PersFormer: 3D Lane Detection via Perspective Transformer and the OpenLane Benchmark

Li Chen[1*†], Chonghao Sima[1*], Yang Li[1*], Zehan Zheng[1], Jiajie Xu[1], Xiangwei Geng[1], Hongyang Li[1,2†], Conghui He[1], Jianping Shi[3], Yu Qiao[1], and Junchi Yan[1,2]

[1] Shanghai AI Laboratory [2] Shanghai Jiao Tong University
[3] SenseTime Research
{lichen,simachonghao,liyang,lihongyang}@pjlab.org.cn
yanjunchi@sjtu.edu.cn

Abstract. Methods for 3D lane detection have been recently proposed to address the issue of inaccurate lane layouts in many autonomous driving scenarios (uphill/downhill, bump, *etc.*). Previous work struggled in complex cases due to their simple designs of the spatial transformation between front view and bird's eye view (BEV) and the lack of a realistic dataset. Towards these issues, we present PersFormer: an end-to-end monocular 3D lane detector with a novel Transformer-based spatial feature transformation module. Our model generates BEV features by attending to related front-view local regions with camera parameters as a reference. PersFormer adopts a unified 2D/3D anchor design and an auxiliary task to detect 2D/3D lanes simultaneously, enhancing the feature consistency and sharing the benefits of multi-task learning. Moreover, we release one of the first large-scale real-world 3D lane datasets: OpenLane, with high-quality annotation and scenario diversity. OpenLane contains 200,000 frames, over 880,000 instance-level lanes, 14 lane categories, along with scene tags and the closed-in-path object annotations to encourage the development of lane detection and more industrial-related autonomous driving methods. We show that PersFormer significantly outperforms competitive baselines in the 3D lane detection task on our new OpenLane dataset as well as Apollo 3D Lane Synthetic dataset, and is also on par with state-of-the-art algorithms in the 2D task on OpenLane. The project page is available at https://github.com/OpenPerceptionX/PersFormer_3DLane and OpenLane dataset is provided at https://github.com/OpenPerceptionX/OpenLane.

1 Introduction

Autonomous driving is one of the most successful applications for AI algorithms to deploy in recent years. Modern Advanced Driver Assistance Systems (ADAS) for either L2 or L4 routes provide functionalities such as Automated Lane Centering (ALC) and Lane Departure Warning (LDW), where the essential need

[*] Equal contribution. [†] Correspondence author.

for perception is a lane detector to generate robust and generalizable lane lines [12]. With the prosperity of deep learning, lane detection algorithms in the 2D image space has achieved impressive results [46,28,39], where the task is formulated as a 2D segmentation problem given front view (perspective) image as input [25,37,34,1]. However, such a framework to perform lane detection in the perspective view is not applicable for industry-level products where complicated scenarios dominate.

On one side, downstream modules as in planning and control *often* require the lane location to be in the form of the orthographic bird's eye view (BEV) instead of a front view representation. Representation in BEV is for better task alignment with interactive agents (vehicle, road marker, traffic light, *etc.*) in the environment and multi-modal compatibility with other sensors such as LiDAR and Radar. The conventional approaches to address such a demand are either to simply project perspective lanes to ones in the BEV space [52,32], or more elegantly to cast perspective features to BEV by aid of camera in/extrinsic matrices [15,19,60]. The latter solution is inspired by the spatial transformer network (STN) [22] to generate a one-to-one correspondence from the image to BEV feature grids. By doing so, the quality of features in BEV depends solely on the quality of the *corresponding* feature in the front view. The predictions using these outcome features are not adorable as the blemish of scale variance in the front view, which inherits from the camera's pinhole model, remains.

On the other side, the height[1] of lane lines has to be considered when we project perspective lanes into BEV space. As illustrated in Fig. 1, the lanes would diverge/converge in case of uphill/downhill if the height is ignored, leading to improper action decisions as in the planning and control module. Previous literature [52,34,44] inevitably hypothesize that lanes in the BEV space lie on a flat ground, *i.e.*, the height of lanes is zero. The planar assumption does not hold true in most autonomous driving scenarios, *e.g.*, uphill/downhill, bump, crush turn, *etc.* Since the height information is unavailable on public benchmarks or complicated to acquire accurate ground truth, 3D lane detection is ill-posed. There are some attempts to address this issue by creating 3D synthetic benchmarks [15,19]. Their performance still needs improvement in complex, realistic scenarios nonetheless (c.f. (b-c) in Fig. 1). Moreover, the domain adaption between simulation and real data is not well-studied [16].

To address these bottlenecks aforementioned, we propose Perspective Transformer, shortened as **PersFormer**, which has a spatial feature transformation module to generate better BEV representations for the task. The proposed framework unifies 2D/3D lane detection tasks, and substantiates performance on the proposed large-scale realistic 3D lane dataset, **OpenLane**.

First, we model the spatial feature transformation as a learning procedure that has an attention mechanism to capture the interaction both among local region in the front view feature and between two views (front view to BEV),

[1] We define the height of lane line z to be the relative height concerning the zero point in the ego vehicle coordinate system (x, y, z) in BEV 3D space. The coordinate of the perspective (front view) 2D space in the image plane is referred to as (u, v).

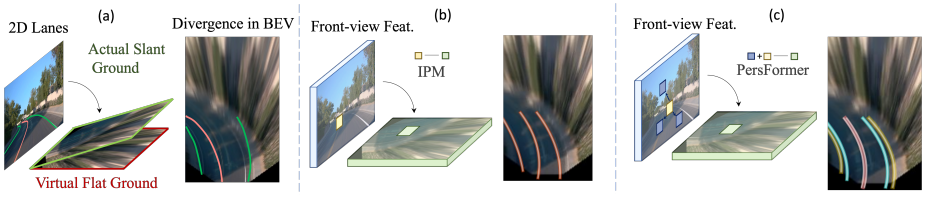

Fig. 1. Motivation of performing lane detection from 2D in (a) to BEV in (b); and the superiority of our method in (c) versus (b). Lanes would diverge/converge in projected BEV on planar assumption, and a 3D solution with height to be considered can accurately predict the parallel topology in this case

consequently being able to generate a fine-grained BEV feature representation. Inspired by [51,8], we construct a Transformer-based module to realize this, while the deformable attention mechanism [62] is adopted to remarkably reduce the computational memory requirement and dynamically adjust keys through the cross-attention module to capture prominent feature among the local region. Compared with direct 1-1 transformation via Inverse Perspective Mapping (IPM), the resultant features would be more representative and robust as it attends to the surrounding local context and aggregates relevant information. We further aim at unifying 2D and 3D lane detection tasks to benefit from the co-learning optimization. Second, we release the first real-world, large-scale 3D lane dataset and corresponding benchmark, OpenLane, to support research into the problem. OpenLane contains 200,000 annotated frames and over 880,000 lanes - each with one of 14 category labels (single white dash, double yellow solid, left/right curbside, *etc.*), which exceeds all of the existing lane datasets. It also has some distinguishing elements such as scenes, weather, and closed-in-path-object (CIPO) for other research topics in autonomous driving.

The main contributions of our work are three-fold: 1) Perspective Transformer, a novel Transformer-based architecture to realize spatial transformation of features; **2)** An architecture to simultaneously unify 2D and 3D lane detection, which is feasibly needed in the application. Experiments show that our PersFormer outperforms state-of-the-art 3D lane detection algorithms; **3)** The OpenLane dataset, the first large-scale realistic 3D lane dataset with high-quality labeling and vast diversity. The dataset, baselines, as well as the whole suite of codebase, is released to facilitate the research in this area.

2 Related Work

Vision Transformers in Bird's-Eye-View (BEV). Projecting features to BEV and performing downstream tasks in it has become more dominant and ensured better performance recently [29]. Compared with conventional CNN structure, the cross attention scheme in Vision Transformers [51,13,8,31,62] is naturally introduced to serve as a learnable transformation of features across

different views in an elegant spirit [29]. Instead of simply projecting features via IPM, the successful application of Transformers in view transformation has demonstrated great success in various domains, including 3D object detection [58,53,18,26], prediction [14,17,36], planning [38,11], *etc.*

Previous work [15,57,53,41,7] bring the BEV philosophy into pipeline, and yet they do not consider attention mechanism and/or 3D vision geometry (in this case, camera parameters). For instance, 3D-LaneNet [15] is set up with camera in/extrinsic matrices; the IPM process generates a virtual BEV representation from front view features. DETR3D [53] also considers camera geometry and formulates a learnable 3D-to-2D query search with attention scheme. However, there is no explicit BEV modelling for robust feature representation; the aggregated features might not be properly represented in 3D space. To address these shortcomings, our proposed PersFormer takes into account both the effect of camera parameters to generate BEV features and the convenience of cross-attention mechanism to model view transformation, achieving better feature representation in the end.

Lane Detection Benchmarks. A large-scale, diverse dataset with high-quality annotation is a pivot for lane detection. Along with the progress of lane detection approaches, numerous datasets have been proposed [25,21,59,49,4,37,55,10]. However, they usually fit into one or the other lane detection scenario. Tab. 1 depicts more details of the existing benchmarks and their comparison with our proposed OpenLane dataset. OpenLane is the first large-scale, realistic 3D lane dataset. It equips with a wide span of diversity in both data distribution and task applicability.

3D Lane Detection. As discussed in Section 1, planar assumption does not always reserve in some cases, *i.e.*, uphill/downhill, bump. Several approaches [33,5,3] utilize multi-modal or multi-view sensors, such as a stereo camera or LiDAR, to get the 3D ground topology. However, these sensors have shortages of high cost in hardware and computation resources, confining their practical applications. Recently, some monocular methods [15,19,23,30] take a single image and employ IPM to predict lanes in 3D space. 3D-LaneNet [15] is the pioneering work in this domain with one simple end-to-end neural network, which adopts STN [22] to accomplish the spatial projection of features. Gen-LaneNet [19] builds on top of 3D-LaneNet and designs a two-stage network for decoupling the segmentation encoder and 3D lane prediction head. These two approaches [15,19] suffer from improper feature transformation and unsatisfying performance in curving or crush turn cases. Confronted with the issues above, we bring in PersFormer to provide better feature representation and optimize anchor design to unify 2D and 3D lane detection simultaneously.

3 Methodology

In this section, we propose PersFormer, a unified 2D/3D lane detection framework with Transformer. We first describe the problem formulation, followed by an introduction to the overall structure in Section 3.1. In Section 3.2, we present

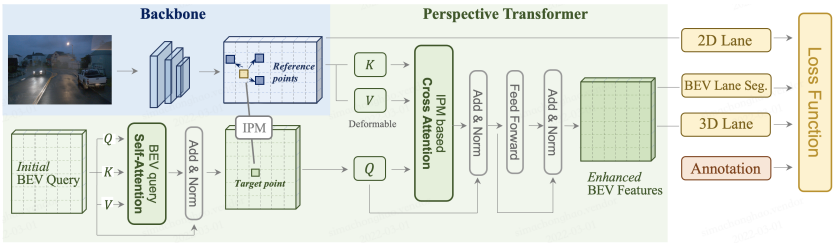

Fig. 2. Our proposed PersFormer pipeline. The core is to learn a spatial feature transformation from front view to BEV space so that the generated BEV features at target point would be more representative by attending local context around reference point. PersFormer consists of the self-attention module to interact with its own BEV queries; the cross-attention module that takes the key-value pair from the IPM-based front view features to generate fine-grained BEV feature

Perspective Transformer, an explicit feature transformation module from front view to BEV space by the aid of camera parameters. In Section 3.3, we give details on the anchor design to unify 2D/3D tasks and in Section 3.4 we further elaborate on the auxiliary task and loss function to finalize our training strategy.

Problem Formulation. Given an input image $I_{org} \in \mathbb{R}^{H_{org} \times W_{org}}$, the goal of PersFormer is to predict a collection of 3D lanes $L_{3D} = \{l_1, l_2, \ldots, l_{N_{3D}}\}$ and 2D lanes $L_{2D} = \{l_1, l_2, \ldots, l_{N_{2D}}\}$, where N_{3D}, N_{2D} are the total number of 3D lanes in the pre-defined BEV range and 2D lanes in the original image space (front view) respectively. Mathematically, each 3D lane l_d is represented by an ordered set of 3D coordinates:

$$l_d = \left[(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_{N_d}, y_{N_d}, z_{N_d})\right], \tag{1}$$

where d is the lane index, and N_d is the max number of sample points of this lane. The form of 2D lane is represented similarly with 2D coordinate (u, v) accordingly. Each lane has a categorical attribute $c_{\text{3D/2D}}$, indicating the type of this lane (*e.g.*, single-white dash line). Also, for each point in a single 2D/3D lane, there exists an attribute property indicating whether the point is visible or not, denoted by $\mathbf{vis}_{\text{fv/bev}}$ as a vector for the lane.

3.1 Approach Overview

The overall structure, as illustrated in Fig. 2, consists of three parts: the backbone, the Perspective Transformer, and lane detection heads. The backbone takes the resized image as input and generates multi-scale front view features, where the popular ResNet variant [47] is adopted. Note that these features might suffer from the defect of scale variance, occlusion, *etc.* - residing from the inherent feature extraction in the front view space. The Perspective Transformer takes the front view features as input and generates BEV features by the aid of camera intrinsic and extrinsic parameters. Instead of simply projecting the one-to-one

feature correspondence from the front view to BEV, we introduce Transformer to attend local context and aggregate surrounding features to form a robust representation in BEV. By doing so, we learn the inverse perspective mapping from front view to BEV in an elegant manner with Transformer. Finally, the lane detection heads are responsible for predicting 2D/3D coordinates as well as lane types. The 2D/3D detection heads are referred to as LaneATT [46] and 3D-LaneNet [15], with modification on the structure and anchor design.

3.2 Proposed Perspective Transformer

We present Perspective Transformer, a spatial transformation method that combines camera parameters and data-driven learning procedures. The general idea of Perspective Transformer is to use the coordinates transformation matrix from IPM as a reference to generate BEV feature representation, by attending related region (local context) in front view feature. On the assumption that the ground is flat and the camera parameters are given, a classical IPM approach calculates a set of coordinate mapping from front-view to BEV, where the BEV space is defined on the flat ground (see [20], Section 8.1.1). Given a point p_{fv} with its coordinate (u, v) in the front-view feature $F_{\text{fv}} \in \mathbb{R}^{H_{\text{fv}} \times W_{\text{fv}} \times C}$, IPM maps the point p_{fv} to the corresponding point p_{bev} in BEV, where (x, y) is the coordinate in the BEV space $\mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times C}$. The transform is achieved with camera in/extrinsic and can be represented mathematically as:

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \alpha_{f2b} \cdot R_\theta \cdot K^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix}, \tag{2}$$

where α_{f2b} implies the scale factor between front-view and BEV, R_θ denotes the pitch rotation matrix from extrinsic, K is the intrinsic matrix, and h stands for camera height. Such a transformation in Eqn.(2) enframes a strong prior on the attention unit in PerFormer to generate more representative BEV features.

The architecture of Perspective Transformer is inspired by popular approaches such as DETR [8], and consists of the self-attention module and cross-attention module (see Fig. 2). We differentiate from them in that the queries are not implicitly updated. However, instead, they are piloted by an explicit meaning - the physical location to detect objects or lanes in BEV. In the **self-attention** module, the output Q_{bev} descends from the triplet (key, value, query) input through their interaction. The formulation of such a self-attention can be described as:

$$Q_{\text{bev}} = \texttt{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

where $K, Q, V \in \mathbb{R}^{(H_{\text{bev}} \times W_{\text{bev}} \times C)}$ are the same query that is pre-defined in BEV, $\sqrt{d_k}$ is the dimensional normalized factor.

In the **cross-attention** module, the input query Q'_{bev} is the outcome of several additional layers feeding the self-attention output Q_{bev} as input. Note that Q'_{bev} is an explicit feature representation as to which part in BEV should be

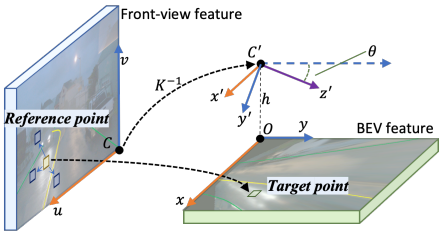

Fig. 3. Generation of keys in the cross attention. Point (x, y) in BEV space casts the corresponding point (u, v) in front view through intermediate state $(x^{'}, y^{'})$; by learning offsets, the network learns target-reference points mapping from green rectangles to yellow and related blue rectangles as keys to Transformer

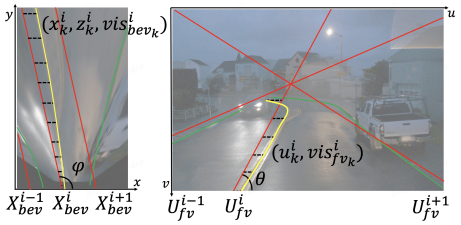

Fig. 4. Unifying anchor design in 2D and 3D. We first put curated anchors (red) in the BEV space (left), then project them to the front view (right). Offset x_k^i and u_k^i (dashed line) are predicted to match ground truth (yellow and green) to anchors. The correspondence is thus built, and features are optimized together

paid more attention since the generation of queries is location-sensitive in BEV. This is quite different compared with queries that do not consider view transformation in most Vision Transformers [53,18,62]. Furthermore, the intuition behind employing Transformer to map features from front view to BEV is that such an attention mechanism would automatically attend which part of features contribute *most* towards the target point (query) in the destination view. The direct feature transformation would suffer from camera parameter noise or scale variance issues, as discussed and illustrated in Section 1. Note that the naive Transformer cannot be applied directly since the number of key-value pairs is huge and thus be confined by computational burden. Inspired by Deformable DETR [62], we attend partial key-value pairs around the local region in a learnable manner to save cost and improve efficiency.

Fig. 3 depicts the feature transformation process and the generation of key-value pairs in cross-attention. Specifically, given a query point (x, y) in the target BEV map $Q^{'}_{\text{bev}}$, we project it to the corresponding point (u, v) in the front view via Eqn.(2). As does similarly in [62], we learn some offsets based on point (u, v) to generate a set of most related points around it. These learned points, together with (u, v) are defined as *reference points*. They contribute most to the query point (x, y), defined as *target point*, in BEV-space. The reference points serve as the surrounding context in the local region that contributes most to the feature representation from perspective view to BEV space. They are the desired keys we try to find, and their features are values for the cross attention module. Note that the initial locations of reference points from IPM are used as preliminary locations for the coordinate mapping; the location are adjusted gradually during the learning procedure, which is the core role of Deformable Attention.

As a result, the output of the cross-attention module can be formulated as:

$$F_{\text{bev}} = \texttt{DeformAttn}(Q^{'}_{\text{bev}}, F_{\text{fv}}, p_{\text{fv2bev}}), \tag{4}$$

where $F_{\text{bev}} \in \mathbb{R}^{(H_{\text{bev}} \times W_{\text{bev}} \times C)}$ is the final desired features for the subsequent 3D head to get lane predictions, Q'_{bev} denotes the input queries, $F_{\text{fv}} \in \mathbb{R}^{(H_{\text{fv}} \times W_{\text{fv}} \times C)}$ indicates the front view features from backbone, and p_{fv2bev} is the IPM-inited coordinate mapping from front view to BEV space. Considering F_{fv} and p_{fv2bev} with the deformable unit, we get the explicit transformed BEV feature F_{bev}.

To sum up, Perspective Transformer extracts front-view features among the reference points to construct representative BEV features. As demonstrated in Section 5, such a feature transformation in an aggregation spirit via Transformer is proven to perform better than a direct IPM-based projection across views.

3.3 Simultaneous 2D and 3D Lane Detection

Although the main focus in this paper lies in 3D detection, we formulate the PersFormer framework to detect 2D and 3D lanes in one shot. On one side, 2D lane detection in the perspective view still draws interest in the community as part of the general high-level vision problems [1,46,28,39]; on the other side, unifying 2D and 3D tasks are naturally feasible since the BEV features to predict 3D outputs descend from the counterpart in the 2D branch. An end-to-end unified framework would leverage features and benefit from the co-learning optimization process as proven in most multi-task literature [27,50,24].

Unified anchor design. Since our method is anchor-based detection, the core issue to achieve the unified framework is to integrate anchors in both 2D and 3D. Unfortunately, anchors in these two domains usually do not share similar distribution. For example, the popular 2D approach LaneATT [46] settles too many anchors, spanning different directions in the image; while the recent 3D work Gen-LaneNet [19] puts too few anchors, which are parallel and sparse in BEV. Based on these observations, we thereby design anchors such that the redesigned anchors could leverage the network to optimize shared features across *two* domains. We start with several groups of anchors (here, the group number is set to 7) sampled with different incline angles in the BEV space and then projected to the front view. Fig. 4 elaborates on the integration of 2D and 3D anchors. Below we describe how the lane line is modeled via anchors.

3D anchor design. To match ground truth lanes tightly, the anchors are placed approximately longitudinal along x-axis, with an incline angle φ. As denoted in Fig. 4(left), the initial line (equally spaced) with staring position along x-axis is denoted by X^i_{bev} for each anchor i. Similar to anchor regression in object detection, the network predicts the relative offset $\mathbf{x}^i$ w.r.t. the initial position X^i_{bev}; hence the resultant lane prediction along x-axis is $(\mathbf{x}^i + X^i_{\text{bev}})$. As indicated in Eqn.(1), each lane is represented as a number of N_d points. The prediction head generates three vectors related to lane shape as follows:

$$(\mathbf{x}^i, \mathbf{z}^i, \mathbf{vis}^i_{\text{bev}}) = \{(x^{(i,k)}, z^{(i,k)}, \text{vis}^{(i,k)}_{\text{bev}})\}_{k=1}^{N_d} \tag{5}$$

where $\mathbf{z}^i$ is the lane height in 3D sense, the binary $\text{vis}^{(i,k)}_{\text{bev}}$ denotes the visibility of each location k in lane i, which controls the endpoint or length of a lane. Note that the lane position along y-axis does not need to be predicted since each y

value of the N_d samples in a lane is pre-defined - we predict the $x^{(i,k)}$ value at the corresponding (fixed) y location. To sum up, the description of a lane's location in the world coordinate system is denoted as $(\mathbf{x}^i + X^i_{\text{bev}}, \mathbf{y}, \mathbf{z}^i)$.

2D anchor design. The anchor description and prediction are similar to those defined in 3D view, except that the (u, v) is in 2D space and there is no height (see Fig. 4(right)). We omit the detailed notations for brevity. It is worth mentioning that each 3D anchor X^i_{bev} with an incline angle φ corresponds to a specific 2D anchor U^i_{fv} with the incline angle θ; the connection is built via the projection in Eqn.(2). We achieve the goal of unifying 2D and 3D tasks simultaneously by setting the *same* set of anchors. Such a design would optimize features together and features being more aligned and representative across views.

3.4 Prediction Loss

Binary Segmentation under BEV. As do in many preceding work [54,35,21], adding more intermediate supervision into the network training would boost the performance of network. Since lane detection belongs to image segmentation and requires general large resolution, we concatenate a U-Net structure [40] head on top of the generated BEV features. Such an auxiliary task is to predict lanes in BEV, but instead in a conventional 2D segmentation manner, aiming for better feature representation for the main task. The ground truth S_{gt} is a binary segmentation map projected from 3D lane ground truth to the BEV space. The prediction output is denoted by S_{pred} and owns the same size as S_{gt}.

Loss function. Equipped with the anchor representation and segmentation head aforementioned, we summarize the overall loss. Given an image input and its ground truth labels, it finally computes a sum of all anchors' loss; the loss is a combination of the 2D lane detection, 3D lane detection and intermediate segmentation with learnable weights (α, β, γ) accordingly:

$$\mathcal{L} = \sum_i \alpha \mathcal{L}_{\text{2D}}(c^i_{\text{2D}}, \mathbf{u}^i, \mathbf{vis}^i_{\text{fv}}) + \beta \mathcal{L}_{\text{ 3D}}\ (c^i_{\text{3D}}, \mathbf{x}^i, \mathbf{z}^i, \mathbf{vis}^i_{\text{bev}}) + \gamma \mathcal{L}_{\text{seg}}(S_{\text{pred}}), \quad (6)$$

where $c^i_{(\cdot)}$ is the predicted lane category in 2D and 3D domain respectively. The loss input above shows the prediction part only; we omit the ground truth notation for brevity. The loss of lane category classification for the 2D/3D task is the cross-entropy; the loss of lane shape regression is the l_1 norm; the loss of lane visibility prediction is the binary cross-entropy loss. The loss of the auxiliary task is a binary cross-entropy loss between two segmentation maps.

4 OpenLane: A Large-scale Realistic 3D Lane Benchmark

4.1 Highlights over Previous Benchmarks

OpenLane is the *first* real world 3D lane dataset and the *largest* scale to date compared with existing benchmarks. We construct OpenLane on top of the influential Waymo Open dataset [45], following the same data format and evaluation

Table 1. Comparison of OpenLane with existing benchmarks. "Avg. Length" denotes the average time duration of segments. "Inst. Anno." indicates whether lanes are annotated instance-wise (c.f. semantic-wise). "Track. Anno." implies if a lane has a unique tracking ID. Numbers in '#Frames' are the number of annotated frames / total frames respectively. Details of "Scenario" can be found in Appendix

Dataset	#Segments	#Frames	Avg. Length	Inst. Anno.	Track. Anno.	Max #Lanes	Line Category	Scenario
Caltech Lanes [2]	4	1224/1224	-	✓	✗	4	-	Easy
TuSimple [49]	6.4K	6.4K/128K	1s	✓	✗	5	-	Easy
3D Synthetic [19]	-	10K/10K	-	✓	-	6	-	Easy
VIL-100 [61]	100	10K/10K	10s	✓	✗	6	10	Medium
VPG [25]	-	20K/20K	-	✗	-	-	7	Medium
OpenDenseLane [10]	1.7K	57K/57K	-	✓	✗	-	4	Medium
LLAMAS [4]	14	79K/100K	-	✓	✗	4	-	Easy
ApolloScape [21]	235	115K/115K	16s	✗	✗	-	13	Medium
BDD100K [59]	100K	100K/120M	40s	✗	✗	-	11	Medium
CULane [37]	-	133K/133K	-	✓	-	4	-	Medium
CurveLanes [55]	-	150K/150K	-	✓	-	9	-	Medium
ONCE-3DLanes [56]	-	211K/211K	-	✓	-	8	-	Medium
OpenLane	1K	**200K/200K**	20s	✓	✓	**24**	**14**	**Hard**

pipeline - leveraging existent practice in the community so that users would not handle additional rules for a new benchmark. Tab. 1 compares OpenLane with existing counterparts in various aspects. In short, OpenLane owns 200K frames and over 880K carefully annotated lanes, 33% and 35% more compared with existing largest lane dataset CurveLanes [55] respectively, with rich annotations.

We annotate all the lanes in each frame, including those in the *opposite* direction if no curbside exists in the middle. Due to the complicated lane topology, *e.g.*, intersection/roundabout, one frame could contain as many as **24** lanes in OpenLane. Statistically, about 25% frames of OpenLane have more than 6 lanes, which exceeds the maximum number in most lane datasets. **14** lane categories are annotated alongside to cover a wide range of lane types in most scenarios, including road edges. Double yellow solid lanes, single white solid and dash lanes take up almost 90% of total lanes. This is imbalanced, and yet it falls into a long-tail distribution problem, which is common in realistic scenarios. In addition to the lane detection task, we also annotate: (a) scene tags, such as weather and locations; (b) the closest-in-path object (CIPO), which is defined as the most concerned target w.r.t. ego vehicle; such a tag is quite pragmatic for subsequent modules as in planning/control, besides a whole set of objects from perception. An annotation example is provided in Fig. 5(d), along with some typical samples in existing 2D lane datasets in Fig. 5(a-c). The detailed statistics, annotation criterion and visualization can be found in Appendix.

4.2 Generation of High-quality Annotation

Building a real-world 3D lane dataset has challenges mainly in an accurate localization system and occlusions. We compare several popular sensor datasets [9,45,6] by projecting 3D object annotations to image planes and constructing 3D

Fig. 5. Annotation samples of OpenLane compared with other lane datasets. OpenLane is challenging with more lane categories per frame in average and has rich labels including scene, weather, hours, CIPO

scene maps using both learning-based [48] or SLAM algorithms [42,43]. The reconstruction precision and scalability of Waymo Open Dataset [45] outperforms other candidates, leading to employing it as our basis.

Primarily, we generate the necessary high-quality 2D lane labels. They contain the final annotations of tracking ID, category, and 2D points ground truth. Then for each frame, the point clouds are first filtered with the original 3D object bounding boxes and then projected back into the corresponding image. We further keep those points related to 2D lanes only with a certain threshold. However, the output directly after a static threshold filtering could lead to an unsatisfying ground truth due to the perspective scaling issue. To solve this and keep the slender shape of lanes, we use the filtered point clouds to interpolate the 3D position for each point in 2D annotations. Afterward, with the help of the localization system, 3D lane points in frames within a segment could be spliced into long, high-density lanes. This process could bring some unreasonable parts into the current frame; thus, points in one lane whose 2D projections are higher than the ending position of its 2D annotation are labeled as invisible. A smoothing step is ultimately deployed to filtrate any outliers and generate the 3D labeling results. We omit some technical details, such as how to deal with a large U-turn during smoothing, and we refer the audience to Appendix.

5 Experiments

We examine PersFormer on two 3D lane benchmarks, the newly proposed real-world OpenLane dataset, and the synthetic Apollo dataset. For both 3D lane datasets, we follow the evaluation metrics designed by Gen-LaneNet [19], with additional category accuracy on OpenLane dataset. For the 2D task, the classical metric in CULane [37] is adopted. We put correlated details in Appendix.

5.1 Results on OpenLane

We provide 3D and 2D evaluation results on the proposed OpenLane dataset. In order to evaluate the models thoroughly, we report F-Score on the entire validation set and different scenario sets. The scenario sets are selected from the entire validation set based on the scene tags of each frame. In Tab. 2, PersFormer gets the highest F-Score on the entire validation set and every scenario set, surpassing

Table 2. Comparison with other open-sourced 3D methods on OpenLane. PersFormer achieves the best F-Score on the entire validation set and every scenario set

Method	All	Up & Down	Curve	Extreme Weather	Night	Intersection	Merge & Split
3D-LaneNet [15]	44.1	40.8	46.5	47.5	41.5	32.1	41.7
Gen-LaneNet [19]	32.3	25.4	33.5	28.1	18.7	21.4	31.0
PersFormer (ours)	**50.5**	**42.4**	**55.6**	**48.6**	**46.6**	**40.0**	**50.7**

Table 3. Comparison with state-of-the-art 2D method on OpenLane. The result from the 2D head of PersFormer also achieves competitive performance

Method	All	Up & Down	Curve	Extreme Weather	Night	Intersection	Merge & Split
LaneATT-S [46]	28.3	25.3	25.8	32.0	27.6	14.0	24.3
LaneATT-M [46]	31.0	28.3	27.4	34.7	30.2	17.0	26.5
CondLaneNet-S [28]	52.3	55.3	57.5	45.8	46.6	48.4	45.5
CondLaneNet-M [28]	55.0	58.5	59.4	49.2	48.6	50.7	47.8
CondLaneNet-L [28]	**59.1**	**62.1**	**62.9**	**54.7**	**51.0**	**55.7**	**52.3**
PersFormer (ours)	42.0	40.7	46.3	43.7	36.1	28.9	41.2

Table 4. Comprehensive 3D Lane evaluation under different metrics. On the strength of unified anchor design, PersFormer outperforms previous 3D methods on the metrics of far error while retains comparable results on near error (m). * denotes projecting 2D lane results from CondLaneNet [28] to BEV using IPM

Method	F-Score	Category Accuracy	X error near	X error far	Z error near	Z error far
3D-LaneNet [15]	44.1	-	**0.479**	0.572	0.367	0.443
Gen-LaneNet [19]	32.3	-	0.591	0.684	0.411	0.521
Cond-IPM*	36.6	-	0.563	1.080	0.421	0.892
PersFormer (ours)	**50.5**	**92.3**	0.485	**0.553**	**0.364**	**0.431**

previous SOTA methods in varying degrees. In Tab. 3, PersFormer outperforms LaneATT [46], which is our baseline 2D method, by **11%**. Detailed comparison with previous 3D SOTAs is presented in Tab. 4. PersFormer outperforms the previous best method in F-Score by **6.4%**, realizes satisfying accuracy on the classification of lane type, and presents the first baseline result. Note that PersFormer is not satisfying on the metric of near error on x-axis. This is probably because the unified anchor design is more suitable in fitting the main body of a lane rather than the starting point. Qualitative results are shown in Fig. 6, indicating that PersFormer is good at catching dense and unapparent lanes in usual autonomous driving scenes. Overall, PersFormer reaches the best performance on 3D lane detection and gains remarkable improvement in 2D on OpenLane.

5.2 Results on Apollo 3D Synthetic

We evaluate PersFormer on Apollo 3D Lane Synthetic dataset [19]. In Tab. 5, while limited by the scale of the dataset (10K frames), our PersFormer still

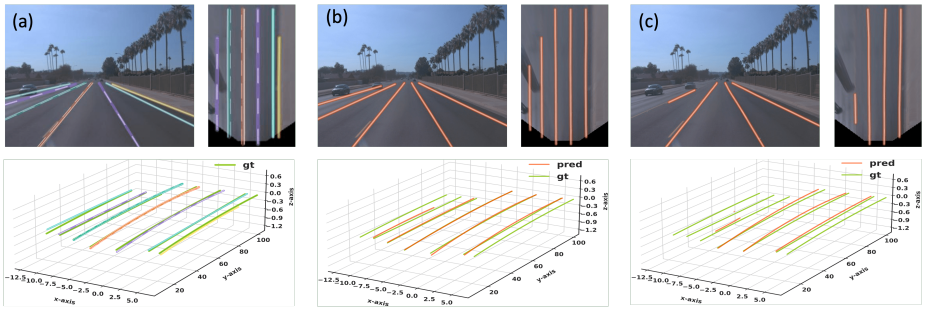

Fig. 6. Qualitative results of PersFormer(a), 3D-LaneNet(b) [15], and Gen-LaneNet(c) [19]. Under a straight road scenario, PersFormer can provide lane-type information and even detect subtle curbside while other methods are missing it

Table 5. Comparison with previous 3D methods on Apollo 3D Lane Synthetic. PersFormer achieves best F-Score on every scene set with comparable X/Z error (m)

Scene	Method	F-Score	X error near	X error far	Z error near	Z error far
Balanced Scenes	3D-LaneNet [15]	86.4	0.068	0.477	0.015	**0.202**
	Gen-LaneNet [19]	88.1	0.061	0.496	0.012	0.214
	3D-LaneNet(l/att) [23]	91.0	0.082	0.439	0.011	0.242
	Gen-LaneNet(l/att) [23]	90.3	0.080	0.473	0.011	0.247
	CLGo [30]	91.9	0.061	0.361	0.029	0.250
	PersFormer (ours)	**92.9**	**0.054**	**0.356**	**0.010**	0.234
Rarely Observed	3D-LaneNet [15]	72.0	0.166	0.855	0.039	**0.521**
	Gen-LaneNet [19]	78.0	0.139	0.903	0.030	0.539
	3D-LaneNet(l/att) [23]	84.1	0.289	0.925	0.025	0.625
	Gen-LaneNet(l/att) [23]	81.7	0.283	0.915	0.028	0.653
	CLGo [30]	86.1	0.147	**0.735**	0.071	0.609
	PersFormer (ours)	**87.5**	**0.107**	0.782	**0.024**	0.602
Vivual Variants	3D-LaneNet [15]	72.5	0.115	0.601	0.032	**0.230**
	Gen-LaneNet [19]	85.3	0.074	0.538	0.015	0.232
	3D-LaneNet(l/att) [23]	85.4	0.118	0.559	0.018	0.290
	Gen-LaneNet(l/att) [23]	86.8	0.104	0.544	0.016	0.294
	CLGo [30]	87.3	0.084	0.464	0.045	0.312
	PersFormer (ours)	**89.6**	**0.074**	**0.430**	**0.015**	0.266

achieves the best F-Score on every scene set. In terms of X/Z error, our model gets comparable results compared to previous methods.

5.3 Ablation Study

We present ablation studies on the anchor design, multi-task strategy, transformer-based view transformation, and auxiliary segmentation task. We mainly report the improvement on 3D lane detection and provide related results on 2D task.

Anchor design and multi-task. Starting with a pure 3D lane detection framework (similar to 3D-LaneNet [15]), PersFormer gains **1.7%** by adopting multi-task scheme (Exp.2) and **0.98%** with new anchor design (Exp.4) respec-

Table 6. Ablative Study on a 300 segments subset of OpenLane. Exp.1 is the baseline 3D method, growing with anchor design and multi-task learning (Exp.2-5). The performance culminates with our spatial feature transformation module and explicit BEV supervision (Exp.6,7)

Exp.	Unified Anchor	3D Det	2D Det	Perspective Transformer	Binary Seg	3D F-Score	2D F-Score
1		✓				41.77	-
2		✓	✓			43.49	32.33
3	✓		✓			-	34.90
4	✓	✓				42.75	-
5	✓	✓	✓			44.29	34.98
6	✓	✓	✓	✓		46.62	37.00
7	✓	✓	✓	✓	✓	47.79	42.00

tively. By jointly using the new anchor and multi-task trick, PersFormer acquires an improvement of **2.5%** in 3D task and **2.6%** in 2D task (Exp.5).

Spatial feature transformation. By using Perspective Transformer with the new anchor design, the improvement increases to **4.9%** (Exp.6), almost doubling the previous improvement. Adding auxiliary binary segmentation task further brings an improvement to **6.02%** (Exp.7), which is our complete model. These ablations support our assumption that PersFormer indeed generates a fine-grained BEV feature, and the spatial feature transformation does illustrate its importance in 3D lane detection task. Surprisingly, a better BEV feature helps 2D task a lot as well, improving **9.7%** (Exp.7).

6 Conclusions

In this paper, we have proposed Persformer, a novel Transformer-based 2D/3D lane detector, along with OpenLane, a large-scale realistic 3D lane dataset. We demonstrate experimentally that a fine-grained BEV feature with explicit prior and supervision can significantly improve the performance of lane detection. Meanwhile, a large-scale real-world 3D lane dataset effectively align the demand from both the academic and the industrial side.

Acknowledgments

The project is partially supported by the Shanghai Committee of Science and Technology (Grant No. 21DZ1100100). This work was supported in part by National Key Research and Development Program of China (2020AAA0107600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). We would like to acknowledge the great support from SenseBee labelling team at SenseTime Research, constructive contribution from Zihan Ding at BUAA, and the fruitful discussions and comments for this project from Zhiqi Li, Yuenan Hou, Yu Liu, Jing Shao, Jifeng Dai.

References

1. Abualsaud, H., Liu, S., Lu, D.B., Situ, K., Rangesh, A., Trivedi, M.M.: Laneaf: Robust multi-lane detection with affinity fields. RA-L (2021) 2, 8
2. Aly, M.: Real time detection of lane markers in urban streets. In: IV (2008) 10
3. Bai, M., Mattyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S.K., Urtasun, R.: Deep multi-sensor lane detection. In: IROS (2018) 4
4. Behrendt, K., Soussan, R.: Unsupervised labeled lane markers using maps. In: ICCV (2019) 4, 10
5. Benmansour, N., Labayrade, R., Aubert, D., Glaser, S.: Stereovision-based 3d lane detection system: a model driven approach. In: ITSC (2008) 4
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 10
7. Can, Y.B., Liniger, A., Paudel, D.P., Van Gool, L.: Structured bird's-eye-view traffic scene understanding from onboard images. In: ICCV (2021) 4
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020) 3, 6
9. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR (2019) 10
10. Chen, X., Liao, W., Liu, B., Yan, J., He, T.: Opendenselane: a new lidar-based dataset for hd map construction. In: ICME (2022) 4, 10
11. Chitta, K., Prakash, A., Geiger, A.: Neat: Neural attention fields for end-to-end autonomous driving. In: ICCV (2021) 4
12. Comma.ai: Openpilot. https://github.com/commaai/openpilot 2
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 3
14. Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., Schmid, C.: Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In: CVPR (2020) 4
15. Garnett, N., Cohen, R., Pe'er, T., Lahav, R., Levi, D.: 3d-lanenet: End-to-end 3d multiple lane detection. In: ICCV (2019) 2, 4, 6, 12, 13
16. Garnett, N., Uziel, R., Efrat, N., Levi, D.: Synthetic-to-real domain adaptation for lane detection. In: ACCV (2020) 2
17. Gu, J., Sun, C., Zhao, H.: Densetnt: End-to-end trajectory prediction from dense goal sets. In: ICCV (2021) 4
18. Guan, T., Wang, J., Lan, S., Chandra, R., Wu, Z., Davis, L., Manocha, D.: M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In: WACV (2022) 4, 7
19. Guo, Y., Chen, G., Zhao, P., Zhang, W., Miao, J., Wang, J., Choe, T.E.: Gen-lanenet: A generalized and scalable approach for 3d lane detection. In: ECCV (2020) 2, 4, 8, 10, 11, 12, 13
20. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004) 6
21. Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The apolloscape open dataset for autonomous driving and its application. TPAMI (2019) 4, 9, 10

22. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NeurIPS (2015) 2, 4
23. Jin, Y., Ren, X., Chen, F., Zhang, W.: Robust monocular 3d lane detection with dual attention. In: ICIP (2021) 4, 13
24. Kumar, V.R., Yogamani, S., Rashed, H., Sitsu, G., Witt, C., Leang, I., Milz, S., Mäder, P.: Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving. RA-L (2021) 8
25. Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., So Kweon, I.: Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In: ICCV (2017) 2, 4, 10
26. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. arXiv preprint arXiv:2203.17270 (2022) 4
27. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-task multi-sensor fusion for 3d object detection. In: CVPR (2019) 8
28. Liu, L., Chen, X., Zhu, S., Tan, P.: Condlanenet: a top-to-down lane detection framework based on conditional convolution. In: CVPR (2021) 2, 8, 12
29. Liu, P.L.: Monocular bev perception with transformers in autonomous driving. https://towardsdatascience.com/monocular-bev-perception-with-transformers-in-autonomous-driving-c41e4a893944 3, 4
30. Liu, R., Chen, D., Liu, T., Xiong, Z., Yuan, Z.: Learning to predict 3d lane shape and camera pose from a single image via geometry constraints. In: AAAI (2022) 4, 13
31. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021) 3
32. Meyer, A., Salscheider, N.O., Orzechowski, P.F., Stiller, C.: Deep semantic lane segmentation for mapless driving. In: IROS (2018) 2
33. Nedevschi, S., Schmidt, R., Graf, T., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C.: 3d lane detection system based on stereovision. In: ITSC (2004) 4
34. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. In: IV (2018) 2
35. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV (2016) 9
36. Ngiam, J., Vasudevan, V., Caine, B., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In: ICLR (2021) 4
37. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In: AAAI (2018) 2, 4, 10, 11
38. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: CVPR (2021) 4
39. Qu, Z., Jin, H., Zhou, Y., Yang, Z., Zhang, W.: Focus on local: Detecting lane marker from bottom up via key point. In: CVPR (2021) 2, 8
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 9
41. Saha, A., Maldonado, O.M., Russell, C., Bowden, R.: Translating images into maps. In: ICRA (2022) 4

42. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Daniela, R.: Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: IROS (2020) 11
43. Shan, T., Englot, B., Ratti, C., Daniela, R.: Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In: ICRA (2021) 11
44. Su, J., Chen, C., Zhang, K., Luo, J., Wei, X., Wei, X.: Structure guided lane detection. In: IJCAI-21 (2021) 2
45. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020) 9, 10, 11
46. Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Keep your eyes on the lane: Real-time attention-guided lane detection. In: CVPR (2021) 2, 6, 8, 12
47. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019) 5
48. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In: NeurIPS (2021) 11
49. TuSimple: https://github.com/TuSimple/tusimple-benchmark (2017) 4, 10
50. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: A survey. TPAMI (2021) 8
51. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 3
52. Wang, J., Mei, T., Kong, B., Wei, H.: An approach of lane detection based on inverse perspective mapping. In: ITSC (2014) 2
53. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: CoRL (2022) 4, 7
54. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016) 9
55. Xu, H., Wang, S., Cai, X., Zhang, W., Liang, X., Li, Z.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: ECCV (2020) 4, 10
56. Yan, F., Nie, M., Cai, X., Han, J., Xu, H., Yang, Z., Ye, C., Fu, Y., Michael, B.M., Zhang, L.: Once-3dlanes: Building monocular 3d lane detection. In: CVPR (2022) 10
57. Yang, W., Li, Q., Liu, W., Yu, Y., Ma, Y., He, S., Pan, J.: Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In: CVPR (2021) 4
58. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In: CVPR (2020) 4
59. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020) 4, 10
60. Yu, Z., Ren, X., Huang, Y., Tian, W., Zhao, J.: Detecting lane and road markings at a distance with perspective transformer layers. In: ITSC (2020) 2
61. Zhang, Y., Zhu, L., Feng, W., Fu, H., Wang, M., Li, Q., Li, C., Wang, S.: Vil-100: A new dataset and a baseline model for video instance lane detection. In: ICCV (2021) 10

62. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: ICLR (2021) 3, 7