

DOGE: Differentiable Bézier Graph Optimization for Road Network Extraction

Jiahui Sun¹, Junran Lu¹, Jinhui Yin¹, Yishuo Xu¹, Yuanqi Li¹, and Yanwen Guo^{1*}

State Key Laboratory of Novel Software Technology, Nanjing University, China

Abstract. Automatic extraction of road networks from aerial imagery is a fundamental task, yet prevailing methods rely on discrete polylines that inherently lack the high-order geometric continuity required by downstream applications. While parametric curve representations, such as our Bézier Graph, offer analytic smoothness and topological compactness, their extraction is bottlenecked by the absence of curve-level vector ground-truth (GT) and the fragility of heuristic vectorization pipelines. To overcome these challenges, we introduce *DOGE*, a novel differentiable rendering-driven framework for robust, curve GT-free road network extraction. *DOGE* elegantly decouples the extraction process via a discrete-continuous coupling mechanism: a *DiffAlign* module continuously optimizes the graph’s geometric alignment against target masks using differentiable rendering, while a *TopoAdapt* module applies discrete operators to evolve the network’s topology. This coupling prevents topological bottlenecks from paralyzing the gradient-based geometric optimization. Extensive experiments demonstrate that our method sets a new state-of-the-art on the large-scale SpaceNet and City-Scale benchmarks. Driven purely by mask guidance, *DOGE* successfully reconstructs highly accurate, geometrically smooth, and topologically compact vector maps, advancing a new paradigm for structured road network reconstruction.

1 Introduction

Road network graphs, representing the geometry and topology of drivable roads, are foundational to critical applications like route planning, autonomous driving, and urban modeling [7, 10, 13, 37, 39, 43, 47]. However, manually creating these maps is a costly bottleneck [30, 31, 46]. The growing abundance of high-resolution satellite imagery presents a powerful alternative for rapid, city-scale automated reconstruction.

Most existing methods rely on a polyline representation, approximating road centerlines with sequences of connected line segments, as shown in Fig. 1. However, this discrete approximation introduces inherent constraints: polylines necessitate a high density of vertices to accurately model curves; they are less

* Corresponding author.

intuitive to edit, as local changes can disrupt geometric smoothness; and only guaranteeing positional (C0) continuity [3, 8, 16, 20, 56].

Our motivation is simple: road geometry is inherently curvilinear (Fig. 1). While parametric curves have shown promise in High-Definition (HD) lane mapping [6], their potential for city-scale road networks remains unexplored. Thus, we introduce a parametric Bézier Graph tailored for SD maps, to our knowledge the first parametric representation for SD road networks using cubic Bézier curves. Distinct from the shared parameterization used in HD lane mapping [6], our formulation employs a decoupled, chord-based reparameterization to prevent degeneracies while maintaining the topological flexibility.

As illustrated in Fig. 1, this parametric representation offers a paradigm shift from discrete polylines, delivering advantages across three dimensions: (1) **Analytic Smoothness for Regularization**: Unlike polylines which are inherently discrete and non-differentiable at vertices, Bézier curves are analytically smooth and can be converted to polylines at any arbitrary resolution. This mathematical property [14] empowers our *DiffAlign* module (Sec. 3.2) to enforce geometric priors like G1 continuity that are ill-defined on jagged approximations. (2) **Efficiency and Interactivity**: The representation yields remarkable topological compactness, significantly reducing edge density (Sec. 4.4). Moreover, it enables intuitive geometric editing—drag-and-drop modification of a few control points allows for global shape adjustment without disrupting smoothness [17]. (3) **Readiness for Downstream Standards**: Crucially, continuous curves directly meet the rigorous demands of industry standards. In autonomous driving (e.g., ASAM OpenDRIVE [1]) and urban simulation [12], parametric continuity is essential for defining stable motion planning priors and reducing the control jerk for autonomous vehicles. [35].

A conventional supervised approach to predict the Bézier Graph is bottlenecked by the difficulty of creating curve ground truth (GT). Converting existing annotations into curves is inherently ambiguous, yielding multiple valid factorizations [4, 34] and risking label bias. Furthermore, existing vectorization pipelines [6] rely on brittle, multi-stage heuristics (e.g., node selection, path splitting, and curve fitting) with dataset-dependent thresholds that are difficult to standardize at scale.

To sidestep this reliance on vector GT, one standard post-processing approach is to extract skeletons from segmentation masks and fit curves to them. However, this discrete, heuristic extraction remains brittle and highly sensitive

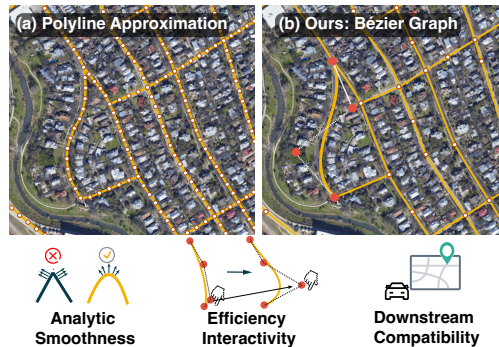


Fig. 1: Polyline versus curve-based road representations. (a) A polyline approximates a curve with discrete segments. (b) A parametric Bézier curve representation. Its geometry is defined by four control points (red).

to mask noise, often yielding fractured topologies (evaluated as *Standard Post-Processing* in our ablation studies, Sec. 4.5). Alternatively, one could use unconstrained differentiable rendering to directly align parametric curves with masks. Yet, without structural priors, this continuous optimization easily collapses into chaotic, self-intersecting geometries (see *Vanilla DiffRender* in Fig. 7).

To overcome the pitfalls of both purely discrete heuristics and unconstrained alignment, we propose *DOGE* (**D**ifferentiable **O**ptimization of Bézier **G**raphs for road network **E**xtraction). Our framework reframes road extraction as a differentiable rendering-driven iterative refinement process. *DOGE* accomplishes this by elegantly decoupling the graph’s continuous geometry and discrete topology into two complementary modules: *DiffAlign* continuously optimizes the geometric alignment via rendering-based gradients, while *TopoAdapt* applies discrete operators to dynamically evolve the topological structure. This discrete-continuous coupling prevents topological bottlenecks during gradient-based optimization. By alternating between them, *DOGE* achieves a holistic refinement of the graph’s shape and connectivity, ultimately extracting high-fidelity road networks directly from segmentation masks without requiring curve GT.

In practice, we employ a fine-tuned SAM2 [22, 33] to provide high-quality segmentation supervision. To the best of our knowledge, *DOGE* is the first to successfully adapt differentiable rendering for vector road extraction in the remote sensing domain, paving the way for future GT-free complex topological reconstruction tasks. Our key contributions are summarized as follows:

- We formulate the Bézier Graph, a curve-based representation for road networks that is topologically compact and analytically smooth. By parameterizing the graph’s spatial geometry within a fully differentiable continuous space, it serves as the mathematical cornerstone that enables geometric optimization and analytic regularization.
- We propose *DOGE*, a novel differentiable rendering-driven framework for robust, curve GT-free road network extraction. By decoupling the extraction into two complementary modules—*DiffAlign* for continuous geometric alignment and *TopoAdapt* for discrete topology refinement—it entirely bypasses the brittle heuristics of traditional pipelines.
- We establish a new state-of-the-art on the large-scale SpaceNet and CityScale benchmarks, validating the robustness of our discrete-continuous design. Furthermore, as a model-agnostic vectorizer driven by mask guidance, *DOGE* is well positioned to continuously benefit from ever-improving visual foundation models.

2 Related Work

2.1 Polyline-based Road Network Modeling

Prior work leveraging polyline representations can be grouped into three main paradigms based on how they construct the graph.

Detection and Connection. These methods first identify road and junctions, then infer connectivity to assemble the graph in a two-stage process [2, 19, 20, 36, 42, 53, 55]. While their specifics vary from node extraction (dense detection, heatmaps, segmentation) to edge inference (pairwise classification, orientation cues, or learned relational reasoning), they all rely on an explicit node-edge decomposition.

Iterative Growth. In contrast, iterative methods reconstruct the network via a sequential decision process, growing the graph from seed points [3, 9, 38, 51, 52]. RoadTracer [3] is a canonical example, using a CNN to guide expansion. While effective at capturing local dependencies, these agent-based models can accumulate errors and are inherently local, as previously generated sections of the graph remain fixed.

One-Shot Reconstruction. This paradigm infers the entire graph in a single forward pass, either by decoding compact topological tensors [18, 54] or by predicting vectorized primitives that are then assembled [2, 21, 50]. For instance, PaLiS [50] predicts patch-wise line segments, while PolyRoad [21] generates polyline road instances with a transformer.

While effective, the polyline approximation often limits geometric fidelity and editability, and often rely on post-processing for vectorization. These limitations motivate our exploration of curve-based representation.

2.2 Curve-based Road Network Modeling

The use of parametric curves for end-to-end SD road network extraction from satellite imagery remains largely unexplored. We therefore turn to the related, finer-grained domain of High-Definition (HD) mapping, where parametric curves like Bézier curves and splines are well-established for modeling precise lane-level geometry [8, 23, 26, 28]. Prior work in this domain has demonstrated various strategies: directly regressing curve control points [15], predicting piecewise segments [32], unifying detection across 2D and 3D modalities [11], and, more recently, forming complete lane graphs with explicit topological constraints [6, 25].

For instance, the lane-level Bézier graphs in [6] enforce geometric smoothness by binding edge tangents to node-shared directions. In contrast, our SD-map formulation offers greater flexibility by optimizing per-edge offsets for internal control points, as detailed in Sec. 3.1.

However, these HD mapping techniques are ill-suited for our task. They operate at lane level in relatively constrained driving scenes, whereas we aim to reconstruct topologically complete SD road networks at city scale from satellite imagery. Moreover, this entire line of work fundamentally relies on dense vector GT annotations for supervision [6, 11, 15, 27, 32], which are subjective and prohibitively expensive to obtain at satellite scale [28]. In contrast, we sidestep this dependency by reframing the problem as a differentiable rendering-driven process, enabling the direct alignment of a curve-based road graph with segmentation masks without requiring any curve GT.

2.3 Differentiable Vector Graphics Rasterization

Differentiable rasterization allows gradients to flow from a pixel-based loss back to the parameters of vector graphics, such as the control points and stroke widths of Bézier curves. This is typically achieved by approximating the non-differentiable, hard-edged pixel boundaries of traditional rasterization with a smooth, differentiable function that measures coverage. The canonical implementation, *diffvg* [24], provides stable gradients that enable optimizing vector shapes to match a target image.

This technique has proven effective for learning compact vector representations in various domains. Applications include layer-wise image vectorization [29], vector font synthesis [44, 45], and generative art where Bézier strokes are optimized to create sketches or SVGs guided by semantic models like CLIP or diffusion models [40, 41, 48, 49].

In contrast to prior applications that optimize largely unstructured vector primitives, our work is the first to leverage differentiable rendering for large-scale, topology-aware road network extraction, coupling a structured curve-graph representation with a dynamic optimization process that jointly enforces geometric fidelity and topological correctness.

3 Method

This section details our framework, *DOGE*, which reconstructs road networks by optimizing a Bézier Graph (Sec. 3.1). Our approach decouples this task into two complementary modules: *DiffAlign* (Sec. 3.2), which uses differentiable rendering for continuous geometric optimization, and *TopoAdapt* (Sec. 3.3), which applies discrete operators for topology refinement. These modules form a discrete-continuous refinement loop (Sec. 3.4) that holistically evolves the graph.

3.1 Parametric Representation of Bézier Graph

We formally model the road layout as a Bézier Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The node set, $\mathcal{V} = \{v^k\}$, consists of vertices with optimizable 2D positions, $\mathbf{p}_k \in \mathbb{R}^2$, representing intersections, termini, or points along a road’s path based on their degree. The edge set, $\mathcal{E} = \{e_k\}$, connects these nodes with curvilinear road segments.

As illustrated in Fig. 2, each edge’s geometry is a cubic Bézier curve:

$$\mathbf{C}_k(t) = \sum_{r=0}^3 \binom{3}{r} (1-t)^{3-r} t^r \mathbf{P}_{k,r}, \quad t \in [0, 1]. \quad (1)$$

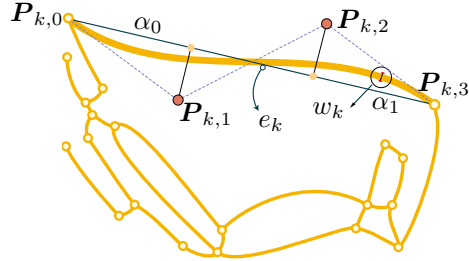


Fig. 2: Parametric definition of a Bézier Graph edge e_k . The edge’s geometry is defined by a cubic Bézier curve with four control points, $\{\mathbf{P}_{k,r}\}_{r=0}^3$, and an optimizable width, w_k .

To ensure a regularized and well-posed representation amenable to optimization, we do not optimize its four control points $\{\mathbf{P}_{k,r}\}_{r=0}^3$ directly. Instead, they are deterministically constructed as follows:

$$\mathbf{P}_{k,0} = \mathbf{p}_i \quad (2a)$$

$$\mathbf{P}_{k,3} = \mathbf{p}_j \quad (2b)$$

$$\mathbf{P}_{k,1} = ((1 - \alpha_{k,0})\mathbf{P}_{k,0} + \alpha_{k,0}\mathbf{P}_{k,3}) + d_{k,0} \cdot \mathbf{n}_{ij} \quad (2c)$$

$$\mathbf{P}_{k,2} = ((1 - \alpha_{k,1})\mathbf{P}_{k,0} + \alpha_{k,1}\mathbf{P}_{k,3}) + d_{k,1} \cdot \mathbf{n}_{ij} \quad (2d)$$

While the endpoints $\mathbf{P}_{k,0}$ and $\mathbf{P}_{k,3}$ are anchored to their corresponding node positions to ensure topological connectivity, we reparameterize the intermediate points $\mathbf{P}_{k,1}$ and $\mathbf{P}_{k,2}$ to reduce the degrees of freedom. As detailed in Eq. 2, their positions are defined by two learnable scalars each: a projection parameter $\alpha_{k,i} \in [0, 1]$ that places a point along the chord $\overline{\mathbf{P}_{k,0}\mathbf{P}_{k,3}}$, and an offset distance $d_{k,i} \in \mathbb{R}$ that displaces it perpendicularly. This reparameterization regularizes the curve’s shape—preventing degeneracies like self-intersections—while maintaining sufficient flexibility. The normal vector $\mathbf{n}_{ij}$ is the normalized perpendicular of the chord vector.

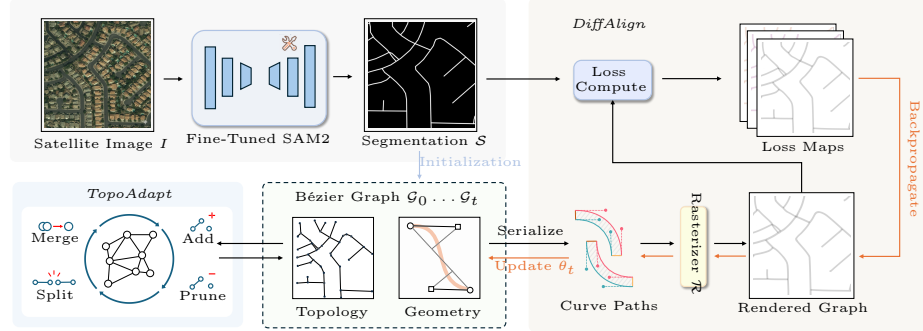


Fig. 3: Overview of the *DOGE* framework. Given a satellite image, a fine-tuned SAM2 provides a target road segmentation $\mathcal{S}$. *DOGE* reconstructs the road network by iteratively optimizing a Bézier Graph $\mathcal{G}$ (Sec. 3.1). The discrete-continuous refinement loop (Sec. 3.4) alternates between two complementary modules: *DiffAlign* (Sec. 3.2), which continuously refines the graph’s geometry by aligning a differentiable rendering of the graph with $\mathcal{S}$, and *TopoAdapt*, which discretely evolves the graph’s topology (Sec. 3.3).

3.2 DiffAlign: Differentiable Geometric Optimization

The geometric optimization is guided by a target segmentation $\mathcal{S}$, obtained from a fine-tuned SAM2 model. As illustrated in Fig. 3, the process begins by serializing each Bézier curve into a closed polygon representing a road segment (details in the supplementary material). These polygons are then passed to a differentiable rasterizer $\mathcal{R}$ (built upon DiffVG [24]) to produce a rendered map $\mathcal{M} = \mathcal{R}(\mathcal{G}_t)$ of the road network.

Gradient Flow via Differentiable Rendering. Because traditional rasterization is non-differentiable, $\mathcal{R}$ employs spatial pre-filtering to compute analytical gradients of pixel coverage. This allows the loss $\mathcal{L}$ to backpropagate into

the graph’s fundamental parameters $\theta = \{\mathbf{p}_k, w_k, \alpha_{k,i}, d_{k,i}\}$ via the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathcal{M}} \frac{\partial \mathcal{M}}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial \theta}, \quad (3)$$

Here, $\frac{\partial \mathcal{M}}{\partial \mathbf{P}}$ captures the gradients computed by the rasterizer $\mathcal{R}$ with respect to the 2D control points $\mathbf{P}$, while $\frac{\partial \mathbf{P}}{\partial \theta}$ distributes them to our decoupled parameters according to the deterministic mapping in Eq. 2. This continuous flow enables precise, sub-pixel alignment of the graph against $\mathcal{S}$.

Objective Function The optimization is driven by a composite objective, $\mathcal{L}_{\text{total}}$, which combines five weighted loss terms. These terms are grouped into a target-alignment loss for data fidelity and four geometric priors that regularize the graph’s structure. The total loss is:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \lambda_{\text{cover}} \mathcal{L}_{\text{cover}} + \lambda_{\text{overlap}} \mathcal{L}_{\text{overlap}} + \lambda_{G1} \mathcal{L}_{G1} \\ & + \lambda_{\text{offset}} \mathcal{L}_{\text{offset}} + \lambda_{\text{spacing}} \mathcal{L}_{\text{spacing}}. \end{aligned} \quad (4)$$

The λ terms are weighting coefficients of each loss components. The specific values are detailed in Sec. 4.1.

Data Fidelity Term The **Coverage Loss** is our data-fidelity term. It penalizes the pixel-wise L2 discrepancy between the target segmentation $\mathcal{S}$ and the union of rendered edges, ensuring the graph covers the road regions:

$$\mathcal{L}_{\text{cover}} = \left\| \left(\bigcup_{e_k \in \mathcal{E}_t} R(e_k) \right) - \mathcal{S} \right\|_2^2 \quad (5)$$

Analytic Geometric Priors To guide the optimization towards a plausible road network, we introduce four regularization priors. These terms operate independently of $\mathcal{S}$ and enforce desirable geometric properties.

First, the **Overlap Loss** promotes correct topology by penalizing the total area of improper intersections between rendered road segments. To compute this, an overlap map is formed by summing all individual edge renderings and clipping the result at a value of 1. The loss is the L1 norm of this map, normalized by the number of edges N_t :

$$\mathcal{L}_{\text{overlap}} = \frac{1}{N_t} \left\| \max \left(0, \left(\sum_{e_k \in \mathcal{E}_t} R(e_k) \right) - 1 \right) \right\|_1 \quad (6)$$

Second, the **G1 Continuity Loss** encourages tangent alignment at degree-2 nodes. For a node v_j connecting edges e_a and e_b , the tangents are defined by vectors $\mathbf{v}_{\text{in}} = \mathbf{p}_j - \mathbf{P}_{a,2}$ and $\mathbf{v}_{\text{out}} = \mathbf{P}_{b,1} - \mathbf{p}_j$. The loss is:

$$\mathcal{L}_{G1} = \frac{1}{N_t} \sum_{\substack{v_j \in \mathcal{V}_t \\ \deg(v_j)=2}} (1 - \cos(\theta_j)) \cdot \mathbf{1}_{(\theta_j < T_{G1})} \quad (7)$$

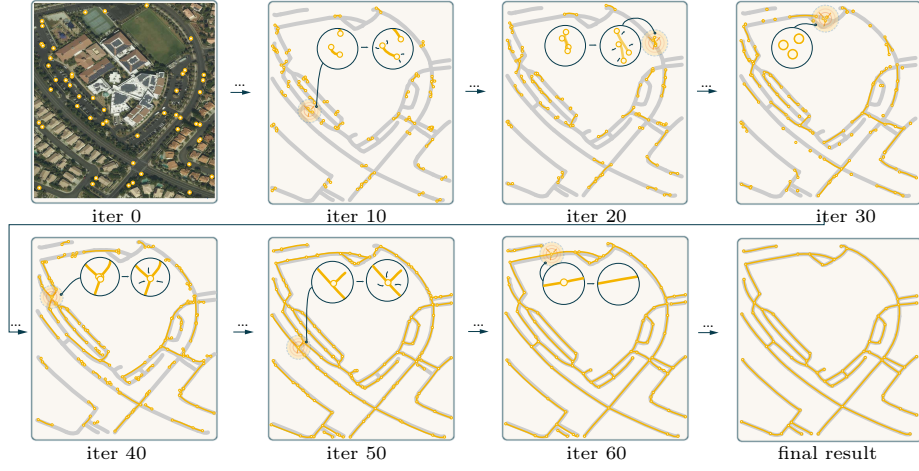


Fig. 4: Discrete-continuous refinement loop of the Bézier Graph. We visualize the alternating optimization between *DiffAlign* and *TopoAdapt*. Key stages include: graph initialization (iter 0); geometric optimization towards the target (iter 10); overlap separation driven by $\mathcal{L}_{\text{overlap}}$ (iter 20); road addition (iter 30); node merging (iter 40); T-junction creation (iter 50); and collinear edge merging (iter 60).

where θ_j is the angle between tangents at node v_j . The loss activates only when θ_j falls below a threshold T_{G1} , penalizing nearly straight connections to preserve legitimate turns.

Finally, two **Curve Regularization** terms penalize ill-formed Bézier geometries. The **Offset Loss** discourages excessive perpendicular offsets of the intermediate control points, while the **Spacing Loss** encourages their projection parameters, $\alpha_{k,0}$ and $\alpha_{k,1}$, to be evenly spaced along the chord connecting the endpoints.

$$\mathcal{L}_{\text{offset}} = \frac{1}{N_t} \sum_{\substack{e_k \in \mathcal{E}_t \\ i \in \{0,1\}}} \max\left(0, \exp\left(\frac{|d_{k,i}|}{L_k} - \tau_d\right) - 1\right) \quad (8)$$

$$\mathcal{L}_{\text{spacing}} = \frac{1}{N_t} \sum_{e_k \in \mathcal{E}_t} ((\alpha_{k,0} - \hat{\alpha}_0)^2 + (\alpha_{k,1} - \hat{\alpha}_1)^2) \quad (9)$$

Here, $L_k = \|\mathbf{P}_{k,3} - \mathbf{P}_{k,0}\|$ is the chord length of edge e_k , τ_d is a threshold for the maximum allowable offset ratio, and the constants $\hat{\alpha}_0, \hat{\alpha}_1$ are set to $1/3$ and $2/3$ respectively, representing the ideal equidistant positions for the control points along the chord.

3.3 TopoAdapt: Structured Discrete Refinement

While *DiffAlign* continuously optimizes geometry, *TopoAdapt* complements it by applying discrete operators to refine the graph topology. We organize *TopoAdapt* into a strictly prioritized pipeline: **Consolidation** $\rightarrow$ **Regularization** $\rightarrow$

Expansion. This ordered execution ensures that topological connectivity is established first, followed by graph simplification to remove artifacts, and finally, structural growth to cover missing regions. These operators are accelerated by a spatial grid and rely on a compact set of physically grounded thresholds fixed across all datasets. A quantitative sensitivity analysis demonstrating this robustness is provided in the supplementary material.

1. Topological Consolidation (Connectivity) To repair fragmentation in the inferred graph, we first prioritize connecting nearby components. This phase follows a specific order to ensure stability:

- **Node Merging:** We first eliminate redundancy by merging pairs of nodes (v_i, v_j) closer than ϵ_{merge} into a single node. This cleans the graph of duplicate endpoints before attempting complex connections.
- **T-Junction Creation:** After merging, we address missing intersections. A node v_i is snapped to a nearby edge e_j if their distance is less than ϵ_{merge} . This splits e_j and creates a valid 3-way intersection, ensuring the network is topologically connected.

2. Geometric Regularization (Simplification) The consolidation phase may introduce redundant intermediate nodes or short artifacts. To maintain a compact representation, we apply regularization operators immediately after connectivity updates:

- **Collinear Edge Merging:** We remove redundant degree-2 nodes that lie on nearly straight paths (tangent angle difference $< \theta_{\text{col}}$). This step is critical to fusing the split edges from T-junctions if they form a straight line, reducing graph complexity without losing accuracy.
- **Invalid Edge Pruning:** We prune geometrically implausible edges (e.g., extremely short or thin edges) and remove isolated nodes, ensuring the gradients in *DiffAlign* propagate only through meaningful structures.

3. Structural Expansion (Road Addition) This operator serves a dual purpose: it initializes the empty graph $\mathcal{G}_0$ from high-confidence regions of $\mathcal{S}$, and in subsequent iterations, identifies missing segments to grow the network. We compute a difference map between the target $\mathcal{S}$ and the current rendering:

$$M_{\text{unfit}} = \mathbf{1}[\mathcal{S} > \tau_{\text{seg}}] \odot \mathbf{1}[\text{Render}(\mathcal{G}_t) < \tau_{\text{render}}]. \quad (10)$$

We sample candidate locations from M_{unfit} and instantiate new edges with random endpoint offsets. These candidate edges are directly appended to the graph, i.e., $\mathcal{G}_t \leftarrow (\mathcal{V}_t \cup \mathcal{V}_{\text{new}}, \mathcal{E}_t \cup \mathcal{E}_{\text{new}})$. This ensures new roads target only genuinely uncovered regions, preventing over-generation in already dense areas.

3.4 Discrete-Continuous Refinement Loop

As illustrated in Fig. 4 and outlined in Algorithm 1 (details in Suppl.), our dynamic graph refinement alternates between *TopoAdapt* and *DiffAlign*. We initialize $\mathcal{G}_0$ via Road Addition on $\mathcal{S}$ to cover regions. To ensure initial geometric stability, *TopoAdapt* activates only after a warmup phase T_{warmup} . Subsequently, *TopoAdapt* refines connectivity $(\mathcal{V}, \mathcal{E})$ and *DiffAlign* optimizes geometry

(θ). Finally, an early-stopping mechanism prevents redundant computation by terminating the loop once the topology stabilizes and rendering loss plateaus (tolerance ϵ) for P epochs.

Algorithm 1: Discrete-Continuous Refinement Workflow of *DOGE*

Input: Target $\mathcal{S}$, Max iter T_{max} , Rate η , Patience P , Tol ϵ
Output: Optimized Bézier Graph $\mathcal{G}^*$

```

1  $\mathcal{G}_0 \leftarrow \text{Init}(\mathcal{S}); k \leftarrow 0; \mathcal{L}_{prev} \leftarrow \infty;$ 
2 for  $t = 0$  to  $T_{max} - 1$  do
3    $(\mathcal{V}'_t, \mathcal{E}'_t) \leftarrow (t > T_{warmup}) ? \text{TopoAdapt}(\mathcal{V}_t, \mathcal{E}_t) : (\mathcal{V}_t, \mathcal{E}_t);$ 
4    $\theta_{t+1}, \mathcal{L}_{curr} \leftarrow \text{DiffAlign}(\mathcal{V}'_t, \mathcal{E}'_t, \theta_t, \mathcal{S}, \eta);$ 
5    $\mathcal{G}_{t+1} \leftarrow (\mathcal{V}'_t, \mathcal{E}'_t, \theta_{t+1});$ 
6   if  $|\mathcal{L}_{prev} - \mathcal{L}_{curr}| < \epsilon \wedge (\mathcal{V}'_t, \mathcal{E}'_t) == (\mathcal{V}_t, \mathcal{E}_t)$  then  $k \leftarrow k + 1;$ 
7   else  $k \leftarrow 0;$ 
8   if  $k \geq P$  then break;
9    $\mathcal{L}_{prev} \leftarrow \mathcal{L}_{curr};$ 
10 end
11 return  $\mathcal{G}^*;$ 

```

4 Experiments

We evaluate on two public benchmarks: City-Scale (Sat2Graph) and SpaceNet. Unless otherwise stated, all images are standardized to 1 m/pixel following prior work, consistent with SAMRoad and SAMRoad++ [20, 53].

City-Scale (Sat2Graph). City-Scale (Sat2Graph) [18] contains 180 RGB satellite tiles at 2048×2048 covering multiple U.S. cities, with road networks provided as vector graphs. We adopt the common split 144/9/27 for train/val/test, identical to Sat2Graph, RNGDet++ [52], and SAMRoad for fair comparison.

SpaceNet. SpaceNet [13] comprises roughly 2.5k RGB tiles at 400×400 from diverse world cities, with vector road-graph annotations. We follow prior work (Sat2Graph/RNGDet++/SAMRoad) and use the 2042/127/382 train/val/test split. Following SAMRoad/SAMRoad++ preprocessing, images are resampled to 1 m/pixel for consistent resolution.

4.1 Implementation Details

Our framework is implemented in PyTorch, with differentiable rendering built upon DiffVG [24]. The optimization is conducted on a single NVIDIA RTX 4090 GPU.

Segmentation Backbone. We use a fine-tuned SAM2 model to generate target road segmentation masks. The model is trained on the respective training splits of each dataset. Further details on the segmentation model’s architecture and training are in the supplementary material.

Optimization. We use the Adam optimizer for geometric parameters. The optimization runs for a maximum of $T_{max} = 300$ iterations, although we employ

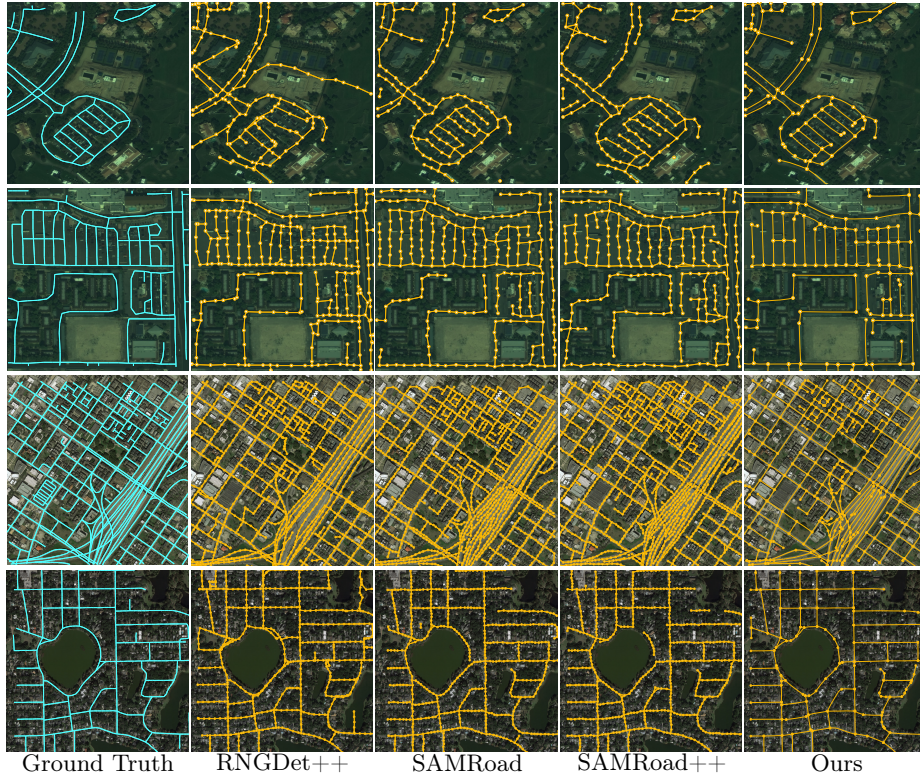


Fig. 5: Qualitative comparison on SpaceNet (top two rows) and City-Scale (bottom two rows). Our method produces geometrically precise, smooth, and topologically correct road graphs, outperforming prior methods across different scales. Notably, our approach uses a significantly more compact graph representation with far fewer nodes.

an early stopping strategy detailed in the supplementary material. Loss weights in Eq. 4 are: $\lambda_{\text{cover}} = 1.0$, $\lambda_{\text{overlap}} = 0.3$, $\lambda_{G1} = 0.012$, and $\lambda_{\text{offset}} = \lambda_{\text{spacing}} = 6 \times 10^{-3}$. The rendering resolution is 512×512 . For *TopoAdapt*, we use a unified proximity threshold of $\epsilon_{\text{merge}} = 4\text{m}$ for both node merging and T-junction creation. This value is physically grounded in the typical width of road lanes. All *TopoAdapt* hyperparameters (detailed in Suppl.) are kept fixed across datasets.

4.2 Evaluation Metrics

We follow standard practice [18, 20, 52] and evaluate with the TOPO [5] and APLS [13] metrics. These metrics primarily assess topology but include weak geometric constraints. For fair comparison with prior work, our Bézier graph is converted to a polyline compatible with the official evaluation scripts, using their public implementations and default parameters. We obtain these polylines by uniformly sampling points along each Bézier edge, ensuring a consistent discretization across all methods. The conversion is detailed in the supplementary.

4.3 Comparative Results

Table 1: Performance comparison with state-of-the-art methods on the SpaceNet and City-Scale datasets. The best results are shown in **bold**, and the second-best are underlined.

Method	SpaceNet				City-Scale			
	TOPO F1 ↑	Precision ↑	Recall ↑	APLS ↑	TOPO F1 ↑	Precision ↑	Recall ↑	APLS ↑
Sat2Graph	80.97	85.93	<u>76.55</u>	64.43	76.26	80.70	72.28	63.14
RNGDet	81.13	90.91	73.25	65.61	76.87	85.97	69.87	65.75
RNGDet++	<u>82.81</u>	91.34	75.24	67.73	78.44	85.65	72.58	67.76
SAMRoad	80.52	93.03	70.97	71.64	77.23	90.47	67.69	<u>68.37</u>
SAMRoad++	81.57	93.68	72.23	<u>73.44</u>	<u>80.01</u>	<u>88.39</u>	<u>73.39</u>	68.34
DOGE	84.58	<u>93.55</u>	78.43	73.48	80.59	84.42	77.40	70.24

We evaluate *DOGE* against state-of-the-art methods on SpaceNet and City-Scale. Qualitative results (Fig. 5) highlight that *DOGE* produces significantly smoother, more geometrically accurate road networks than prior methods (e.g., RNGDet++, SAMRoad++). This is a direct benefit of our Bézier curve representation, which naturally models the curvilinear nature of roads. As detailed in the supplementary material, kinematic smoothness metrics confirm that *DOGE* yields a physically compliant network. Furthermore, by adaptively partitioning road geometry, *DOGE* effectively models complex intersections without the jagged artifacts common in discrete approaches. This topological superiority at complex junctions is rigorously validated via the Split Detection Accuracy metric in the supplementary material.

The quantitative results in Tab. 1 confirm that *DOGE* establishes a new state-of-the-art across both benchmarks. It consistently leads the primary comprehensive metrics, achieving top scores on SpaceNet (TOPO F1: **84.58**, APLS: **73.48**) and City-Scale (TOPO F1: **80.59**, APLS: **70.24**). This dominance is driven by a substantial improvement in Recall (+3.19 on SpaceNet and +4.01 on City-Scale over the next best methods). Crucially, *DOGE* secures this highly complete topological reconstruction while preserving competitive Precision (e.g., 93.55 on SpaceNet). Crucially, *DOGE*’s gains are not merely a fine-tuned SAM2 segmentation upgrade. Under identical SAM1 masks on SpaceNet, *DOGE* achieves 83.20 TOPO F1 and 21.33 ED, compared with 80.08/47.12 for SAMRoad and 79.68/47.64 for SAMRoad++, respectively. As detailed in Sec. 4.5, applying Standard Post-Processing or unconstrained continuous optimization (*Vanilla DiffRender*) on the exact same masks yields significantly lower accuracy, confirming the necessity of our framework. On GlobalScale, *DOGE* improves over SAMRoad++ reported F1/APLS on both in-domain (64.39/65.15 vs. 62.33/62.19) and out-domain splits (51.47/44.80 vs. 48.34/43.17); full tables, off-the-shelf SAM2 results, and HD lane-mapping examples are in Suppl.

4.4 Analysis of Topological Compactness

Our method produces more accurate and significantly more compact road networks than prior work.

As Fig. 6 shows, we achieve higher topological accuracy (APLS) with substantially lower edge density, reducing edges by 73.2% against the GT polyline. Crucially, this compactness directly benefits downstream applications like autonomous routing, theoretically enabling a **nearly 4 $\times$ speedup** in classic graph traversal algorithms (e.g., Dijkstra or A*, detailed in Suppl.).

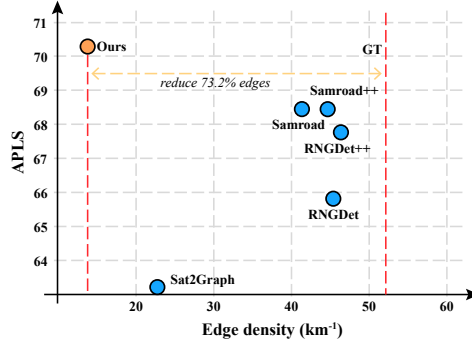


Fig. 6: Performance versus compactness on the City-Scale dataset. The plot shows APLS against edge density (edges/km).

4.5 Ablation Studies

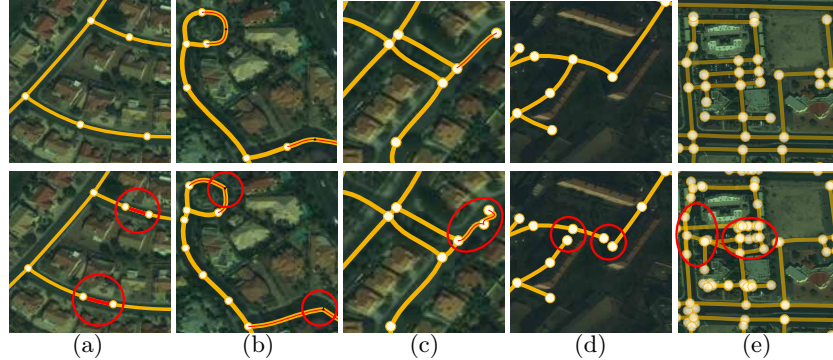


Fig. 7: Qualitative ablation study on key components and lower bounds. Top: Full model results. Bottom: Circled failure cases show (a) w/o $\mathcal{L}_{\text{overlap}}$, roads incorrectly intersect; (b) w/o $\mathcal{L}_{G1}$, junctions have sharp angles; and (c) w/o curve regularization ($\mathcal{L}_{\text{offset}}$, $\mathcal{L}_{\text{spacing}}$), curves degenerate. (d) *Standard Post-Processing* yields fractured topology from mask noise, while (e) *Vanilla DiffRender* produces chaotic graphs lacking structural constraints (rendered with transparency to reveal overlaps).

We conduct ablation studies on the SpaceNet dataset (Tab. 2). To verify that *DOGE*’s performance is driven by its discrete-continuous mechanism rather than merely the fine-tuned SAM2 backbone, we evaluate two lower bounds using exact the same segmentation. ***Standard Post-Processing*** applies traditional skeletonization and curve fitting directly to the SAM2 masks, without *DOGE*. It yields sub-optimal topology (APLS 63.10) because heuristic extraction is highly sensitive to mask jaggedness and minor disconnections, often resulting in fractured graphs (Fig. 7d). Conversely, ***Vanilla DiffRender*** optimizes the graph using only coverage loss ($\mathcal{L}_{\text{cover}}$). It performs even worse (APLS 37.10); while it minimizes L_2 error, the unconstrained continuous optimization results in a chaotic topology with severe overlaps (Fig. 7e). In contrast, *DOGE* successfully reconstructs high-fidelity road networks with precise geometry and topology.

Table 2: Ablation study of key components on SpaceNet. We systematically ablate our geometric priors and topological operators. ED denotes edge density. Furthermore, we include two lower-bound baselines: *Vanilla DiffRender* (naive gradient-based pixel matching) and *Standard Post-Processing* (traditional skeletonization on SAM2 masks without optimization) to highlight the necessity of *DOGE*.

Method	$\mathcal{L}_{\text{cover}}$	Geometric Priors				TopoAdapt	TOPO F1 $\uparrow$	APLS $\uparrow$	ED $\downarrow$
		$\mathcal{L}_{\text{overlap}}$	$\mathcal{L}_{G1}$	$\mathcal{L}_{\text{offset}}$	$\mathcal{L}_{\text{spacing}}$				
Full Model (<i>DOGE</i>)	✓	✓	✓	✓	✓	✓	84.58	73.48	22.62
w/o TopoAdapt	✓	✓	✓	✓	✓		82.80	69.45	66.60
w/o Overlap Loss	✓		✓	✓	✓	✓	83.38	65.95	40.30
w/o G1 Loss	✓	✓		✓	✓	✓	82.54	69.28	23.71
w/o Offset Loss	✓	✓	✓		✓	✓	83.24	72.28	33.14
w/o Spacing Loss	✓	✓	✓	✓		✓	83.66	72.07	37.25
w/o Curve Reg.	✓	✓	✓			✓	82.57	65.97	23.48
w/o Geom. Priors	✓					✓	74.58	51.18	44.17
Vanilla DiffRender	✓						70.94	37.10	98.96
Standard Post-Processing							79.57	63.10	69.21

Effectiveness of Analytic Geometric Priors. Analytic geometric priors are crucial for taming the continuous optimization. Removing them entirely ("w/o Geom. Priors") or individually causes severe performance drops. Fig. 7 visually confirms this, illustrating how ablating specific priors leads to artifacts like unnatural intersections (without $\mathcal{L}_{\text{overlap}}$) or sharp, degenerate curves (without $\mathcal{L}_{G1}$, $\mathcal{L}_{\text{offset}}$, and $\mathcal{L}_{\text{spacing}}$).

Effectiveness of TopoAdapt. *TopoAdapt* is vital for overcoming topological bottlenecks. Disabling it drops APLS by 4.03 and increases ED from 22.62 to 66.60, showing that discrete structural edits support both topology recovery and compactness. **Stability.** Across five random initializations on SpaceNet, *DOGE* remains stable, with TOPO F1 84.49 ± 0.08 and APLS 73.37 ± 0.23 .

5 Conclusion

We introduced *DOGE*, a new paradigm for road network extraction that reframes the task as a global optimization of a parametric Bézier Graph. By pioneering the use of differentiable rendering to align with segmentation masks, our method eliminates the need for vector GT. By decoupling geometric optimization (via *DiffAlign*) and topological refinement (via *TopoAdapt*), we reconstruct topologically accurate, geometrically smooth, and compact road networks, achieving state-of-the-art results on the SpaceNet and City-Scale benchmarks.

Limitations and Future Work. Despite strong performance, *DOGE* faces two visual-scene limitations (visualized in Suppl.). First, severe occlusions (e.g., dense canopies) can mislead *DiffAlign*, causing disconnections. Second, multi-level interchanges remain topologically ambiguous without height cues. In addition, *DOGE* is an offline per-image optimization method rather than a one-pass predictor; on SpaceNet, *DOGE*-512 takes 302 min for the test set on a single RTX 4090, with faster settings and comparisons reported in Suppl. Future work will explore integrating occlusion-robust priors and height cues, while improving efficiency to extend *DOGE* to broader GT-free vector reconstruction tasks.

Acknowledgements. This work was supported in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), the Natural Science Foundation of Jiangsu Province (No. BK20251195), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. ASAM e.V.: ASAM OpenDRIVE®: Static Road Network Description, Version 1.8.1. Association for Standardization of Automation and Measuring Systems (November 2024), <https://www.asam.net/standards/detail/opensdrive/>, standard Specification. Accessed 28 June 2026
2. Bahl, G., Bahri, M., Lafarge, F.: Single-Shot End-to-end Road Graph Extraction. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1402–1411. IEEE (Jun 2022). <https://doi.org/10.1109/CVPRW56347.2022.00146>
3. Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., DeWitt, D.: RoadTracer: Automatic Extraction of Road Networks from Aerial Images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4720–4728. IEEE (Jun 2018). <https://doi.org/10.1109/CVPR.2018.00496>
4. Bessmeltsev, M., Solomon, J.: Vectorization of line drawings via polyvector fields. *ACM Transactions on Graphics (TOG)* **38**(1), 1–12 (2019)
5. Biagioni, J., Eriksson, J.: Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation research record* **2291**(1), 61–71 (2012)
6. Blayney, H., Tian, H., Scott, H., Goldbeck, N., Stetson, C., Angeloudis, P.: Bézier Everywhere All at Once: Learning Drivable Lanes as Bézier Graphs. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15365–15374. IEEE (Jun 2024). <https://doi.org/10.1109/CVPR52733.2024.01455>
7. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J.: Argoverse: 3d tracking and forecasting with rich maps. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8740–8749 (2019). <https://doi.org/10.1109/CVPR.2019.00895>
8. Chen, Z., Deng, L., Luo, Y., Li, D., Junior, J.M., Gonçalves, W.N., Nurunnabi, A.A.M., Li, J., Wang, C., Li, D.: Road extraction in remote sensing data: A survey. *International journal of applied earth observation and geoinformation* **112**, 102833 (2022)
9. Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Liu, M.Y., Torralba, A., Fidler, S.: Neural turtle graphics for modeling city road layouts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
10. Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R.: DeepGlobe 2018: A challenge to parse the earth through satellite images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018). <https://doi.org/10.1109/CVPRW.2018.00031>

11. Dong, Z., Zhu, X., Cao, X., Ding, R., Li, W., Zhou, C., Wang, Y., Liu, Q.: BézierFormer: A unified architecture for 2d and 3d lane detection. 2024 IEEE International Conference on Multimedia and Expo (ICME) pp. 1–6 (2024)
12. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
13. Etten, A.V., Lindenbaum, D., Bacastow, T.M.: SpaceNet: A remote sensing dataset and challenge series (2018)
14. Farin, G.E.: Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann (2002)
15. Feng, Z., Guo, S., Tan, X., Xu, K., Wang, M., Ma, L.: Rethinking Efficient Lane Detection via Curve Modeling. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17041–17049. IEEE (Jun 2022). <https://doi.org/10.1109/CVPR52688.2022.01655>
16. Foley, J.D.: Computer graphics: principles and practice, vol. 12110. Addison-Wesley Professional (1996)
17. Forrest, A.R.: Interactive interpolation and approximation by bézier polynomials. The Computer Journal **15**(1), 71–79 (1972)
18. He, S., Bastani, F., Jagwani, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Elshrif, M.M., Madden, S., Sadeghi, M.A.: Sat2Graph: Road Graph Extraction Through Graph-Tensor Encoding. In: Computer Vision – ECCV 2020, vol. 12369, pp. 51–67. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-58586-0_4
19. He, Y., Garg, R., Chowdhury, A.R.: TD-Road: Top-Down Road Network Extraction with Holistic Graph Construction. In: Computer Vision – ECCV 2022, vol. 13669, pp. 562–577. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-20077-9_33
20. Hetang, C., Xue, H., Le, C., Yue, T., Wang, W., He, Y.: Segment Anything Model for Road Network Graph Extraction (Apr 2024). <https://doi.org/10.48550/arXiv.2403.16051>
21. Hu, Y., Wang, Z., Huang, Z., Liu, Y.: PolyRoad: Polyline Transformer for Topological Road-Boundary Detection. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–12 (2024). <https://doi.org/10.1109/TGRS.2023.3344103>
22. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment Anything. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3992–4003. IEEE (Oct 2023). <https://doi.org/10.1109/ICCV51070.2023.00371>
23. Li, Q., Wang, Y., Wang, Y., Zhao, H.: HDMapNet: An Online HD Map Construction and Evaluation Framework. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 4628–4634 (May 2022). <https://doi.org/10.1109/ICRA46639.2022.9812383>
24. Li, T.M., Lukáč, M., Gharbi, M., Ragan-Kelley, J.: Differentiable vector graphics rasterization for editing and learning. ACM Transactions on Graphics **39**(6), 1–15 (Dec 2020). <https://doi.org/10.1145/3414685.3417871>
25. Liao, B., Chen, S., Jiang, B., Cheng, T., Zhang, Q., Liu, W., Huang, C., Wang, X.: Lane Graph as Path: Continuity-Preserving Path-Wise Modeling for Online Lane Graph Construction. In: Computer Vision – ECCV 2024, vol. 15102, pp. 334–351. Springer Nature Switzerland (2025). https://doi.org/10.1007/978-3-031-72784-9_19
26. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: MapTR: Structured modeling and learning for online vectorized HD map construction. In: International Conference on Learning Representations (2023)

27. Liu, Y., Yuan, T., Wang, Y., Wang, Y., Zhao, H.: VectorMapNet: End-to-end vectorized HD map learning. In: International conference on machine learning. PMLR (2023)
28. Lu, X., Weng, Q.: Deep learning-based road extraction from remote sensing imagery: Progress, problems, and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing* **228**, 122–140 (2025)
29. Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., Fu, Y., Shi, H.: Towards layer-wise image vectorization. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2022)
30. Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P.: Can semantic labeling methods generalize to any city? the Inria aerial image labeling benchmark. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 3226–3229 (2017). <https://doi.org/10.1109/IGARSS.2017.8127684>
31. Mnih, V., Hinton, G.E.: Learning to detect roads in high-resolution aerial images. In: ECCV (2010)
32. Qiao, L., Ding, W., Qiu, X., Zhang, C.: End-to-End Vectorized HD-map Construction with Piecewise Bézier Curve. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13218–13228. IEEE (Jun 2023). <https://doi.org/10.1109/CVPR52729.2023.01270>
33. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: SAM 2: Segment Anything in Images and Videos (Oct 2024)
34. Reddy, P., Gharbi, M., Lukac, M., Mitra, N.J.: Im2vec: Synthesizing vector graphics without vector supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7342–7351 (2021)
35. Scheuer, A., Fraichard, T.: Continuous-curvature path planning for car-like vehicles. In: Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS’97. vol. 2, pp. 997–1003. IEEE (1997)
36. Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., Menze, B.: Relationformer: A Unified Framework for Image-to-Graph Generation. In: Computer Vision – ECCV 2022, vol. 13697, pp. 422–439. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-19836-6_24
37. Sun, P., Kretschmar, H., Dotiwala, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2443–2451 (2020). <https://doi.org/10.1109/CVPR42600.2020.00252>
38. Tan, Y.Q., Gao, S.H., Li, X.Y., Cheng, M.M., Ren, B.: VecRoad: Point-Based Iterative Graph Exploration for Road Graphs Extraction. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8907–8915. IEEE (Jun 2020). <https://doi.org/10.1109/CVPR42600.2020.00893>
39. Tang, Z., Naphade, M., Liu, M.Y., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D., Hwang, J.N.: CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8789–8798 (2019). <https://doi.org/10.1109/CVPR.2019.00900>

40. Vinker, Y., Alaluf, Y., Cohen-Or, D., Shamir, A.: CLIPascene: Scene sketching with different types and levels of abstraction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4146–4156 (October 2023)
41. Vinker, Y., Pajouheshgar, E., Bo, J.Y., Bachmann, R.C., Bermano, A.H., Cohen-Or, D., Zamir, A., Shamir, A.: CLIPasso: Semantically-aware object sketching. *ACM Trans. Graph.* **41**(4) (jul 2022). <https://doi.org/10.1145/3528223.3530068>
42. Wang, L., Dai, M., He, J., Huang, J.: Regularized Primitive Graph Learning for Unified Vector Mapping. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16771–16780. IEEE (Oct 2023). <https://doi.org/10.1109/ICCV51070.2023.01542>
43. Wang, S., Bai, M., Mattyus, G., Chu, H., Luo, W., Yang, B., Liang, J., Cheverie, J., Fidler, S., Urtasun, R.: TorontoCity: Seeing the world with a million eyes. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 3028–3036 (2017). <https://doi.org/10.1109/ICCV.2017.327>
44. Wang, Y., Lian, Z.: Deepvecfont: synthesizing high-quality vector fonts via dual-modality learning. *ACM Transactions on Graphics (TOG)* **40**(6), 1–15 (2021)
45. Wang, Y., Wang, Y., Yu, L., Zhu, Y., Lian, Z.: Deepvecfont-v2: Exploiting transformers to synthesize vector fonts with higher quality. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18320–18328 (2023)
46. Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., Zhu, F., Shao, L., Xia, G.S., Bai, X.: iSAID: A large-scale dataset for instance segmentation in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 28–37 (2019)
47. Xie, H., Chen, Z., Hong, F., Liu, Z.: Compositional generative model of unbounded 4D cities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025). <https://doi.org/10.1109/TPAMI.2025.3603078>
48. Xing, X., Wang, C., Zhou, H., Zhang, J., Yu, Q., Xu, D.: DiffSketcher: Text guided vector sketch synthesis through latent diffusion models. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), <https://openreview.net/forum?id=CY1xatvEQj>, accessed 28 June 2026
49. Xing, X., Zhou, H., Wang, C., Zhang, J., Xu, D., Yu, Q.: SVGDreamer: Text guided SVG generation with diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4546–4555 (June 2024)
50. Xu, J., Xu, B., Xia, G.S., Dong, L., Xue, N.: Patched line segment learning for vector road mapping. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(6), 6288–6296 (Mar 2024). <https://doi.org/10.1609/aaai.v38i6.28447>
51. Xu, Z., Liu, Y., Gan, L., Sun, Y., Wu, X., Liu, M., Wang, L.: RNGDet: Road Network Graph Detection by Transformer in Aerial Images. *IEEE Transactions on Geoscience and Remote Sensing* **60**, 1–12 (2022). <https://doi.org/10.1109/TGRS.2022.3186993>
52. Xu, Z., Liu, Y., Sun, Y., Liu, M., Wang, L.: RNGDet++: Road Network Graph Detection by Transformer With Instance Segmentation and Multi-Scale Features Enhancement. *IEEE Robotics and Automation Letters* **8**(5), 2991–2998 (May 2023). <https://doi.org/10.1109/LRA.2023.3264723>
53. Yin, P., Li, K., Cao, X., Yao, J., Liu, L., Bai, X., Zhou, F., Meng, D.: Towards satellite image road graph extraction: A global-scale dataset and a novel method. In:

- Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 1527–1537 (June 2025)
54. Zao, Y., Zou, Z., Shi, Z.: Road Graph Extraction via Transformer and Topological Representation. *IEEE Geoscience and Remote Sensing Letters* **21**, 1–5 (2024). <https://doi.org/10.1109/LGRS.2024.3380593>
 55. Zao, Y., Zou, Z., Shi, Z.: Topology-Guided Road Graph Extraction From Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* **62**, 1–14 (2024). <https://doi.org/10.1109/TGRS.2023.3344150>
 56. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* **27**(3), 236–239 (Mar 1984). <https://doi.org/10.1145/357994.358023>