International Conference on Computer Vision*, vol. 54. London, UK, 1987, pp. 433–438.
- [32] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [33] G. Boeing, “Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks,” *Computers, environment and urban systems*, vol. 65, pp. 126–139, 2017.