Learning and Aggregating Lane Graphs for Urban Automated Driving

Martin Büchner[1*] Jannik Zürn[1*] Ion-George Todoran[2] Abhinav Valada[1] Wolfram Burgard[3]
[1]University of Freiburg [2]Woven by Toyota [3]University of Technology Nuremberg

Abstract

Lane graph estimation is an essential and highly challenging task in automated driving and HD map learning. Existing methods using either onboard or aerial imagery struggle with complex lane topologies, out-of-distribution scenarios, or significant occlusions in the image space. Moreover, merging overlapping lane graphs to obtain consistent large-scale graphs remains difficult. To overcome these challenges, we propose a novel bottom-up approach to lane graph estimation from aerial imagery that aggregates multiple overlapping graphs into a single consistent graph. Due to its modular design, our method allows us to address two complementary tasks: predicting ego-respective successor lane graphs from arbitrary vehicle positions using a graph neural network and aggregating these predictions into a consistent global lane graph. Extensive experiments on a large-scale lane graph dataset demonstrate that our approach yields highly accurate lane graphs, even in regions with severe occlusions. The presented approach to graph aggregation proves to eliminate inconsistent predictions while increasing the overall graph quality. We make our large-scale urban lane graph dataset and code publicly available at http://urbanlanegraph.cs.uni-freiburg.de.

1. Introduction

Most automated driving vehicles rely on the knowledge of their immediate surroundings to safely navigate urban environments. Onboard sensors including LiDARs and cameras provide perception inputs that are utilized in multiple tasks such as localization [7, 21, 27], tracking [4], or scene understanding [20, 24, 26, 37] to aggregate representations of the environment. However, robust planning and control typically require vastly more detailed and less noisy world models in the form of HD map data [12]. In particular, information on lane parametrization and connectivity is essential for both planning future driving maneuvers as well as high-level navigation tasks. Creating and maintaining HD maps in the form of lane graphs is a time-consuming and arduous task due to the large amount of detail required in the annotation and the data curation process including map updates based on local environment changes such as construction sites.

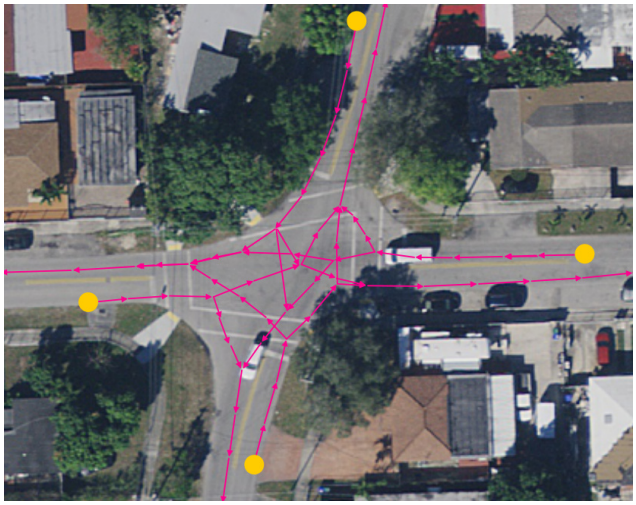

Figure 1. Our approach predicts accurate lane graphs from aerial images of complex urban environments. We visualize the estimated lane graph in magenta and indicate model initialization points with yellow circles.

Previous approaches to lane graph estimation have shown shortcomings in predicting lane graphs due to multiple deficiencies: On the one hand, methods using onboard imagery typically degrade at complex real-world intersections and under significant occlusions, e.g., when following another vehicle [5, 6]. On the other hand, methods based on aerial imagery show reduced performance when confronted with occlusions in the bird's-eye-view (BEV) due to, e.g., vegetation or shadows, and suffer from catastrophic drift when unconstrained in out-of-distribution scenarios [30]. Previous works treat intersections and non-intersections inherently differently [15] and thus require elaborated heuristics and post-processing to merge single predictions into a consistent lane graph. Moreover, prior works do not focus on use cases where multiple predicted graphs must be merged into a single consistent solution, which is essential for enabling the automatic generation of highly detailed lane graphs of large contiguous regions.

*Equal contribution

Related to the aforementioned challenges, we propose a novel two-stage graph neural network (GNN) approach termed LaneGNN that operates on single aerial color images for lane graph prediction. Inspired by methods in the field of trajectory prediction [8], we formulate a bottom-up approach according to which we place a virtual agent into a local crop of the aerial image and predict reachable successor lane graphs from its positions. To transform multiple disjoint local solutions into a single global solution, we aggregate a global representation by iteratively inferring the lane graph from consecutive poses, ultimately imitating real-world driving behavior. This iterative approach not only increases the predicted area covered but also improves graph accuracy based on data association and rejection. Note that we do not require any human in the loop to perform the graph aggregation. We visualize the output of our graph aggregation procedure in Fig. 1, in which we superimpose the predicted graph on the aerial image input. Using this framework, we envision two applications: ego-centered successor path prediction and full lane graph estimation by aggregation.

To summarize, the main contributions of this work are:

- An innovative bottom-up approach to lane graph estimation in challenging environments that explicitly encodes graph-level lane topology from input aerial images in a scenario-agnostic manner.
- A novel graph aggregation scheme enabling robust and method-agnostic merging of graph-level predictions.
- The large-scale lane graph dataset *UrbanLaneGraph* comprising high-resolution aerial images aligned with dense lane graph annotations aggregated from the Argoverse2 dataset that we make publicly available.
- Extensive experiments and ablation studies demonstrating the significance of our findings.

2. Related Works

In recent years, the prediction of topological road features such as road graphs and lane graphs have been extensively studied. In our discussion, we differentiate between road graph learning and lane graph learning. While road graphs encode the topological connections between road segments, lane graphs describe the locations and connectivity between all lanes, resulting in a spatially much denser graph. Many prior works focus on vehicle trajectory prediction, conditioned on HD map features such as lane centerline and boundary positions [9, 10, 28]. These models do not aim at exclusively predicting the road or lane graph structure from onboard vehicle images or aerial images but instead at predicting future vehicle states such as position and orientation.

Road Graph Learning: Prior works investigate estimating road graphs from both onboard sensors [18] and from aerial images [1, 22, 34] or focus on extracting pixel-level road segmentation from images and extracting graphical road representations, i.e., using morphological image operators or graph neural networks to extract the connectivity between different roads within the image [1, 19]. Other approaches investigate iterative methods and interpret road graph prediction as a sequential prediction task [2, 22].

Lane Graph Learning from Vehicle Data: Some earlier works in lane graph learning from onboard vehicle sensors such as cameras and LiDAR formulate lane extraction as an image-based lane centerline regression task [16]. Homayounfar *et al*. [17] aggregate onboard LiDAR data on highways and leverage a recurrent neural network to generate highway lane graphs in an iterative manner. Zhou *et al*. [35] utilize the OpenStreetMap database and a semantic particle filter to accumulate projected semantic predictions from a vehicle ego-view into a map representation. Zhang *et al*. [31] propose an online road map extraction system for a sensor setup onboard a moving vehicle and construct a graph representation of the road network using a fully convolutional neural network. More recently, Can *et al*. proposed two different methods [5, 6] for lane connectivity learning in intersection scenarios from onboard camera images.

Lane Graph Learning from Birds-Eye-View Data: Despite the advantages of leveraging readily available aerial data for training lane graphs, only a few works considered using aerial images as an input modality for the graph learning task. Zürn *et al*. [36] propose a lane centerline regression model jointly with a Graph R-CNN backbone to predict nodes and edges of the lane graph from a local aggregated bird's-eye-view image crop. More recently, He *et al*. [15] propose a two-stage graph estimation pipeline. They first extract lanes at non-intersection areas and subsequently predict the connectivity of each pair of lanes, and extract the valid turning lanes to complete the map at intersections.

In contrast to existing works, we do not estimate the complete lane graph visible in a given crop but only the part of the graph that is reachable from a virtual agent pose located within the crop, simplifying the graph estimation problem based on reduced topological complexity. Furthermore, we leverage a GNN to explicitly model the relationships between graph nodes, allowing us to leverage recent developments in the field of geometric deep learning. This explicit graph encoding and prediction allows us to formulate a model that does not internally differentiate between intersection areas and non-intersection areas, in contrast to some related works. Additionally, we propose a novel large-scale dataset for lane graph estimation from aerial images, allowing the research community to easily evaluate and compare their approaches.

3. Dataset

To evaluate our approach on challenging real-world data, we compiled the *UrbanLaneGraph* dataset. It is a first-of-its-

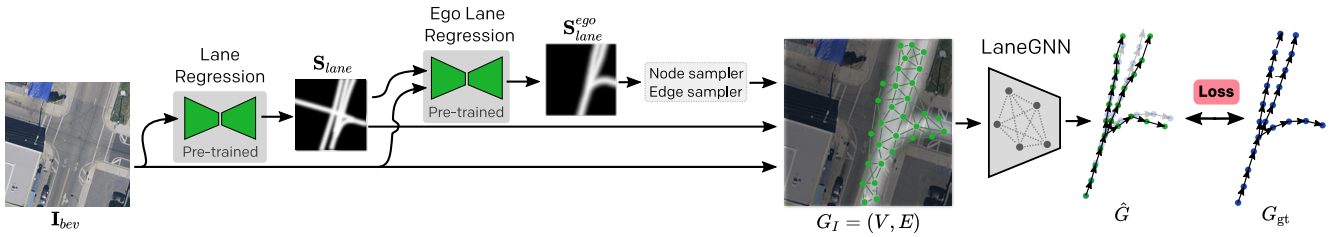

Figure 2. Overview of our LaneGNN model $\mathcal{M}$ predicting successor lane graphs. As a pre-training step, we train lane centerline and ego lane centerline regressor models. The ego lane regression $\mathbf{S}^{ego}_{lane}$ is used as a prior for sampling proposal nodes V in corridors that have a high likelihood of entailing the successor graph. The model learns a binary classification of node and edge scores while also predicting the probability of a node being an endpoint of a given lane segment.

Dataset	City	Lane Splits/Merges	Total Length [km]
UrbanLaneGraph *(from Argoverse2)*	Palo Alto	4752	796.4
	Austin	8495	531.7
	Miami	7642	850.8
	Pittsburgh	14610	1314.3
	Washington D.C.	2066	739.6
	Detroit	5424	990.5
LaneExtraction [15]	Boston, Seattle, Phoenix, Miami	2262	398.6
NuScenes [11]	Boston	1630	76.9

Table 1. Key statistics of our *UrbanLaneGraph* dataset, aggregated from the Argoverse2 dataset [29], used for our experiments, compared with other recent datasets for lane graph estimation.

kind dataset for large-scale lane graph estimation from aerial images. The dataset contains aerial images from the cities of Austin, Miami, Pittsburgh, Palo Alto, Detroit, and Washington DC. The images have a resolution of $15\,\text{cm}$ per pixel. To obtain the corresponding lane annotations, we leverage the Argoverse2 dataset [29] which entails lane graphs for large sections of the respective cities. The lane graphs in the Argoverse2 dataset are provided on a per-scenario basis, covering only small local areas in one graph. Therefore, we collect all local lane graphs of each city and aggregate them into one globally consistent graph per city, thereby removing inconsistent or redundant nodes or edges. The annotated regions feature a diverse range of environments including urban, suburban, and rural regions with complex lane topologies. Accumulated over all cities, the overall length of all lanes spans over $5.000\,\text{km}$. We split each city into disjoint training and testing regions. We list key dataset statistics in Tab. 1, indicating the scale of our generated dataset compared with other recently proposed datasets containing graph annotations. For more details on the dataset and exemplary visualizations, please refer to Sec. S.1 in the supplementary material.

4. Technical Approach

Our approach is divided into two stages: lane graph learning and lane graph aggregation. In the first stage (Sec. 4.1), we train our GNN model, denoted as LaneGNN, to predict the successor lane graph, entailing the nodes and edges that can be logically visited from the pose of a virtual vehicle agent. In the second phase (Sec. 4.2), we use our trained LaneGNN model to traverse a large map area. This is achieved by selecting an initial starting pose and predicting the successor lane graph from this pose. Subsequently, we iteratively estimate the traversable lane graph from the current pose and move forward along the predicted graph while aggregating. In the following, we detail both stages of our approach.

4.1. Lane Graph Learning

We formulate the task as a supervised learning problem where a successor lane graph $\hat{G}$ is estimated based on an aerial image $\mathbf{I}_{bev}$. First, a directed graph $G_I = (V, E)$ covering relevant regions is constructed by sampling from likely regions of $\mathbf{I}_{bev}$ (see Fig. 2). For all our models and experiments we choose a spatial resolution of 256×256 pixels. The graph comprises a set of nodes $i \in V$ that are connected via directed edges $E \subseteq \{(i,j)|(i,j) \in V^2 \text{ and } i \neq j\}$ that constitute potentially valid lane graph edges. The graph is attributed using both node features $\mathbf{X} \in \mathbb{R}^{\|V\| \times D}$ and edge features $\mathbf{X}_e \in \mathbb{R}^{\|E\| \times (D_{geo}+D_{bev})}$, where D_{geo} and D_{bev} denote the dimensionality of the involved edge features (see Sec. 4.1). The overall model estimates an output graph $\hat{G} = \mathcal{M}(G_I|\theta)$, where $\mathcal{M}$ is parameterized by network model weights $\theta := (\theta_{reg}, \theta_{GNN})$.

Lane Regression and Graph Construction: We train two regression networks: Firstly, a centerline regression network predicting the likelihood map of lane centerlines $\mathbf{S}_{lane}$, and secondly, a segmentation network predicting all reachable lanes $\mathbf{S}^{ego}_{lane}$ starting from the initial virtual agent pose at the bottom center of $\mathbf{I}_{bev}$. We use identical PSPNet [33] architectures with a ResNet-152 feature extractor for this task. We sample equally-distributed node positions using Halton sequences [14] that are later filtered based on the obtained ego-lane segmentation mask $\mathbf{S}^{ego}_{lane}$, which serves as a region of interest for sampling (see Fig. 2). Directed edges E among nodes are initialized for pairs of nodes with a Euclidean distance $d_{ij} \in [d_{min}, d_{max}]$. Initial node features $\mathbf{X}$ are crafted solely based on their 2D positions $\mathbf{x}_i = (x_i, y_i)$, while geometric edge features are defined as

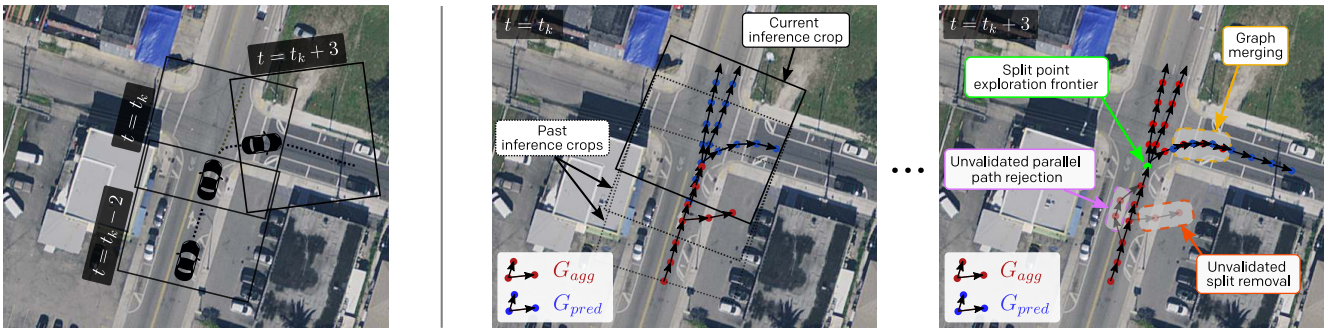

Figure 3. In our graph aggregation procedure, we iteratively obtain oriented image crops based on virtual agent poses along the currently predicted successor graph. For each crop, our LaneGNN model predicts a successor graph G_{pred}, which is aggregated into a globally consistent lane graph G_{agg}.

$$\mathbf{x}_{ij,geo} = \left(\tan^{-1} \frac{\Delta y_{ij}}{\Delta x_{ij}}, \sqrt{\Delta x_{ij}{}^2 + \Delta y_{ij}{}^2}, \overline{x}_{ij}, \overline{y}_{ij} \right), \quad (1)$$

where Δx_{ij} and Δy_{ij} represent node position differences and $\overline{x}_{ij}$, $\overline{y}_{ij}$ are edge middle point coordinates. In addition to the geometric edge feature, we generate aerial edge features: Per edge, a small oriented region of $\mathbf{I}_{bev}$ and the lane segmentation $\mathbf{S}_{lane}$ is obtained based on the direction of the edge, which provides $\mathbf{X}_{ij,bev} = [\mathbf{I}^*_{bev}, \mathbf{S}^*_{lane}]$ with dimension $D_{bev} = 4 \times 32 \times 32$.

Feature Encoding and Message Passing: In our approach, we estimate edge probabilities, node probabilities, and whether a node is terminal. While the nodes themselves hold only unidirectional information, we encode the notion of direction using the proposed edge feature as outlined above. We utilize a causal variant of neural message passing as proposed by Brasó *et al.* [3]. By imposing a causality prior, our network encodes predecessor and successor features during message passing. This formulation of message passing renders our approach direction-aware. Initial node and geometric edge features are encoded using multi-layer perceptrons $f^{\square}_{enc}$ (MLP) while the aerial edge feature is transformed using a ResNet-18 architecture $f^{e,bev}_{enc}$. The geometric and aerial edge features are concatenated and fused subsequently to arrive at various node and edge embeddings $\mathbf{H}^{(0)}_v$ and $\mathbf{H}^{(0)}_e$:

$$f^{e,bev}_{enc}(\mathbf{X}_{e,bev}) = \mathbf{H}^{(0)}_{e,bev}, \; f^{e,geo}_{enc}(\mathbf{X}_{e,geo}) = \mathbf{H}^{(0)}_{e,geo}, \quad (2)$$

$$f^{v}_{enc}(\mathbf{X}) = \mathbf{H}^{(0)}_{v}, \; f^{e}_{fuse}([\mathbf{H}^{(0)}_{e,geo}, \mathbf{H}^{(0)}_{e,bev}]) = \mathbf{H}^{(0)}_{e}. \quad (3)$$

In the following, multiple message-passing steps are performed using various ReLU-activated MLPs denoted by $f_{\square}$ as follows. The edge feature is updated based on the current neighboring node features $\mathbf{h}^{(l-1)}_i$, $\mathbf{h}^{(l-1)}_j$ and the edge feature $\mathbf{h}^{(l-1)}_{ij}$:

$$\mathbf{h}^{(l)}_{ij} = f_e \left(\left[\mathbf{h}^{(l-1)}_i, \mathbf{h}^{(l-1)}_j, \mathbf{h}^{(l-1)}_{ij} \right] \right). \quad (4)$$

Messages $\mathbf{m}^{(l)}_{ij}$ are crafted based on either predecessors $\mathcal{N}_{pred}(i)$ or successors $\mathcal{N}_{succ}(i)$ of a node i and propagated based on the constructed adjacency. In the next step, all predecessors and successor messages, respectively, are aggregated using a permutation-invariant sum:

$$\mathbf{h}^{(l)}_{i,pred} = \sum_{j \in \mathcal{N}_{pred}(i)} \underbrace{f^{pred}_v \left(\left[\mathbf{h}^{(l-1)}_j, \mathbf{h}^{(l)}_{ij}, \mathbf{h}^{(0)}_j \right] \right)}_{\mathbf{m}^{(l)}_{ji}}, \quad (5)$$

$$\mathbf{h}^{(l)}_{i,succ} = \sum_{j \in \mathcal{N}_{succ}(i)} \underbrace{f^{succ}_v \left(\left[\mathbf{h}^{(l-1)}_j, \mathbf{h}^{(l)}_{ij}, \mathbf{h}^{(0)}_j \right] \right)}_{\mathbf{m}^{(l)}_{ij}}. \quad (6)$$

Note that the message crafting includes skip connections to initial node embeddings $\mathbf{h}^{(0)}_j$. Nodes are updated by combining the two features using concatenation:

$$\mathbf{h}^{(l)}_i = f_v \left(\left[\mathbf{h}^{(l)}_{i,pred}, \mathbf{h}^{(l)}_{i,succ} \right] \right). \quad (7)$$

Finally, sigmoid-valued edge scores $\hat{e}_{ij}$ and node score $\hat{s}_i$ are predicted from the obtained edge and node embeddings. In a separate network head, we classify each node as being a terminal node or not, denoted as a scalar $\hat{t}_i$. Therefore, we optimize the following combined binary cross entropy:

$$\mathcal{L} = -\sum_{|V|} s_i \log \hat{s}_i - \sum_{|V|} t_i \log \hat{t}_i - \sum_{|E|} e_{ij} \log \hat{e}_{ij}. \quad (8)$$

The ground truth graph G_{GT} as a learning target (s_i, t_i, e_{ij}) is generated based on the map annotations for the given cropped region and the corresponding closest nodes. This is further outlined in the supplementary material in Sec. S.3.

4.2. Iterative Temporal Graph Aggregation

In the second stage of our approach, we aggregate local successor graphs into a globally consistent lane graph. First, we prune the predicted per-crop lane graph, and second, we iteratively aggregate the sparse graphs. In the following, we detail both components.

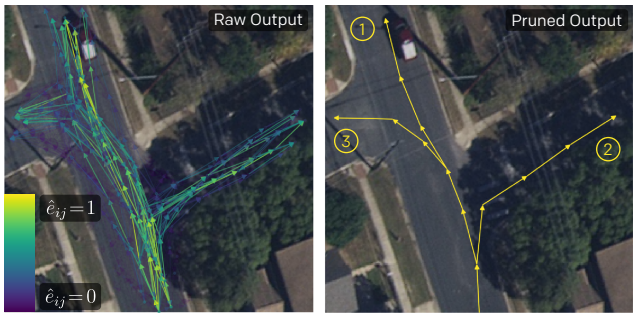

Figure 4. Comparison of raw and pruned lane graph predictions. On the left-hand side, we visualize estimated edge scores $\hat{e}_{ij}$ and node scores $\hat{s}_i$ using the same color scaling. On the right-hand side, we visualize the pruned graph. The circled numbers denote the order of traversal based on estimated terminal node scores.

Pruning: The graph prediction obtained from the LaneGNN model follows the graph connectivity initially generated during sampling. Since the model prediction contains a number of redundant paths with high predicted node and edge scores, we prune the obtained solution to obtain sparse lane representations. We formulate the graph pruning problem as a search problem from a starting node to possibly multiple predicted terminal nodes (see Sec. 4.1). Predicted lane split points should coincide with actual split points. Thus, different branches should share the same set of edges up to a split point. We use Dijkstra's algorithm to iteratively find high-score paths between the initial pose and terminal nodes, ordered from high to low scores until all terminal nodes are reached. Edge scores contained in found paths are set to zero. In Fig. 4, we visualize the output of this step.

Aggregation: We iteratively aggregate predicted successor lane graphs $G_{pred} = (V_{pred}, E_{pred})$ into a globally consistent and complete graph $G_{agg} = (V_{agg}, E_{agg})$ as depicted in Fig. 3. The predicted successor graph G^t_{pred} is added to the current aggregated graph G^{t-1}_{agg} at time step t: $G^t_{agg} \leftarrow \texttt{aggregate}(G^t_{pred}, G^{t-1}_{agg})$. Plausible branches of G_{pred} are merged into G_{agg} and thus extend the aggregated graph with every iteration if new grounds are covered in the respective crops. The next virtual agent pose is extracted from the set of forward-facing edges of G_{agg} given the current pose. Only edges with a significant weight due to previous aggregations are selected for this set. As the node positions differ slightly with each model forward pass we observe roughly similar paths in the lateral sense wrt. the ground-truth graph. However, along the longitudinal dimension of a branch, we observe deviations in node position, which must be circumvented when aggregating the graph. Based on this, we only take the lateral deviation wrt. G^t_{agg} into account when merging G^t_{pred}. Thus, G^t_{agg} is only updated in a lateral sense while the longitudinal misalignment of the two sets of nodes is neglected (see also Sec. S.5 in the supplementary material). The nodes of G^t_{pred} have a weight of 1 while a node of G^t_{agg} holds a weight equal to the number of merges it has observed so far. If a novel node $i \in V_{pred}$ is not close to any other node in $k \in V_{agg}$ it is added to the aggregated graph including its incident edges. This ultimately allows a weighting-based merging of arbitrary pairs of graphs, which is used for global lane graph estimation (see Sec. 5.4).

We observe that the more graphs we aggregate, the more we are certain about the significance of particular graph branches. Following a weighting-based approach allows us to set certain thresholds that allow modification of G_{agg}. Thus, implausible graph branches, semantically similar parallel paths, redundant edges, and isolated nodes are deleted based on confidence and distance as well as angle criteria (see Fig. 3). As a result, we are able to decrease the number of false positive split and merge points to obtain more consistent global graphs. We observe that this approach greatly improves lane graph prediction accuracy in difficult occluded and out-of-distribution scenarios since the sum of model predictions covering the same region shed light onto what potentially constitutes, e.g., a valid and an invalid branch.

Multiple forward passes naturally lead to a multitude of lane split points (both true positive and false positive splits). We interpret each split point as an element of an exploration frontier. A queue of unexplored, high-probability graph branches is maintained, which is queried in a depth-first manner as soon as the currently traversed branch terminates. Following the weighting-based approach, a branch is only explored if its depth-limited oriented successor tree weight exceeds a certain level of confidence. Due to this flexible approach, we can essentially handle arbitrary lane graph topologies with a single holistic approach. For more details on the graph pruning and aggregation procedures, please refer to the supplementary material. Finally, we apply multiple iterations of Laplacian smoothing to G^t_{pred}, which modifies the original node positions in order to even out position irregularities caused by sampling while keeping the adjacency represented by E^t_{pred} constant.

5. Experimental Results

In the following, we present our experimental findings. We first define and illustrate three tasks on which we benchmark our method. Subsequently, we describe the evaluation metrics and compare against other methods as well as our own baselines. We provide extensive qualitative and quantitative evaluations on our *UrbanLaneGraph* dataset.

5.1. Proposed Tasks

We propose three distinct and complementary tasks. In the first task, successor lane graph prediction *(Successor-LGP, Sec. 5.3)*, we aim at predicting a feasible ego-reachable successor lane graphs from the current pose of the virtual agent. The purpose of the Successor-LGP task is to measure

Table 2. Quantitative results of our model including ablation studies, in comparison with baseline models for the Successor-LGP task on our *UrbanLaneGraph* dataset. P/R denotes Precision/Recall. For our LaneGNN model variants, we denote CMP as causal message passing, aerial node features as AerN, aerial edge features as AerE, and the ego-lane regression with $\mathbf{S}_{lane}^{ego}$. For all metrics, higher values mean better results.

Model	CMP	AerN	AerE	$\mathbf{S}_{lane}^{ego}$	TOPO P/R ↑	GEO P/R ↑	APLS ↑	SDA_{20} ↑	SDA_{50} ↑	Graph IoU ↑
Skeletonized Regression	–	–	–	–	0.597/0.613	0.578/0.601	**0.315**	0.020	0.185	0.180
LaneGraphNet [36]	–	–	–	–	0.0/0.0	0.0/0.0	0.179	0.0	0.0	0.063
LaneGNN (ours)	✗	✗	✓	✓	0.549/0.677	0.548/0.671	0.188	0.168	0.323	0.312
	✓	✓	✗	✓	0.562/0.656	0.562/0.655	0.192	0.151	0.298	0.320
	✓	✗	✗	✓	0.545/0.693	0.545/0.688	0.200	0.188	0.311	0.336
	✓	✗	✓	✗	0.578/0.669	0.577/0.659	0.150	0.132	0.227	0.250
	✓	✗	✓	✓	**0.600/0.699**	**0.599/0.695**	0.202	**0.227**	**0.377**	**0.347**

the prediction quality of potential future driving paths when no HD map coverage is available. In the second task, full lane graph prediction (*Full-LGP*, Sec. 5.4), we evaluate the quality of regionally aggregated lane graphs in the context of HD map estimation. This task aims at measuring the predictive power of our full two-stage model performing lane graph inference and graph aggregation in conjunction. For such purposes, the full ground truth lane graph of a given map area is compared to the aggregated predictions of our model. Finally, we carry out a high-level path planning task (Sec. 5.5) on the predicted lane graphs, intended to analyze the fidelity of routes planned on the predicted graphs.

5.2. Evaluation Metrics

We leverage multiple complementary metrics for performance evaluation as detailed below.

Graph IoU: This metric measures the intersection-over-union (IoU) of two graphs rendered as a binary image [36], where pixels closer than $d = 5$ pixels are assigned the label 1 and the remaining pixels the label 0. Equivalent to the evaluation of semantic segmentation models, we determine the IoU values for the non-zero pixels.

The **APLS metric** sums the differences in optimal path lengths between nodes in the ground truth graph G and the proposal graph G' [25]. The APLS metric scales from 0 (worst) to 1 (best). Formally, it is defined as

$$\text{APLS} = 1 - \frac{1}{N_p} \sum \min \left\{ 1, \frac{|d(v_1, v_2) - d(v_1', v_2')|}{d(v_1, v_2)} \right\}, \quad (9)$$

where v_i and v_i' are nodes in G and G', respectively. N_p denotes the number of paths in G and $d(\cdot, \cdot)$ is the path length. For more details, please refer to [25].

TOPO / GEO metrics: Following previous works in road network extraction and lane graph estimation, we use the GEO metric and the TOPO metric. For definitions and details on these metrics, please refer to [15] and to the supplementary material.

Split Detection Accuracy ($\mathbf{SDA_R}$): This metric evaluates how accurately a model predicts the lane split within a circle of radius R pixels from a given ground truth lane split.

5.3. Successor Lane Graph Prediction

In the following, we evaluate LaneGNN by ablating and comparing it with two baselines on the Successor-LGP task: morphological skeletonization of the ego-lane regression as well as a modified LaneGraphNet [36] to be used for successor lane graph prediction. We list quantitative results in Tab. 2. Our results demonstrate that none of the baseline methods are capable of estimating accurate lane graphs given the challenging topology of the lane graphs in the dataset. The LaneGraphNet [36] model fails to model the graph for many samples, yielding low scores in all metrics. Despite its simplicity, the skeletonized regression model achieves the highest APLS score and comparably high GEO/TOPO scores. However, it fails to accurately predict lane split points, resulting in low SDA scores. Increasing the number of nodes of the skeleton leads to many more false positive splits and thus deteriorates further.

Regarding different variants of LaneGNN, we find that, e.g., causal message passing (CMP) increases performance over standard message passing, which underlines the significance of the imposed causality prior for lane graph learning. In order to show the efficacy of computationally more demanding aerial image edge features (AerE) compared to uni-directional aerial image node features (AerN), we ablate on this in Tab. 2 as well. We observe stark increases across the TOPO, GEO and SDA metrics when utilizing aerial edge features. Lastly, we replace the ego-lane segmentation mask $\mathbf{S}_{lane}^{ego}$ used for sampling with the standard lane segmentation mask. Our findings show that especially APLS, SDA, and Graph IoU drastically decrease as graph estimation becomes more difficult due to a generally enlarged sampling region (see Fig. 2).

These results are further illustrated in Fig. 5, where we show qualitative comparisons of predictions of our best-performing model with predictions from the two baselines. We find that the quality of predictions by LaneGraphNet [36] is generally low, rendering it unsuitable for the task. The skeletonized regression baseline is capable of following basic lane graph topologies, but lane split points cannot be resolved accurately. In contrast, our LaneGNN model is capable of modeling most graphs with high accuracy; both in intersection areas and in straight road sections. For more

Figure 5. Qualitative results on the Successor-LGP task. We compare predictions of our model with LaneGraphNet [36] and a morphological image skeletonization baseline. Predicted nodes are visualized with points while predicted edges are visualized as directed arrows. We illustrate failure cases in the two rightmost columns. Best viewed zoomed in.

Table 3. Quantitative evaluation for the Full-LGP task on the test-set of our *UrbanLaneGraph* dataset. We compare a baseline model with graphs aggregated with a naïve aggregation scheme and our iterative temporal aggregation scheme. P/R denotes Precision/Recall. Higher values mean better results.

Model	TOPO P/R ↑	GEO P/R ↑	APLS ↑	Graph IoU ↑
LaneExtraction [15]	0.405/0.507	0.491/0.454	0.072	0.213
Aggregation (naïve)	0.366/0.654	0.523/**0.727**	0.101	0.376
Aggregation (ours)	**0.481/0.670**	**0.649**/0.689	**0.103**	**0.384**

results, please refer to the supplementary material, Sec. S.6.

5.4. Full Lane Graph Prediction

For the Full-LGP task, we compare our approach with LaneExtraction [15]. Since their used graph representation is incompatible with ours, we train it on their provided dataset. To allow for a fair comparison, we evaluate both our method and LaneExtraction only on scenes in the city of Miami, Florida, as it is contained in both of the datasets. We select a testing region that is not part of the training data for either of the models. To initialize our aggregation scheme, we select starting poses obtained from intermediate segmentation predictions including yaw angles of the LaneExtraction model. Tab. 3 lists the evaluation results for the Full-LGP task obtained with LaneExtraction [15] and the results obtained with our LaneGNN model in conjunction with two aggregation schemes: a naïve aggregation scheme baseline and our full aggregation scheme. The naïve aggregation scheme merges nodes in close proximity while not relying on unvalidated split/merge or parallel path removal as well as the lateral weighting-based merging (Sec. 4.2). Our experiments show that our aggregation scheme outperforms both the LaneExtraction model and the naïve aggregation scheme on nearly all evaluation metrics. We note that our method improves the TOPO/GEO precision metrics while maintaining similar recalls due to better handling of redundant nodes. Fig. 6 illustrates successive aggregations from our model while indicating the used initialization points from the LaneExtraction model. Since our aggregation approach does not differentiate between intersection and non-intersection regions, it does not deteriorate in regions that do not exactly fit this categorization. Furthermore, our model exhibits superior performance in reduced visibility settings introduced by stark illumination changes or road occlusions from vegetation, as illustrated in Fig. 6. One of the decisive assets of our bottom-up method is that it allows to *explore* regions that are missed by LaneExtraction [15] as they are entailed in their predicted segmentation masks. For more qualitative and quantitative results, please refer to the supplementary material, Sec. S.7.

5.5. Path Planning

To illustrate the efficacy of our aggregation scheme, we evaluate the quality of the lane graph on a planning task. We

Figure 6. Qualitative results on the Full-LGP task. We visualize predictions of LaneExtraction [15] (top row) and aggregated LaneGNN predictions (bottom row). Our model is initialized at poses using predicted lane direction masks of LaneExtraction (indicated with yellow circles). Best viewed zoomed in.

Table 4. Quantitative evaluation for the planning task. MMD denotes mean minimum distance, MED denotes mean endpoint distance, and SR denotes the path planning success rate.

Model	MMD [m] ↓	MED [m] ↓	SR ↑
LaneExtraction [15]	157.0	339.4	0.47
Aggregation (ours)	2.2	19.7	0.46

generate 1000 randomly selected starting poses in the Miami graph test area from which a plan to a random goal within the graph must be found, using A* search. We place the points such that a maximal optimal route length of $200\,\mathrm{m}$ is not exceeded. To evaluate the planned routes, we compare the mean minimum distance (MMD) and the mean route endpoint distance (MED) between the paths on the predicted graph and the ground truth graph, respectively. We also report the success rate (SR), indicating the number of cases in which a path between start and goal exists. We list the results in Tab. 4. While the SR for our aggregation scheme and the LaneExtraction predictions is similar, the low MMD and MED values of our aggregation scheme indicate that our generated lane graph entails shorter and more direct paths, compared to LaneExtraction. We show additional results in the supplementary material, Sec. S.8.

5.6. Limitations

Due to its bottom-up architecture, the proposed approach performs well for most evaluated scenes in urban and suburban surroundings but struggles with highly complex graph topologies such as multi-lane intersections or roundabouts. Moreover, due to the iterative formulation of our aggregation scheme, the inference time of our approach increases with the number of nodes and edges. To speed up inference time, future work might include adaptively changing the distance between consecutive virtual agent positions and leveraging efficient neighborhood lookup methods such as k-d trees. Parallel execution of multiple agents would additionally boost run time to match top-down approaches and is feasible in terms of the proposed aggregation scheme.

6. Conclusion

In this work, we presented a novel lane graph estimation framework complemented with a novel dataset comprising aerial images. We showed that formulating the lane graph estimation problem as bottom-up graph neural network approach leveraging agent-centric views yields promising results. In addition, we presented a novel aggregation scheme to merge successive lane graphs to produce large-scale solutions. A first-of-its-kind dataset and benchmark for lane graph estimation from aerial images will enable further research in this field. Future work could address end-to-end training and exploiting further modalities such as onboard vehicle cameras for additional context information.

Acknowledgements: This work was partly funded by the German Research Foundation (DFG) Emmy Noether Program grant number 468878300, DFG grant number BU 865/10-2, and a hardware grant from NVIDIA.

References

[1] Wele Gedara Chaminda Bandara, Jeya Maria Jose Valanarasu, and Vishal M Patel. Spin road mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 343–350. IEEE, 2022. 2

[2] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Sam Madden, and David DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4720–4728, 2018. 2

[3] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[4] Martin Büchner and Abhinav Valada. 3d multi-object tracking using graph neural networks with cross-edge modality attention. *IEEE Robotics and Automation Letters*, 7(4):9707–9714, 2022. 1

[5] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15661–15670, 2021. 1, 2

[6] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Topology preserving local road network estimation from single onboard camera image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17263–17272, 2022. 1, 2

[7] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 38(4):2074–2093, 2022. 1

[8] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99. PMLR, 2020. 2

[9] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022. 2

[10] Nemanja Djuric, Henggang Cui, Zhaoen Su, Shangxuan Wu, Huahua Wang, Fang-Chieh Chou, Luisa San Martin, Song Feng, Rui Hu, Yang Xu, et al. Multixnet: Multiclass multistage multimodal motion prediction. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 435–442. IEEE, 2021. 2

[11] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 3

[12] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. *IEEE Robotics and Automation Letters*, 7(2):1968–1975, 2022. 1

[13] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008. 12

[14] J Halton and G Smith. Radical inverse quasi-random point sequence, algorithm 247. *Commun. ACM*, 7(12):701, 1964. 3

[15] Songtao He and Hari Balakrishnan. Lane-level street map extraction from aerial imagery. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2080–2089, 2022. 1, 2, 3, 6, 7, 8, 13, 21

[16] Namdar Homayounfar, Wei-Chiu Ma, Shrinidhi Kowshika Lakshmikanth, and Raquel Urtasun. Hierarchical recurrent attention networks for structured online maps. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3417–3426, 2018. 2

[17] Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. Dagmapper: Learning to map by discovering lane topology. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2911–2920, 2019. 2

[18] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Shenlong Wang, and Raquel Urtasun. Convolutional recurrent network for road boundary extraction. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9512–9521, 2019. 2

[19] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3438–3446, 2017. 2

[20] Rohit Mohan and Abhinav Valada. Amodal panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21023–21032, 2022. 1

[21] Kürsat Petek, Kshitij Sirohi, Daniel Büscher, and Wolfram Burgard. Robust monocular localization in sparse hd maps leveraging multi-task uncertainty estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4163–4169. IEEE, 2022. 1

[22] Yong-Qiang Tan, Shang-Hua Gao, Xuan-Yi Li, Ming-Ming Cheng, and Bo Ren. Vecroad: Point-based iterative graph exploration for road graphs extraction. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8910–8918, 2020. 2

[23] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 11

[24] Abhinav Valada, Ankit Dhall, and Wolfram Burgard. Convoluted mixture of deep experts for robust semantic segmentation. In *IEEE/RSJ International conference on intelligent robots and systems (IROS) workshop, state estimation and terrain perception for all terrain mobile robots*, volume 2, 2016. 1

[25] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018. 6

[26] Johan Vertens and Wolfram Burgard. Usegscene: Unsupervised learning of depth, optical flow and ego-motion with

semantic guidance and coupled networks. *arXiv preprint arXiv:2207.07469*, 2022. 1

[27] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. *arXiv preprint arXiv:2203.01578*, 2022. 1

[28] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17134–17142, 2022. 2

[29] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 3, 11

[30] Zhenhua Xu, Yuxuan Liu, Lu Gan, Yuxiang Sun, Xinyu Wu, Ming Liu, and Lujia Wang. Rngdet: Road network graph detection by transformer in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 2022. 1

[31] Li Zhang, Faezeh Tafazzoli, Gunther Krehl, Runsheng Xu, Timo Rehfeld, Manuel Schier, and Arunava Seal. Hierarchical road topology learning for urban map-less driving. *arXiv preprint arXiv:2104.00084*, 2021. 2

[32] Tongjie Y Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984. 15

[33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3, 16

[34] Lichen Zhou, Chuang Zhang, and Ming Wu. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 182–186, 2018. 2

[35] Yiyang Zhou, Yuichi Takeda, Masayoshi Tomizuka, and Wei Zhan. Automatic construction of lane-level hd maps for urban scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6649–6656. IEEE, 2021. 2

[36] Jannik Zürn, Johan Vertens, and Wolfram Burgard. Lane graph estimation for scene understanding in urban driving. *IEEE Robotics and Automation Letters*, 6(4):8615–8622, 2021. 2, 6, 7

[37] Jannik Zürn, Sebastian Weber, and Wolfram Burgard. Trackletmapper: Ground surface segmentation and mapping from traffic participant trajectories. In *6th Annual Conference on Robot Learning*, 2022. 1