2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

RCAL:A Lightweight Road Cognition and Automated Labeling System for Autonomous Driving Scenarios

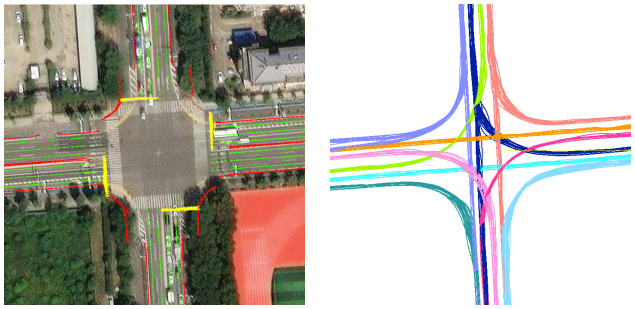
Jiancheng Chen*, Chao Yu*, Huayou Wang, Kun Liu, Yifei Zhan, Xianpeng Lang, Changliang Xue

***Abstract*— Vectorized reconstruction and topological cognition of road structures are crucial for autonomous vehicles to handle complex scenes. Traditional frameworks rely heavily on high-definition (HD) maps, which place significant demands on storage, computation, and manual labor. To overcome these limitations, we introduce a lightweight Road Cognition and Automated Labeling (RCAL) system. It leverages lightweight road data captured from mass-produced vehicles to vectorize road elements and cognize their topology. RCAL compiles multi-trip data on cloud servers for enhanced accuracy and coverage, addressing the limitations of single-trip data. In the field of element extraction, we proposed a pivotal point priority sampling strategy that can balance the contradiction between road scale and processing efficiency. Additionally, traffic flow is utilized to enhance the accuracy of road topology cognition. With its impressive automation, reliability, and efficiency, RCAL stands as an advanced solution in the field. Our evaluations on the intersection dataset from the real world confirm that RCAL not only achieves comparable precision to traditional HD map labeling systems but also substantially reducing resource costs.**

I. INTRODUCTION

In the domain of autonomous driving, recent advancements have stimulated an increasing demand for annotated data, particularly for the annotation of road elements and feature attributes within the bird's-eye view (BEV) space, which need aligns with the vehicle's surround-view images. To response to the evolution of autonomous driving capability, automating the feature labeling process and accurately recognizing the road structure are essential.

The traditional approach to generating cognition of road structures is mainly to produce high-definition (HD) maps, which rely on precise three-dimensional reconstructions of real scenes. This process typically requires skilled operators to manually identify and extract road features, including lane lines, road markings, and traffic signals. Additionally, virtual elements such as centerlines also require operators to draw them imaginatively base on existing instances. These methods[1][2][3] commonly process the raw point cloud of LIDAR, surround-view images, and vehicle location captured by survey vehicles, followed by the intricate alignment of multi-trip raw point clouds to develop optimized pose graph. However, this process demands extensive communication, computing and storage resources, resulting in significant overhead costs. Recently, learning methods for online road cognition have been emerging, such as HDMapNet[4], MapTR[5], and TopoNet[6]. These approaches leverage BEV perception to comprehend road structure information and employ an encoder-decoder paradigm to directly regress lane line instances for sequential points. This paradigm has enhanced the efficiency of cognitive road structure and labeling elements. However, existing methods operate on single-trip trajectory and are unable to facilitate cognition interaction and validation among different vehicles, which can easily lead to cognitive defect.



(a) The global road structure after multi-trip aggregation. (b) The traffic flow captured by mass-produced vehicles.

Fig. 1: **lightweight aggregated data and traffic flow.**

In response to the aforementioned issues, we propose a lightweight road cognition and automated labeling (RCAL) system, which serves as a universal platform solution aimed at efficient labeling. RCAL abandons the reliance on raw sensor data. As illustrated in Fig.1, it utilizes lightweight data which include a few point clouds of road structure and traffic flow captured from mass-produced vehicles. These point clouds are generated by the onboard BEV perception model and multi-frame aggregation is performed aligned with the vehicle's instantaneous positioning, which can effectively solve the perception accuracy of the camera in motion. Additionally, RCAL processes multi-trip data at the cloud server to construct global intersection instances across scales of hundreds of meters. This overcomes the cognitive defect of onboard models, such as the inability to fully observe extensive intersections within the views of a single trajectory, or scenarios where vehicular congestion obscures critical road markers. Based on the intersection instances, RCAL identifies and extracts road elements (dividers, centerlines, road boundaries) from lightweight data, then reconstructs the full geometric structure of the intersection. Moreover, by utilizing the vast amount of trajectories, RCAL effectively recognizes the centerline connection relationships of intersections, enabling comprehensive cognition of the intersection instances and topological structures. Finally, RCAL aligns cognitive data and vehicle's location to produce

All authors are with Li Auto Inc., Beijing, China. {chenjiancheng, yuchao1, wanghuayou, liukun, zhanyifei, xuechangliang}@lixiang.com.
* equal contribution.

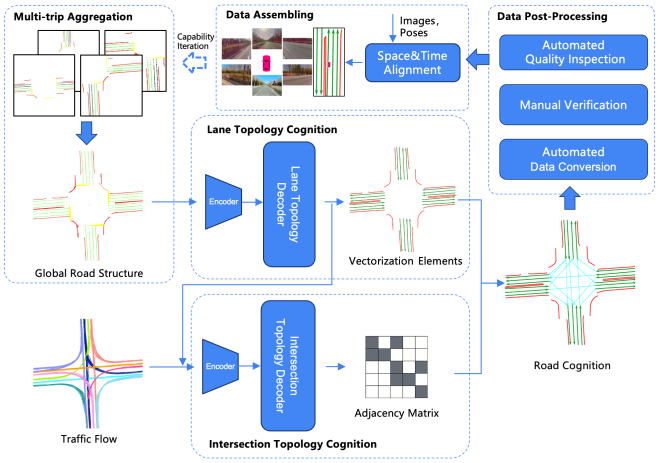


Fig. 2: **An overview of RCAL.** RCAL employs lightweight data and traffic flow to recognize the driving scene. It uses point sequences and adjacency matrices to represent road topology. The output results are compiled, corrected, and quality checked before being uploaded to the cloud. These results, combined with location data, are used to generate new annotated road elements in BEV.

annotated road elements in BEV, and then synchronizes with surround-view images. These measures significantly reduce the resource costs of manual labeling.

Particularly, due to the absence of ground truth road topology in public datasets, such as NuScenes and Argoverse2, and coupled with inadequate traffic flow coverage for individual roads, we have opted to use our proprietary dataset from the real world for performance evaluation.

Our contributions are summarized as follows:

- We propose a road cognition and automated labeling system based on a large volume of lightweight data, designed to enhance labeling efficiency and reduce the resource costs.
- A novel sampling method has been designed to enhance the precision of lane element recognition. This method prioritizes the sampling of pivotal points of road elements, which can reduce the information loss caused by sampling and improve the recognition accuracy.
- An end-to-end topology inference model utilizes element vector and traffic flow to directly predict the centerline adjacency matrix, which completes the cognition of intersection road structures.

The remainder of this paper is structured as follows. Section II provides a review of the related work. The framework and modular design of RCAL are elaborated in Section III. Experimental comparisons and validation are presented in Section IV. The paper concludes with a discussion and potential directions for future expansion in Section V.

II. RELATED WORK

A. Auto-Labeling System

Automated labeling systems have become a key factor in the rapid development of autonomous driving[7]. The initial methods designed to accelerate the labeling process utilize the reconstruction by LiDAR data, and fuse segmentation results from images to establish an automated annotation pipeline[8][9]. In order to reduce manual labor, learning models have been deployed. A notable development in the field is THMA[3], which has created an HD map toolkit that integrates a suite of AI models. This facilitates the extraction of road elements from point cloud maps. To address the challenges posed by the massive size of point cloud maps during automated labeling, the VMA system[2] proposes a 'divide-and-conquer' approach, which significantly improves annotation efficiency. While most systems rely on integrating raw LiDAR data for road cognition, RCAL is unique in utilizing only lightweight data. The goal is to streamline the process, particularly in recognizing roads element in BEV space and aligned with vehicle's surround-view images.

B. Road Elements Generate

Initial learning methods of automatically extract key road elements only produce semantic feature images in BEV space. It firstly focused on improving the feature transformation from images to BEV space, for instance, utilizing Multilayer Perceptrons (VPN[10]) or based on depth estimation (LSS[11] and BEVDet[12]) to address perspective discrepancies. And base on the generated semantic map, then groups the per-pixel segmentation results in post-processing(HDMapNet[4]).

Facing the issue of lengthy post-processing time, recent work has started to explore end-to-end vectorized map learning approaches, aiming to directly learn and generate more compact forms of vectorized road elements. These methods still confront challenges in modeling the topological relationships between and within the geometric structures of road elements. The coarse-to-fine architecture and autoregressive network of VectorMapNet[13] increase model complexity and may lead to longer inference times and cumulative errors. DETR-like[14][15][16][17] methods employ BEV queries and introduce geometric priors through attention mechanisms using Transformers architecture. Notably, the MapTR series[5][18] models road elements with a fixed number of points, but is prone to information loss when dealing with large curvatures and right-angled boundaries. PivotNet[19] learns the representation of vectorized maps in an end-to-end manner, using a dynamic number of key points to model road elements, which can reduce information loss due to equal spaced sampling. However, it requires a specialized loss function and multiple hyperparameters for dynamic pivotal points, and these measures also increasing the difficulty of model training. In order to better adapt to the automatic labeling of large-scale roads, we propose a pivotal point-prioritized sampling scheme for road element identification.

C. Traffic Flow Information

Traffic flow, with its low capture costs and inherent geographic information, is widely used in road cognition to identify drivable areas and infer road topology[20][21][22]. In order to generate drivable areas of the road, weakly-supervised[23] and self-supervised[24] learning methods are

used to aggregate vehicle's trajectories and images to obtain the range of lanes in the road. In paper [25], it leveraged the subtle geographic information contained within trajectories to detect the core areas of intersections and calibrate their topology.

In terms of road topology cognition, Bayesian Graphical Models have been employed to process trajectories and infer the topology relationships of centerlines within intersection regions, as discussed by Joshi et al.[26]. Zhou et al.[27] used semantic segmentation filtering and a directed cyclic graph, constructed from map skeletons, to obtain the lane topology of intersections. However, these methods require an HD map as a prior. Zurn et al.[28] demonstrated the use of trajectories and BEV perspective images to predict the heat map of the lane graph, and subsequently skeletonized a Successors graph without the need for other priors. In [29], traffic flow was used to identify preliminary entry and exit points at intersections by intersecting trajectories with polygon outlines. The mean shift clustering method was then applied to further refine the positioning of entry and exit points, and thus generate a crowdsourced map. Our work focuses on the end-to-end fusion of traffic flow with the vector of road elements to facilitate the cognition of intersection topology.

III. METHODOLOGY

This section details the four procedural strategies implemented in the RCAL system framework shown in Fig.2. Firstly, the multi-trip aggregation for a global road structure using single-trip lightweight data is described. Subsections II and III discuss the cognition of lane and intersection topology within driving scenarios, respectively. Finally, we outline the post-processing of cognition data and the method for automatically producing annotated data.

A. Multi-Trip Aggregation

The single-trip semantic scene reconstruction is mainly composed of semantic element tracking, smooth denoising, and splicing. This process is independently and parallelly conducted inside each trip. For the perceived line-type semantic features, we use the state-of-the-art Catmull-Rom Spline-based lane reconstruct method proposed by Qiao[30]. This method utilizes a bipartite graph to model the lane association process as an assignment problem and achieves edge weight assignment by incorporating chamfer distance, attitude uncertainty, and lateral sequence consistency. Additionally, this algorithm carefully designs control point initialization, spline parameterization, and optimization to create, extend, and refine splines step by step. Through these careful designs, this step can denoise the consecutive frame perception results and obtain smooth single-trip splicing results.

Since the sensing data of each trip only contains a local area around the vehicle's trajectory, the aggregate of multi-trip perception results are crucial to constructing the complete road. We first register the single-frame sensing data between trips, and then build a factor graph optimization algorithm to fuse the inter-trip single-frame registration,

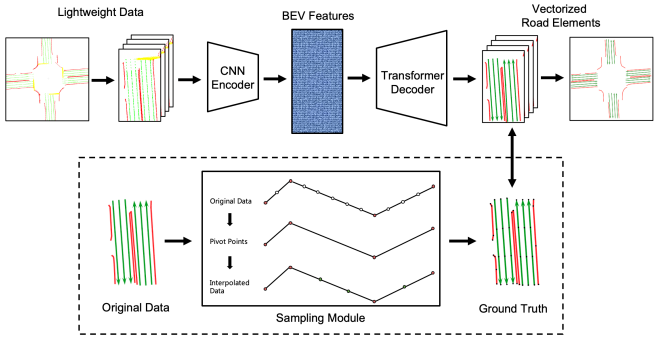


Fig. 3: **Overall architecture of LTC.** The upper part shows LTC adopts an encoder-decoder architecture. CNN encoder extracts BEV features from lightweight data. The Transformer decoder restores the vectorized road structure through the attention mechanism. The lower part embodies our proposed ground truth sampling scheme. Ground truth is obtained from the original data through pivotal point extraction and interpolation processes. The resampled ground truth will match the model predict results and supervise the model training.

INS, and odometry observations. Finally, the global semantic scene reconstructed by aggregating multi-trip data as shown in Fig .1(a).

B. Lane Topology Cognition

In order to support intersection topology cognition and automated annotation, we built a Lane Topology Cognition (LTC) model that adopts pivotal point priority sampling method. The model identifies and extracts road vector elements including dividing lines, center lines and boundaries through lightweight lane structure data. The processing pipeline first obtains the aggregated data and key trajectories, and then splits the data into fixed-size images based on the trajectories as model input. After model inference, the obtained vectorized topology of road elements will also be recombined into a complete aggregate road structure according to the same trajectory. As shown in Fig. 3, this section will discuss the model architecture, key steps, and specific optimization strategies compared with traditional methods.

1) Model Architecture: The model consists of two primary components: a Convolutional Neural Network (CNN) Encoder and a Transformer Decoder. The CNN Encoder processes the input images to extract image features. It employs semantic segmentation for auxiliary supervision, converting the BEV image of lightweight input into a BEV feature layer. On the other hand, the Transformer Decoder fuses these features with instance queries to extract vectorized road structure data, resulting in precise vectorized outputs.

2) Road Elements Decoding: The decoder uses a set-based method to predict road elements and assigns a series of learnable embedding sequences to each element. Each embedding determines the initial coordinates (x, y) on the scene feature map through a multilayer perceptron (MLP). The coordinates of the points are then iteratively refined through Transformer attention. In each iteration, the model performs feature sampling at decoded coordinates (x, y)

and predicts displacements (Δx, Δy). These offsets are then used to update the coordinates (x, y). Repeating the cycle of feature sampling and coordinate adjustment, the model will output point sequence representations of map elements and their respective semantic types and attributes with higher accuracy.

3) Optimization Strategy: During testing of the model on real data, we noticed that the road static elements of most scenes are highly structured and can be effectively depicted using a small number of points. By reducing the number of points for predicting road elements, we found that the training convergence of the model was accelerated. However, as the number of sampling points decreases, pivotal points of the road structure may be missed when using the traditional equally spaced sampling method, resulting in a large difference between the predicted shape of road elements and the actual shape. In order to overcome this problem, we optimized the sampling strategy while maintaining a fixed number of sampling points as shown in Fig. 3. First, pivotal points that are critical to the shape of road elements are identified, and these pivotal points are prioritized to be included in the sampling point set. Subsequently, the set of sample points is interpolated based on the distribution of these pivotal points to maintain consistency in the final number of sample points.

Given a set of road elements, the optimized sampling strategy consists of the following key steps:

Step 1: Pivotal Points Identification:

$$P_{\text{pivotal}} = \{p \mid \mathcal{A}(p) \geq \text{threshold}_{\text{area}} \vee \theta(p) \geq \text{threshold}_{\text{angle}}\} \tag{1}$$

where $\mathcal{A}$ is the area of the triangle formed by point p and the two adjacent points, θ is the angle between the two adjacent sides of point p, $\text{threshold}_{\text{area}}$ and $\text{threshold}_{\text{angle}}$ are preset thresholds for the area and angle respectively.

Step 2: Interpolate Points for Each Segment:

1) Calculate total length of the polyline and average segment length.

$$S_{\text{total}} = \sum_{i=1}^{n-1} \|p_{i+1} - p_i\|, \quad L_{\text{avg}} = \frac{S_{\text{total}}}{N_{\text{target}} - 1} \tag{2}$$

2) Calculate number of points to insert per segment.

$$m_i = \left\lceil \left(\frac{\|p_{i+1} - p_i\|}{L_{\text{avg}}} - 1 \right)_+ \right\rceil, \quad i \in [1, n-1] \tag{3}$$

3) Interpolate new points in the i-th segment.

$$p_t = (1-t) \cdot p_i + t \cdot p_{i+1}, \quad t \in \left\{ \frac{k}{m_i + 1} \right\}_{k=1}^{m_i} \tag{4}$$

where p is the set of pivotal points with n points, S_{total} is the total length of the polyline, L_{avg} is the average length of the segments that results in N_{target} total points, and m_i is the number of points to insert between the i-th and $(i+1)$-th point. The set $\{k\}_{k=1}^{n_i}$ represents the indices of the newly inserted points in each segment.

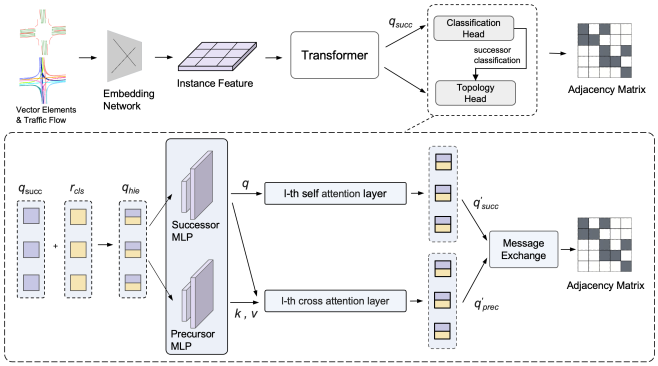


Fig. 4: **An overview of our peoposed ITC.**The upper layer explains that the ITC model uses an encoder-decoder architecture to obtain the centerline features, and uses two heads to complete the classification and regression tasks. The lower layer explains how to share knowledge for classification tasks and improve the inference effect of the adjacency matrix.

Step 3. Combine pivotal and interpolated points.

$$P_{\text{target}} = P_{\text{pivotal}} + P_{\text{Interpolate}} \tag{5}$$

C. Intersection Topology Cognition

Recognizing a complex intersection requires spatial information such as the positions and sequences of nearby centerlines. However, relying solely on these features often leads to inadequate interpretations of the surrounding topological network. The limited information makes it challenging to conclusively determine whether a specific lane is intended exclusively for straight-through traffic or also accommodates turning movement. To address these ambiguities, our intersection topology cognition (ITC) module directly uses the set of centerline vectors N_{lc} and traffic flow N_{tf} to predict the topological relationships of the centerlines within a certain range of the intersection. The overall structure is shown in the Fig.4. Since both the centerline instances and traffic flow information are ordered sequences of points in space, we utilize the same embedding network to extract their semantic information.

The head of ITC mainly involves two parts: the topology classification of the centerlines and the regression of the adjacency matrix. We process the query for each centerline through a simple MLP, which allows us to easily obtain a binary classification of whether a centerline has successors. For the inferring adjacency matrix, we introduced a hierarchical query q_{hie} that combines the one-hot results of the classifier r_{cls} with the successor query q_{succ}.

$$q_{hie} = q_{succ} + r_{cls} \tag{6}$$

Hierarchical query q_{hie} is subsequently updated through a decoding process that employs an MLP alongside a multi-layer attention. It is particularly noteworthy that we continuously enhance the successor information q'_{succ} through self-attention layers, while cross-attention layers are utilized to

capture precursor information q'_{prec} of the centerlines.

$$q'_{succ} = \varphi_{self}(MLP(q_{hie})) \tag{7}$$

$$q'_{prec} = \varphi_{cross}(MLP_{succ}(q_{hie}), MLP_{prec}(q_{hie})) \tag{8}$$

where, $\varphi_{self}(\cdot)$ is the multi-layer self-attention module, and $\varphi_{cross}(\cdot)$ is the multi-layer cross-attention module.

Finally, The predicted adjacency matrix $\hat{A}$ is derived through exchanging messages between the updated queries q'_{succ} and q'_{prec}, it is also a similarity assessment. Specifically, we employ the matrix dot product to facilitate this message-passing:

$$\hat{A} = q'_{succ} q'_{prec} \tag{9}$$

In the training for the ITC model, we adopt a composite loss function that comprises two distinct parts: a successor classification loss ($\mathcal{L}_{cls}$) and an adjacency matrix loss ($\mathcal{L}_{adj}$). The $\mathcal{L}_{cls}$ targets a classification task wherein centerline $i \in N_{lc}$ is assessed for the presence of successors, matching the predicted labels $\hat{c}_i$ with the corresponding ground truth c_i. Meanwhile, the $\mathcal{L}_{adj}$ is concerned with the adjacency matrix, paying special attention to whether the successors link by centerlines are correct. We apply a sigmoid function to process the predicted adjacency matrix, thereby obtaining $\widetilde{A}$ and enabling an accurate loss computation in relation to the definitive boolean ground truth adjacency matrix A. Importantly, for both types of loss, the computation leverages the Focal loss function.

$$\mathcal{L}_{cls} = \sum_{i=0}^{N_{lc}-1} \mathcal{L}_{Focal}(\hat{c}_i, c_(i)) \tag{10}$$

$$\mathcal{L}_{adj} = \sum_{i=0}^{N_{lc}-1} \sum_{j=0}^{N_{lc}-1} \mathcal{L}_{Focal}(\widetilde{A}_{ij}, A_{ij}) \tag{11}$$

$$\mathcal{L} = \alpha \mathcal{L}_{cls} + \beta \mathcal{L}_{adj} \tag{12}$$

D. Post Processes and Auto Labeling

The road cognition data automatically generated by RCAL undergoes a series of post-processing steps, including data conversion, manual verification, and automated quality inspection. Manual verification serves as the sole introduction of human effort, ensuring the quality of the cognition results. Compared to traditional methods, RCAL shifts human labor from the production stage to the verification stage. After undergoing automated quality inspection, the cognition data is converted into the vehicle's coordinate system based on the vehicle's location and divided into sizes appropriate for BEV space. Additionally, the timestamps are synchronized with the surround-view images to assemble the data frame required for the vehicle's onboard perception model. Notably, the utilization of annotated data by the vehicle's onboard perception model creates a closed loop with the RCAL system. Over time, these iterative enhancements in capability are expected to significantly reduce the reliance on human resources.

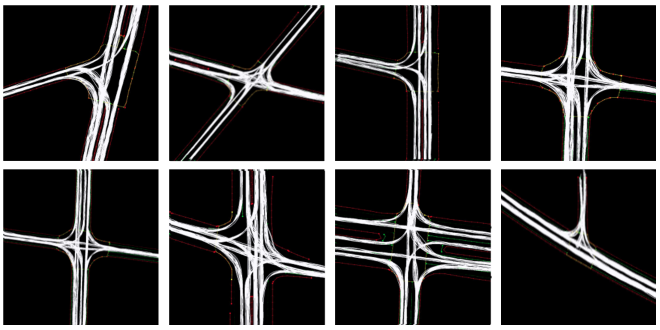


Fig. 5: **Intersection example of the intersection dataset.** It shows a variety of sample scenarios, intersection structures and traffic flow coverage in the dataset.

IV. Experiment

A. Dataset

Existing publicly available datasets, such as NuScenes and Argoverse2, lack detailed road topology data and sufficient trajectory coverage for comprehensive evaluation. Therefore, we constructed an intersection dataset from real-world scenarios for performance evaluation. This dataset encompasses common urban scenes and considers the size and structural complexity of intersections. As depicted in the Fig.5, we have segmented these intersections into 200*200 meter sections, creating thousands of detailed intersection datasets. These datasets include a mass of lightweight data collected from mass-produced vehicles, which include surround-view images, lightweight point clouds of road structures, and vehicle trajectory information. Manually annotated HD map provide the ground truth for the geometric positions and topologies of these elements. The variety of intersection designs and lane counts featured in the dataset ensures unbiased and objective quantitative evaluations.

B. Experimental Setting

We developed the automated labeling model based on our dataset, leveraging eight NVIDIA A100 GPUs with a batch size of 8 per GPU. In LTC model, we deployed the AdamW optimizer with a learning rate of 2e-4 and weight decay of 1e-4, and combined with a cosine annealing learning rate scheduler. For the feature querying process, the number of query points was established at $\{20, 10, 6\}$ for comparative test validation. The results are shown in the TABLE I. In order to meet the requirements of scene data structure and extreme performance testing, 6 query points are used for model comparison testing in TABLE II. Meanwhile, the same strategy is used on the ITC model, but with a learning rate of 1e-4.

To quantitatively measure the model's performance, this study adopts a comprehensive evaluation methodology to measure the performance of the proposed LTC and ITC models, along with the efficiency of the entire automated labeling platform. For assessing the LTC model, Average Precision (AP) is used to evaluate the quality of instances. Additionally, to delve into the precision of the model, the Chamfer Distance is employed for comparing the geometric

accuracy between the predicted instances and the ground truth. The evaluation of the ITC model focuses on its ability to accurately predict classification and adjacency matrices, employing precision and recall as the metrics for this purpose. Moreover, the effectiveness of the automatic annotation platform is determined by evaluating its consumption of computational resources, storage, and the extent of human effort required. These varied metrics together establish a comprehensive framework for verifying the effectiveness of the methods proposed in this study.

C. Quantitative Evalution

1) Lane Topology Cognition: As discussed in the Optimization Strategy section, we observe that reducing sampling points can enhance the results of the evaluation metrics when operating under the traditional equally spaced sampling method. TABLE I describes the evaluation results of the metrics mAP (mean average precision) and mAD (mean average distance).

For the distance metric, we adopted the Bidirectional Chamfer Distance D_{Chamfer}. Given two sets of points representing the predicted lines $P = \{p_1, p_2, \ldots, p_n\}$ and the ground truth lines $G = \{g_1, g_2, \ldots, g_n\}$, the bidirectional chamfer distance can be defined as the average of the unidirectional distances:

$$D_{\text{Chamfer}}(P,G) = \frac{1}{2}\left(\frac{1}{n}\sum_{p\in P}\min_{g\in G} D(p,g) + \frac{1}{n}\sum_{g\in G}\min_{p\in P} D(g,p)\right) \tag{13}$$

where $\mathrm{D}(p,g)$ denotes the Euclidean distance between points p and g.

TABLE I: Evaluation results of different sample points.

Sample Points	mAP↑	mAD↓
20	0.589	0.518
10	0.745	0.418
6	**0.799**	**0.404**

We focus on constructing three key road elements: lane dividers, lane centerlines, and road boundaries. To accurately define the model's prediction range, we set the range to cover an area of 30 meters to the front and rear of the vehicle and 15 meters to the left and right. The main metric to evaluate model performance is the average accuracy (AP) based on D_{Chamfer}. A prediction is identified as a true positive (TP) when the D_{Chamfer} between the model prediction and the actual road element is less than the specified thresholds. Fig. 6 shows the prediction results of our scheme compared with MapTRv2.

In order to better demonstrate the improvement in distance accuracy brought by our model, we further calculated the proportion of prediction results within a distance of {0.1,0.2,0.3,0.5,1.0} meters under the 1.5 meter threshold. The statistical results in TABLE II show that compared with MapTRv2, LTC has achieved an increase of more than 3% in the prediction ratio in most cases, and can reach a maximum improvement of nearly 10% (road boundary evaluation 0.2 meter threshold ratio).

TABLE II: Distance threshold ratio.

Map Element	Method	Threshold Ratio(%)				
		0.1m	0.2m	0.3m	0.5m	1.0m
Divider	MapTRv2	35.54	58.63	68.51	79.31	92.12
	LTC(Ours)	39.67(+4.13)	64.88(+6.25)	74.72(+6.21)	84.01(+4.70)	94.05(+1.93)
Boundary	MapTRv2	21.33	43.72	59.27	74.35	92.81
	LTC(Ours)	24.41(+3.08)	**53.37**(+9.65)	67.00(+7.73)	78.87(+4.52)	94.56(+1.75)
Centerline	MapTRv2	35.97	59.33	69.56	79.97	93.23
	LTC(Ours)	39.27(+3.30)	65.23(+5.90)	73.65(+4.09)	82.55(+2.58)	93.74(+0.51)

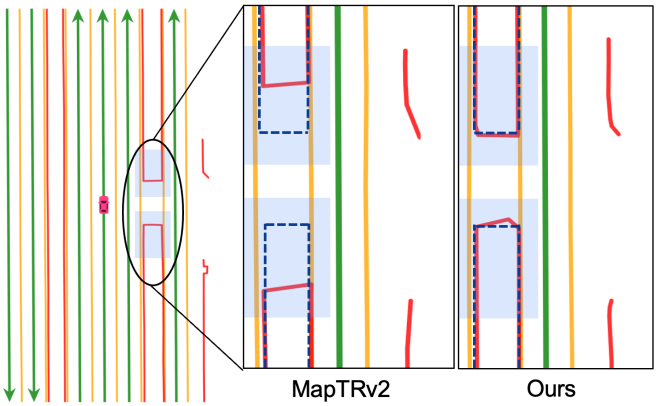


Fig. 6: **Road elements prediction of MapTRv2 and LTC.** Solid line: yellow signifies the divider, red denotes the boundary, and green indicates the centerline. Dashed black line represent the ground truth (GT) of the comparison element. From the comparison, our method can better restore the real results at conners.

2) Topology Cognition: As discussed before, recall and precision metrics were used to evaluate the performance of ITC model. For the classification task, recall and precision were defined as follows:

$$Precision_{cls} = \frac{\left|Cls_{true\_positive}\right|}{\left|Cls_{positive}\right|} \tag{14}$$

$$Recall_{cls} = \frac{\left|Cls_{true\_positive}\right|}{\left|Cls_{gt}\right|} \tag{15}$$

where $Cls_{positive}$ and Cls_{gt} denote the result of successor classifier for positive predicts and ground truth, respectively. $Cls_{true\_positive}$ signifies the true positive made by the classifier. Different metrics are employed for evaluating the predictions of the adjacency matrix. Here, the value within the adjacency matrix indicates the confidence coefficient of a successor link extending from the source centerline to the target. A link is considered valid if it possesses a confidence coefficient greater than 0.5. The metrics of the adjacency matrix are delineated as follows:

$$Precision_{adj} = \frac{\sum_{v\in V}\left|N_{\hat{A},true\_positive}\right|}{\sum_{v\in V}\left|N_{\hat{A},positive}\right|} \tag{16}$$

$$Recall_{adj} = \frac{\sum_{v\in V}\left|N_{\hat{A},true\_positive}\right|}{\sum_{v\in V}\left|N_{A,gt}\right|} \tag{17}$$

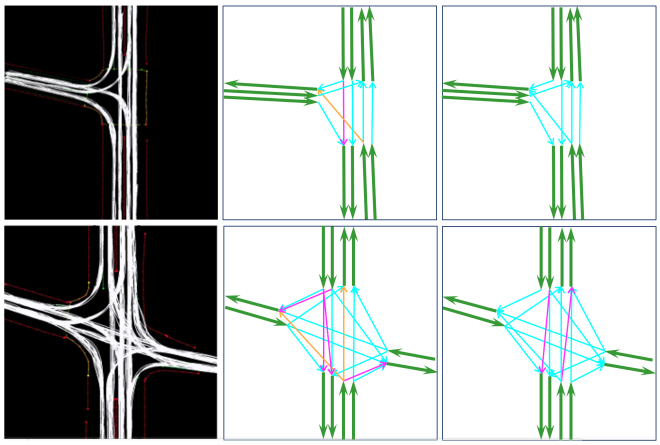


Fig. 7: **ITC Experiment Comparison**. Left: BEV of the selected intersection; Middle: Topology recognition results without traffic flow; Right: Topology recognition results with traffic flow considered. Blue arrows represent true positives of successor connections, orange arrows represent connections that were false negatives, and magenta arrows represent false positive identified connections.

where v meant the source centerline in centerlines groups V of recognized scene. $N_{\hat{A},positive}$ and $N_{A,gt}$ represent the number of successor link in the predict adjacency matrix $\hat{A}$ and ground truth matrix $N_{A,gt}$. Meanwhile, $N_{\hat{A},true\_positive}$ accounts for the number of correct positive in $\hat{A}$.

In order to investigate the impact of specific components on the model's performance, the ablation study examines several iterations of the original ITC method, which include the following: ITC (baseline) is the original model. ITC-single mlp head refers to a version of the ITC model comprising only a single MLP and self-attention mechanism within the topology head. ITC-no traffic flow is a variation that does not utilize traffic flow information. Furthermore, a version of ITC named ITC-no cls fusion omits the fusion with classification results for topology reasoning.

TABLE III: Ablation study results for the ITC methods.

Method	P_{cls}	R_{cls}	P_{adj}	R_{adj}
ITC	0.843	0.937	**0.818**	0.768
ITC-single mlp head	0.833	0.934	0.073	0.784
ITC-no traffic flow	0.827	0.941	0.596	0.677
ITC-no cls fusion	**0.845**	**0.949**	0.743	**0.793**

As ablation study results presented in TABLE III, the baseline ITC method, equipped with full features, has demonstrated commendable strength across various performance metrics. We observed that the precursor MLP and cross attention in the topology head of the model are pivotal for accuracy. Simplifying this element to a sole successor MLP (ITC-single MLP head) results in a drastic reduction in adjacency precision (P_{adj}) to 0.073. Furthermore, the exclusion of traffic flow information (ITC-no traffic flow) led to a notable decrease in P_{adj} to 0.596, affirming the significant role of traffic data in enhancing the model's inference capabilities, some experiment results are shown as Fig.7. In particular, while decoupling the fusion of classification outcomes with topology predictions (ITC-no cls fusion) resulted in a lower P_{adj} of 0.743, there was a marginal increase in classification recall (R_{cls}) to 0.949, It implies that the fusion of classification results brings the accuracy of topological regression at the expense of reducing the performance of a small part of the classifier. Collectively, these findings highlight the importance of each component within the ITC framework for improving the balance between precision and recall in classifying centerlines and inferring about the adjacency matrix.

D. Source Cost

To objectively evaluate the cost benefits achieved by the RCAL system, we referenced the traditional laser mapping process used to produce HD map for generating annotated data. We conducted a quantitative analysis, taking into account factors such as data storage requirements, processing time, and labor costs. It is noteworthy that the time spent on multi-trip aggregation has a correlation with both the number of kilometers per task and the extent of trajectory coverage. For clarity and ease of comparison, the unit 'T-km' has been created to represent the density of 100 trajectory coverages per kilometer for a task. The detailed results are presented in TABLE IV.

TABLE IV: Comparison of Traditional Method and RCAL

Cost	Metric	Traditional method	RCAL
Storage	Surrounding Images (Gb/min)	2.4	2.4
	Lider Point Clouds (Gb/min)	2.1	0.004
	Intermediate Data (Gb/T-km)	>200	2.7
Time	Road Reconstruction (min/T-km)	588.1	28.5
	Road Cognition (min/km)	-	<1
Labor	Map Production (km/person-day)	≈ 6	-
	Data Verification (km/person-day)	≈ 10	≈ 8

It can be observed from the table above that RCAL significantly reduces the memory consumption of both raw and intermediate data by discarding the raw lidar data. Moreover, RCAL's road cognition inference is much faster than the process of manually drawing HD map. Although the verification stage of RCAL currently necessitates manual fine-tuning of element vector and attribute, making it somewhat slower than traditional verification. It is expected that as the amount of data grows and the algorithm undergoes further iterations, the need to modify elements will decrease and the efficiency disparity will progressively diminish.

V. CONCLUSIONS

The RCAL system provides a new paradigm for autonomous driving data annotation. It greatly improves processing efficiency and coverage by using lightweight road data and a cloud server strategy to aggregate multi-trip data. The fusion of the pivotal point priority sampling strategy and traffic flow data further optimizes the extraction of road elements and the recognition of topological structures. Evaluation results on our intersection dataset show that RCAL

is able to achieve accuracy comparable to traditional high-definition map annotation systems while reducing resource costs. However, this system also has certain limitations, including reduced performance when the data is sparse or of low quality, and the problem of invalid data such as filtering occlusions not being considered in the automated process. The future work should focus on enhancing the system's robustness to varying qualities data and increasing automated process coverage. With the accumulation of data and optimization of algorithms, the RCAL system, as an important base for autonomous driving data closed loop, is expected to provide important technical support for the future development of data-driven end-to-end architecture.

REFERENCES

[1] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A robust pose graph approach for city scale lidar mapping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1175–1182.

[2] S. Chen, Y. Zhang, B. Liao, J. Xie, T. Cheng, W. Sui, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene," *arXiv preprint arXiv:2304.09807*, 2023.

[3] K. Tang, X. Cao, Z. Cao, T. Zhou, E. Li, A. Liu, S. Zou, C. Liu, S. Mei, E. Sizikova, *et al.*, "Thma: Tencent hd map ai system for creating hd map annotations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 13, 2023, pp. 15 585–15 593.

[4] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4628–4634.

[5] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized hd map construction," *arXiv preprint arXiv:2208.14437*, 2022.

[6] T. Li, L. Chen, X. Geng, H. Wang, Y. Li, Z. Liu, S. Jiang, Y. Wang, H. Xu, C. Xu, *et al.*, "Topology reasoning for driving scenes," *arXiv preprint arXiv:2304.05277*, 2023.

[7] S. Zhang, O. Jafari, and P. Nagarkar, "A survey on machine learning techniques for auto labeling of video, audio, and text data," *arXiv preprint arXiv:2109.03784*, 2021.

[8] E. E. Aksoy, S. Baci, and S. Cavdar, "Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving," in *2020 IEEE intelligent vehicles symposium (IV)*. IEEE, 2020, pp. 926–932.

[9] M. Elhousni, Y. Lyu, Z. Zhang, and X. Huang, "Automatic building and labeling of hd maps with deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 08, 2020, pp. 13 255–13 260.

[10] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4867–4873, 2020.

[11] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[12] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[13] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Vectormapnet: End-to-end vectorized hd map learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 352–22 369.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[15] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[16] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[17] Y. Lin, Y. Yuan, Z. Zhang, C. Li, N. Zheng, and H. Hu, "Detr does not need multi-scale or locality design," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6545–6554.

[18] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Maptrv2: An end-to-end framework for online vectorized hd map construction," *arXiv preprint arXiv:2308.05736*, 2023.

[19] W. Ding, L. Qiao, X. Qiu, and C. Zhang, "Pivotnet: Vectorized pivot learning for end-to-end hd map construction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3672–3682.

[20] J. Zürn, S. Weber, and W. Burgard, "Trackletmapper: Ground surface segmentation and mapping from traffic participant trajectories," in *6th Annual Conference on Robot Learning*, 2022.

[21] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 222–17 231.

[22] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[23] D. Barnes, W. Maddern, and I. Posner, "Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 203–210.

[24] J. Zürn, W. Burgard, and A. Valada, "Self-supervised visual terrain classification from unsupervised acoustic feature learning," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 466–481, 2020.

[25] L. Zhao, J. Mao, M. Pu, G. Liu, C. Jin, W. Qian, A. Zhou, X. Wen, R. Hu, and H. Chai, "Automatic calibration of road intersection topology using trajectories," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 1633–1644.

[26] A. Joshi and M. R. James, "Joint probabilistic modeling and inference of intersection structure," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1072–1078.

[27] Y. Zhou, Y. Takeda, M. Tomizuka, and W. Zhan, "Automatic construction of lane-level hd maps for urban scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6649–6656.

[28] J. Zürn, I. Posner, and W. Burgard, "Autograph: Predicting lane graphs from traffic observations," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 73–80, 2023.

[29] T. Qin, H. Huang, Z. Wang, T. Chen, and W. Ding, "Traffic flow-based crowdsourced mapping in complex urban scenario," *IEEE Robotics and Automation Letters*, 2023.

[30] Z. Qiao, Z. Yu, H. Yin, and S. Shen, "Online monocular lane mapping using catmull-rom spline," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7179–7186.