2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

WS-3D-Lane: Weakly Supervised 3D Lane Detection With 2D Lane Labels

Jianyong Ai[1] Wenbo Ding[1] Jiuhua Zhao[1] Jiachen Zhong[1]

***Abstract*— Compared to 2D lanes, real 3D lane data is difficult to collect accurately. In this paper, we propose a novel method for training 3D lanes with only 2D lane labels, called weakly supervised 3D lane detection *WS-3D-Lane*. By assumptions of constant lane width and equal height on adjacent lanes, we indirectly supervise 3D lane heights in the training. To overcome the problem of the dynamic change of the camera pitch during data collection, a camera pitch self-calibration method is proposed. In anchor representation, we propose a double-layer anchor with non-maximum suppression (NMS) method, which enables the anchor-based method to predict two lane lines that are close. Experiments are conducted on the base of 3D-LaneNet under two supervision methods. Under weakly supervised setting, our *WS-3D-Lane* outperforms previous 3D-LaneNet: F-score rises to 92.3% on Apollo 3D synthetic dataset, and F1 rises to 74.5% on ONCE-3DLanes. Meanwhile, *WS-3D-Lane* in purely supervised setting makes more increments and outperforms state-of-the-art. To the best of our knowledge, *WS-3D-Lane* is the first try of 3D lane detection under weakly supervised setting. Our code is available on `https://github.com/SAIC-Vision/WS-3D-Lane`.**

I. INTRODUCTION

Image-based lane detection is an important perception task in autonomous driving. Traditional methods [1]–[3] detect or segment lanes in image domain and then project the results onto a flat ground, which is inaccurate in case of uphill or downhill and may lead to dangerous behavior of an autonomous vehicle. Compared to 2D lane, 3D lane detection [4]–[7] can directly obtain the slope information of the lane, to help autonomous vehicles make better decisions. However, the data collection of 3D lanes in real-world is very difficult. Current published datasets, such as ONCE-3DLanes [7], [8] and OpenLane [6], are collected by LIDAR and front-view images with 2D labels. 3D points in LIDAR are converted to camera coordinate system and picked out by 2D lane labels. Tricks like filters and curve fitting are adopted to reduce errors. The quality of these datasets has room to improve. In the last decade, abundant high quality 2D lanes are collected [9]–[12]. An intuitive thinking is fully taking advantage of these 2D lane data. Therefore, we propose a weakly supervised training method for 3D lane detection, called *WS-3D-Lane*, which uses 2D lane labels to supervise the training of 3D lane detection. The framework of *WS-3D-Lane* is shown in Fig.1. Based on the assumption of constant lane width and equal height on adjacent lane lines, our *WS-3D-Lane* utilizes the predicted lane width difference (Fig.1B) and height difference on adjacent lane lines (Fig.1C) to supervise the height of lane line.

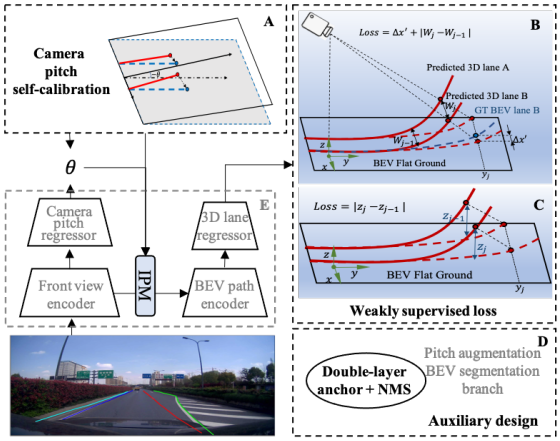

Fig. 1. **Framework for *WS-3D-Lane*.** We use 3D-LaneNet [5] (E) as our basic network. Our weakly supervised loss function is designed in the assumptions of constant lane width (B) and equal height on adjacent lane lines (C). Camera pitch self-calibration method (A) is proposed for calculate the label of camera pitch per frame in real scenes. Auxiliary designs (D) are adopted in the training, including double-layer anchor with non-maximum suppression(NMS) method, BEV segmentation branch [6] and camera pitch augmentation [13]. Bold text is our major contributed designs.

Besides lane label, camera extrinsic is also crucial in 3D lane detection, especially for camera pitch angle [5], [14]. But camera pitch often suffers from shake and cannot be dynamically calibrated during data collection. Previous data collection method often provide the same camera extrinsic for every frame[1]. It might cause large errors on the ground truth in world coordinate system.Therefore, we proposed camera pitch self-calibration method (Fig.1A), to get the label of camera pitch per frame.

Furthermore, we propose double-layer anchor with non-maximum suppression (NMS) method which are able to improve the performance under both weakly supervised and purely supervised setting. Combined with two useful techniques: Pitch augmentation [13] and BEV segmentation [6], our auxiliary designs (Fig.1D) help model achieve the state-of-the-art performance.

Our contributions are summarized as follows:

- For the first time, our *WS-3D-Lane* provides a weakly supervised 3D lane detection method with only 2D lane labels, and outperforms previous 3D-LaneNet in the evaluation.

[1]Jianyong Ai, Wenbo Ding, Jiuhua Zhao and Jiachen Zhong are with SAIC AI Lab, 51 Zhengxue Rd., Shanghai, China (Email:{aijianyong, dingwenbo, zhaojiuhua, zhongjiachen}@saicmotor.com)

[1]By the time we finished the experiments in this work, these two datasets [6], [7] have not provided dynamic pose of each frame

- Our camera pitch self-calibration method provides a way to calculate accurate camera pitch per frame.
- With our auxiliary designs, our *WS-3D-Lane* is able to achieve the state-of-the-art level performance under both purely and weakly supervised setting.

II. RELATED WORKS

A. 3D Lane Detection

The methods of 3D lane detection could be divided into LIDAR-based [15]–[18], monocular [19]–[21] and multi-sensor [22]–[24]. Our work focuses on the monocular 3D lane detection.

The pioneering work 3D-LaneNet [5] predicts camera pitch, and converts the multi-scale image-view features to BEV features by the inverse perspective mapping(IPM) projection. It predicts 3D lanes on BEV path using anchors. Each anchor consists of lane probability and several reference points along y-axis. Each point contains the visibility and position of x-offset and z. 3Dlanenet+ [19] uses anchor-free representations to predict 3D lanes and gets better results on their private dataset. Gen-laneNet [4] makes a great contribution on the anchor representation. It predicts 2D points on BEV flat ground with the lane height, and converts to 3D points by geometric relationship. This improvement makes a great better performance on Apollo-Sim-3D. Persformer [6] achieves the state-of-the-art on three datasets via replacing IPM with transformer and adding several auxiliary tasks. However, the previous anchor representation is not able to predict two close lane lines, such as forks and curb with nearby lane line. We propose the double-layer anchor to solve this problem. Reference [13] is the first literature to use lane width in loss function. It contributes an auxiliary loss function to keep the same change rate of lane width and uses a greedy matching algorithm to calculate lane width. We use the lane width differently. Firstly, the lane width is calculated by the equivalent short and straight lines. Secondly, we focus on the weakly supervised setting and the constant lane width assumption is the main supervised signal for lane height without LIDAR.

B. Weakly Supervised 3D Perception

Previous works on 3D lane detection are fully supervised, but our work focuses on weakly supervised setting. Weakly supervised methods are widely used in 3D points segmentation and object detection [25]–[27], to save the cost of annotation of 3D point cloud. These methods are proposed to supervise the lacked part by rules and assumptions in different conditions, e.g. labeled 2D images with unlabeled 3D point cloud [28] and labeled virtual scene with unlabeled real scene [29]. To the best of our knowledge, no previous literature focuses on the weakly supervised 3D lane detection, and our *WS-3D-Lane* is the first work to explore it. For the difficulty of collecting 3D lane labels in real-world, *WS-3D-Lane* provides a novel method using 2D lane labels to supervise the 3D lane detection.

III. METHOD

For weakly supervised 3D lane detection, 2D lane labels should be converted to BEV flat ground with camera pitch and camera height first. Then we use the ground truth on BEV flat ground and our assumptions to supervise the model training. Following previous literature [4], [13], the world coordinate center is the projection point of the camera center on the ground, and its axes are shown in Fig.1B. Camera roll and yaw are set to 0.

A. Assumption of Constant 3D Lane Width

The height of the 3D lane is implicitly included in the BEV panel. As shown in Fig. 2, when the 3D lane is not flat, the lane width on the BEV flat ground changes. It tends to be wider when the front road is uphill and narrower when the front road is downhill. Equation (1) shows the geometric relationship between the point on the 3D lane (x,y,z) and its corresponding 2D projection lane point on the BEV flat ground $(x',y',0)$ [4], from which the relationship between the 3D lane width $W_{i,j}$ and the 2D lane width on the BEV flat ground $W'_{i,j}$ can be derived as in Equation (2), where $i \in \{0,1,2...\}$ means the serial number of lane line along the x-axis and $j \in \{0,1,2...\}$ means the serial number of anchor points along the y-axis. $W_{i,j}$ is the lane width between lane line i and $i-1$ at reference point j.

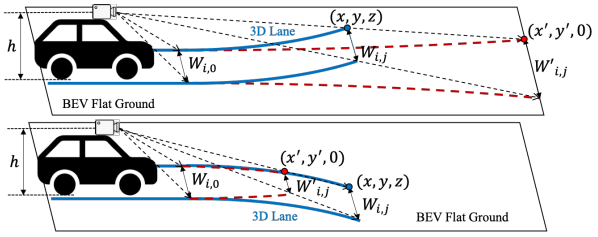

Fig. 2. **Changes of lane width on BEV flat ground when 3D lane is not flat.**

$$\frac{x}{x'} = \frac{y}{y'} = \frac{h-z}{h} \tag{1}$$

$$\frac{W_{i,j}}{W'_{i,j}} = \frac{h-z_{i,j}}{h} \tag{2}$$

In most cases, the lane width is fixed for the same lane, so we can assume that the lane width is a constant value $W_{i,0}$. At the closest point, i.e. $z \approx 0$, the width of the closest lane on BEV flat ground $W'_{i,0}$ is equal to $W_{i,0}$, as shown in Fig. 2.

For a point $P_{i,j}$ on a lane line, the key of lane width calculation is to find its corresponding closest point $P^0_{i,j}$ on its neighbored lane line. As shown in Fig. 3, we use several short and straight lines to represent the curves, so the point $P^0_{i,j}$ is approximated to the perpendicular foot of $P_{i,j}$ on straight line $P_{i-1,j}P_{i-1,j-1}$. If the lane line is represented by the anchor, $P_{i-1,j}$ and $P_{i-1,j-1}$ could be two neighbored anchor points on a line, and then the lane width could be calculated as follows:

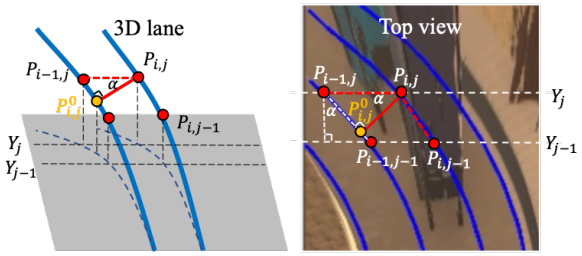

Fig. 3. **3D lane width calculation on bend.**

$$W_{i,j} = |P_{i,j}P^0_{i,j}| \approx |P_{i,j}P_{i-1,j}| \cos\alpha = \frac{|P_{i,j}P_{i-1,j}||P_{i-1,j}P_{i-1,j-1}|_{x=0}}{|P_{i-1,j}P_{i-1,j-1}|} \tag{3}$$

where,

$$\begin{cases} |P_{i,j}P_{k,l}| = \left\|(x_{i,j}-x_{k,l}, y_{i,j}-y_{k,l}, z_{i,j}-z_{k,l})\right\|_2 \\ |P_{i,j}P_{k,l}|_{x=0} = \left\|(0, y_{i,j}-y_{k,l}, z_{i,j}-z_{k,l})\right\|_2 \\ x_{i,j} = \hat{x'}_{i,j}\frac{h-z_{i,j}}{h} \\ y_{i,j} = \hat{y'}_{i,j}\frac{h-z_{i,j}}{h} \end{cases} \tag{4}$$

where, the symbol $\hat{x'}, \hat{y'}$ means the ground truth of x', y'.

Following previous literature [4], for a predicted anchor $X^A_{i,j}$ and its corresponding ground truth $\hat{X}^A_{i,j} = \left\{(\hat{p}^A_i, \hat{v}^A_{i,j}, \hat{x'}^A_{i,j})\right\}$, each anchor point at a pre-defined position along y-axis y_j predicts the x-offset $x'_{i,j}$ on BEV flat ground, the lane height $z_{i,j}$, and the visibility $v_{i,j}$. $\hat{p}^A_i$ is the probability of lane line in the anchor i. A loss function of weakly supervised 3D lane can be written as follows:

$$L_{width} = \sum_{i=1}^{N_p-1}\sum_{j=1}^{Y-1}\left\|W_{i,j} - W_{i,j-1}\right\|_1 \tag{5}$$

where, N_p is the number of the anchors where $\hat{p}^A_i = 1$. Y is the number of y steps in a anchor. This loss function is to force the model to predict lane with equal width.

B. Assumption of Equal Height on Adjacent Lanes

At the same distance along the y-axis, we assume the height of the lane line is equal to the height of its adjacent lanes. Therefore, in Fig. 3, the height of point $P_{i,j}$ is equal to $P_{i-1,j}$, and a loss function of weakly supervised 3D lane can be written as follows:

$$L_{height} = \sum_{i=1}^{N_p-1}\sum_{j=1}^{Y-1}\left\|z_{i,j} - z_{i-1,j}\right\|_1 \tag{6}$$

C. Camera Pitch Self-calibration

The camera pitch self-calibration for each frame is necessary for transforming 2D labels from front-view to BEV flat ground [4]. In the self-calibration, we assume that the lane height is 0 when it is close enough, i.e. the $y' < y_{close}$. Therefore, when the lane lines are not parallel on BEV flat ground, the camera pitch should be corrected. As shown in Fig. 4, we first assume camera pitch = 0 and transform the 2D lane lines to the BEV flat ground. Secondly, the transformed 2D lane lines are fitted by straight lines at $y' < y_{close}$. Therefore, the correct camera pitch θ is the angle between the 3D ground and the BEV flat ground under the assumption of camera pitch = 0. θ can be calculated as follows:

$$\begin{cases} \frac{x'_1}{x_1} = \frac{x'_2}{x_2} = \frac{y'}{y} = \frac{h-z'}{h} \\ x'_1 = k_1 y'_1 + c_1 \\ x'_2 = k_2 y'_2 + c_2 \\ -\tan\theta = \frac{z'}{y'} \end{cases} \tag{7}$$

$$\Rightarrow \theta = \tan^{-1}\left(h\frac{k_2-k_1}{c_2-c_1}\right)$$

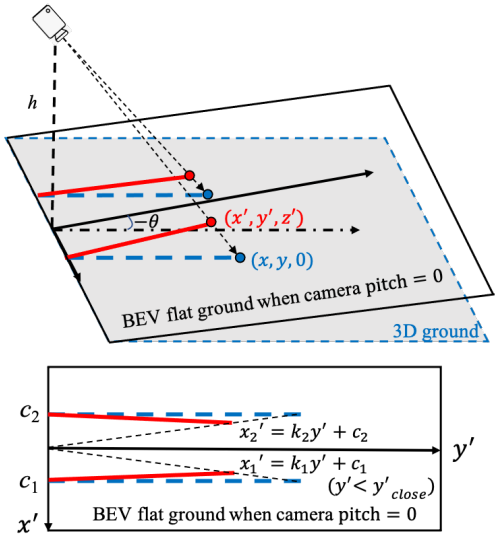

Fig. 4. **Camera pitch self-calibration.**

A calibration result is shown in Fig. 5. The front view image shows the automobile driving on flat ground. Before self-calibration, the lanes have a toe-in angle. After the calibration, the lane lines become parallel and the ground looks flat. On Apollo-Sim-3D, the average error between the ground truth and the calibrated camera pitch is 0.11°. When the camera pitch is predicted in the network, L1 loss is used to regress the camera pitch as L_{pitch}.

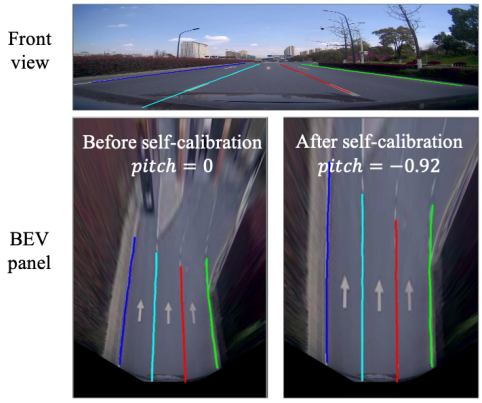

Fig. 5. **Camera pitch self-calibration result.**

D. Auxiliary Designs

Double-layer anchor with NMS. Based on the work of 3D-LaneNet, we design the double-layer anchors to improve the accuracy in the case where the lane lines are too close, as shown in Fig. 6. The first layer follows the 3D-LaneNet design [5], and the second layer has the same x steps as the first layer. When there are two close lanes in the anchor, $\hat{p}_i^A$ of the second layer will be set to 1. The anchors of the first layer predict the left line of the two close lane lines, while the second layer anchors predict the right. However, the constant width assumption in the weakly supervised setting is usually invalid if two lines are too close to each other, e.g. fork road or curb with a nearby lane line as in Fig.8. Therefore, in real implementation, we simply drop L_{width} and L_{height} loss terms described in section III-A and section III-B when the second-layer anchor is activated during model training.

Double-layer anchor may cause multiple anchors predicting the same lane line, therefore, we use a non-maximum suppression (NMS) method to solve this issue. For each anchor prediction X_i^A, the mean value of x_i^A distance between X_i^A and another neighbored anchor X_k^A is calculated as $\bar{d_{i,k}}$. If $\bar{d_{i,k}} < d_{thresh}$ and $p_i^A < p_k^A$, p_i^A will be set to 0.

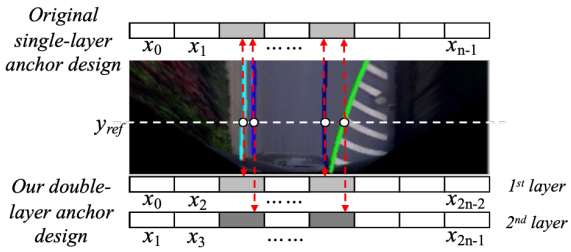

Fig. 6. **Single-layer anchor design [5] v.s. Our double-layer anchor design.**

Besides, we also employ pitch augmentation [13] and BEV segmentation branch [6] in order to further improve our model performance. Detail ablation study may be found in Section IV-C.

E. Loss Function

The loss function of our *WS-3D-Lane* L_{ws} consists of a purely supervised part on BEV flat ground L_{bev}, a weakly supervised part $L_{width}+L_{height}$, a BEV segmentation loss L_{seg} and a camera pitch regression loss L_{pitch}.

$$L_{ws} = L_{bev} + L_{width} + L_{height} + L_{seg} + L_{pitch} \tag{8}$$

The purely supervised part L_{bev} of the loss function is the same with Gen-LaneNet [4] except the difference between prediction and label along z-axis L_z:

$$\begin{aligned} L_{bev} = &- \sum_{i=0}^{N-1} \left(\hat{p}_i \log p_i + (1-\hat{p}_i)\left(\log(1-p_i)\right)\right) \\ &+ \sum_{i=0}^{N-1}\sum_{j=0}^{Y-1} \hat{p}_i \cdot \left(\left\| \hat{v}_{i,j} \cdot \left(x'_{i,j} - \hat{x}'_{i,j}\right) \right\|_1 \right) \\ &+ \sum_{i=0}^{N-1}\sum_{j=0}^{Y-1} \hat{p}_i \cdot \left\| v_{i,j} - \hat{v}_{i,j} \right\|_1 \end{aligned} \tag{9}$$

where N is the number of anchors along x-axis. For comparison, the loss function of our *WS-3D-Lane* in purely supervised setting L_{sup} use L_z to train the lane height.

$$L_{sup} = L_{bev} + L_z + L_{seg} + L_{pitch} \tag{10}$$

$$L_z = \sum_{i=0}^{N-1}\sum_{j=0}^{Y-1} \hat{p}_i \cdot \left(\left\| \hat{v}_{i,j} \cdot (z_{i,j} - \hat{z}_{i,j}) \right\|_1 \right) \tag{11}$$

IV. Experiments

A. Experiments Setup

Datasets. We conduct the experiments on both Apollo-Sim-3D and ONCE-3DLanes. Apollo-Sim-3D is a published synthetic 3D lane dataset [4] and is widely used in 3D lane detection tasks [4], [6], [13]. It provides 10.5K images with 3D lane labels and corresponding camera pitch. ONCE-3DLanes collect 211K images in real-world by LIDAR and auto-labeled 2D images, but the camera extrinsics is not recorded.

Comparison. For a fair comparison with previous works, the camera extrinsics is assumed to be known on Apollo-Sim-3D and the evaluation method is following the previous literature [4]. On ONCE-3DLanes, the camera pitch is calculated per frame by our self-calibration method with $y_{close} = 10m$ and the predicted lane lines are turned to the camera coordinate system to follow the same evaluation in [7]. To show the upper limit of our work, *WS-3D-Lane* in the purely supervised setting, called *WS-3D-Lane_sup*, is also compared in the experiment. The implementation details of *WS-3D-Lane_sup* is the same with *WS-3D-Lane* except the loss function. *WS-3D-Lane* use equation (8) as loss function while *WS-3D-Lane_sup* use equation (10).

Implementation details. Our experiments are carried out on our proposed *WS-3D-Lane* benchmark. We use 3D-LaneNet [5] with anchor representation from [4] as basic network. On the two datasets, the input shape is 360×480, and the IPM feature map shape is 208×128. For model training, we use the Adam optimizer [30] with the initial learning rate of $1e-3$ and the weight decay of $2e-4$. The learning rate is linearly annealed to $1e-7$ during training. We train the network with the batch size of 16 on one Nvidia Tesla V100 GPU. The training epochs are set to 100 on Apollo-Sim-3D and 30 on ONCE-3DLanes. For data augmentation, we adopt the random camera pitch noise with range $[-1°, 1°]$. For anchor representation, we adopt the BEV region with x-range $[-10, 10]m$ and use $Y_{ref} = 5m$ to associate each anchor with its closet lane. The y-range is set to $[0, 100]m$ on Apollo-Sim-3D and $[0, 50]m$ on ONCE-3DLanes. We design the y-reference points as {0, 2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100} m on Apollo-Sim-3D and $1m$ per step on ONCE-3DLanes. In the post process, we adopt NMS method with $d_{thresh} = 0.05m$.

B. Results and Qualitative Comparison

Results. The evaluation results on Apollo-Sim-3D are shown in Table I. Compared to 3D-LaneNet [5], 3D-

TABLE I

EVALUATION RESULTS ON APOLLO-SIM-3D

Method	F-Score(%)↑	AP(%)↑	x error near (m) ↓	x error far (m)↓	z error near (m)↓	z error far (m)↓
3D-LaneNet [5]	86.4	89.3	0.068	0.477	0.015	0.202
Gen-lanenet [4]	88.1	90.1	0.061	0.496	0.012	0.214
3D-LaneNet# [4]	90.0	92.0	-	-	-	-
3D-LaneNet*	89.8	91.9	0.054	0.408	0.010	0.243
Gen-lanenet-reconstruct [13]	91.9	93.8	0.049	0.387	0.008	**0.213**
Persformer [6]	92.9	-	0.054	0.356	0.010	0.234
WS-3D-Lane(ours)	92.3	94.6	0.060	0.373	0.023	0.233
WS-3D-Lane_sup(ours)	**93.5**	**95.7**	**0.027**	**0.321**	**0.006**	0.215

'near' means $y < 40m$ and 'far' means $y >= 40m$. 3D-LaneNet# is 3D-LaneNet with anchor representation in Gen-Lanenet [4]. 3D-LaneNet* is our reproduced 3D-LaneNet#. **WS-3D-Lane** is our weakly supervised 3D-Lane network. **WS-3D-Lane_sup** is WS-3D-Lane under purely supervised setting.

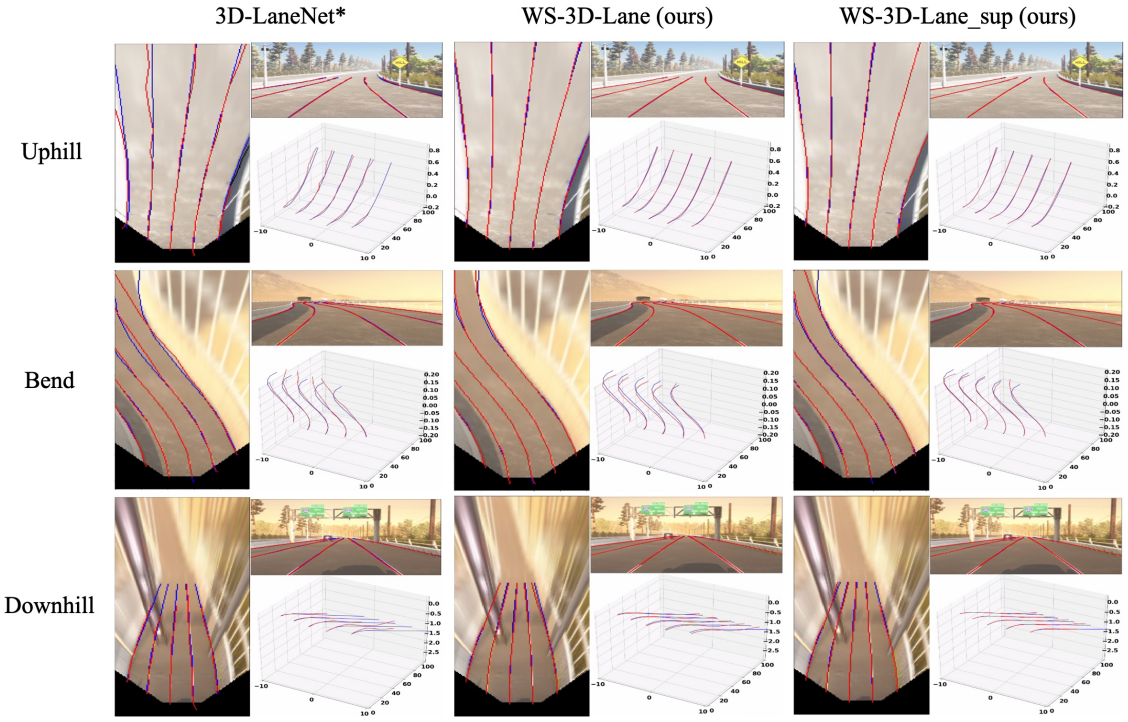

Fig. 7. **Qualitative comparison on Apollo-Sim-3D.** Blue lines: ground truth, Red lines: prediction. Each visualized result consists of the front view image (right top), BEV panel (left), and 3D lanes (right down).

LaneNet# [4], and our reproduction 3D-LaneNet*, our *WS-3D-Lane* consistently shows much better F-score and AP using only weakly supervised setting. Under purely supervised setting, our *WS-3D-Lane_sup* performs 93.5% on F-score outperforming previous state-of-the-art PersFormer's [6] 92.9%, and achieves state-of-the-art results on other metrics compared to previous works. As some scenes are inconsistent with our assumptions, *WS-3D-Lane* underperforms *WS-3D-Lane_sup*, especially on z error. In Table II, we report experimental results on ONCE-3DLanes. Under only weakly supervised setting, our *WS-3D-Lane* is able to archive similar results to previous supervised state-of-the-art method PersFormer [6]. By using supervised setting, our *WS-3D-Lane_sup* outperforms PersFormer [6] on all metrics by a large margin.

Qualitative comparison. Figure 7 shows the qualitative comparison of reproduced 3D-LaneNet*, our *WS-3D-Lane* and our *WS-3D-Lane_sup* on Apollo-Sim-3D. The predicted 3D lanes are projected onto the front view image and BEV panel. In the scenes of uphill and bend, 3D-LaneNet* predicts incorrect location along x-axis on far lane lines, but *WS-3D-Lane* and *WS-3D-Lane_sup* predict it more correctly. In the scene of downhill, the predicted lane lines of 3D-LaneNet* are much shorter than the ground truth. It is caused by the prediction error in the visibility. *WS-3D-Lane* and *WS-3D-Lane_sup* perform much better on this case with correct visibility prediction. Figure 8 shows the qualitative comparison of ONCE-3DLanes. In the scenes of fork and curb with nearby lane lines, 3D-LaneNet* is not able to predict two close lane lines at the same time, and the model tends to predict the center line of the two lane lines. After using the double-layer anchor, our *WS-3D-Lane* predicts two lane lines correctly. However, compared to *WS-3D-Lane_sup*, *WS-3D-Lane* get a little larger errors on the height in all the scenes, which is reasonable because of the lack of direct supervision on lane height.

TABLE II

EVALUATION RESULTS ON ONCE-3DLANES

Method	F1 (%)↑	Precision (%)↑	Recall (%)↑	CD error (m)↓
3D-LaneNet [7]	44.73	61.46	35.16	0.127
SALAD [7]	64.07	75.90	55.42	0.098
Persformer [6]	74.33	80.30	69.18	0.074
3D-LaneNet*	70.07	80.55	61.79	0.074
WS-3D-Lane(ours)	74.56	81.31	68.85	0.072
WS-3D-Lane_sup(ours)	**77.02**	**84.51**	**70.75**	**0.058**

CD error: Chamfer distance error. **WS-3D-Lane** is our weakly supervised 3D-Lane network. **WS-3D-Lane_sup** is our WS-3D-Lane under purely supervised setting. 3D-LaneNet* is our reproduced 3D-LaneNet# [4] result with our self-calibrated camera pitch and training hyperparameters for fairer comparison.

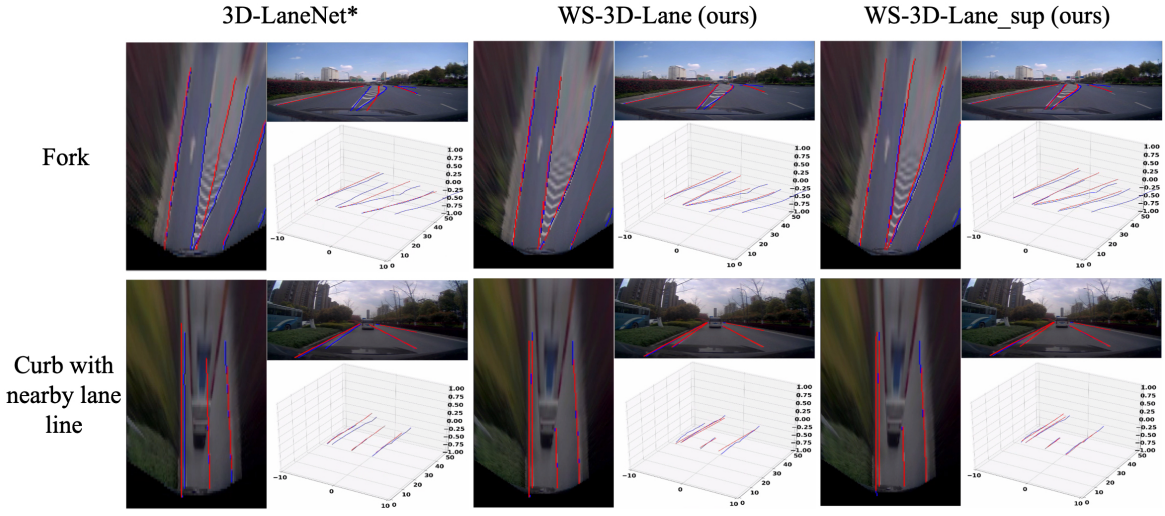

Fig. 8. **Qualitative comparison on ONCE-3DLanes.** Blue lines: ground truth, Red lines: prediction.

C. Ablation Study

We conduct the ablation study on ONCE-3DLanes to evaluate the impact of our designs and the results are shown in Table III. As a result, high quality pitch angles are critical to 3D lane detection. With our per-frame self-calibrated pitch angle, our baseline model significantly outperforms 3D-LaneNet [7] which uses only fixed camera extrinsics provided by ONCE [8]. Pitch augmentation contributes extra performance boost by augmenting the diversity of pitch angle. Double-layer anchor also improves the metric by equipping model with the ability to predict two close lane lines and the performance gain becomes even larger after post-processed by NMS method. BEV segmentation branch shows marginally improvements in supervised setting but brings side effects under weakly supervised setting, which might be caused by the errors of auto-labeled 2D lanes in the dataset.

D. Limitation and Future Work

Our weakly supervised method heavily relies on the assumptions of constant lane width and equal height on adjacent lane lines. Therefore, it is hard to handle several cases like road with only single lane line, dynamic lane width, and exit/entrance ramps. The model performance and generalization on these corner cases are not well studied in this work and can be a meaningful future research direction.

TABLE III

RESULTS OF ABLATION STUDY ON ONCE-3DLANES

	Supervised		Weakly Supervised	
Method	F1 ↑	CD error ↓	F1 ↑	CD error ↓
3D-LaneNet [7]	44.73	0.127	-	-
Baseline(3D-LaneNet*)	70.07	0.059	68.73	0.074
+PA	74.27	0.059	72.23	0.072
+PA+DA	75.27	**0.056**	73.83	**0.071**
+PA+DA+NMS	76.13	0.057	**74.84**	**0.071**
+PA+DA+NMS+BS	**77.02**	0.058	74.56	0.072

Baseline(3D-laneNet*): our reproduced 3D-LaneNet# [4] our self-calibrated pitch. PA: pitch augmentation [13]. DA: double-layer anchor. NMS: non-maximum suppression method. BS: BEV segmentation branch [6].

V. CONCLUSIONS

In this paper, we present a weakly supervised 3D lane detection method that allows the model to be trained with only 2D labels by using the assumptions of constant lane width and equal height on adjacent lane lines. A self-calibration method is proposed to improve camera pitch quality of data. Several auxiliary designs are applied in order to improve the overall model performance. We conduct comprehensive experiments to validate the effectiveness of our method. We believe a weakly supervised 3D lane detection approach by using only 2D annotations is valuable in both academic research and real industrial production, and we hope our *WS-3D-Lane* can stimulate more related research in the future.

REFERENCES

[1] H. Abualsaud, S. Liu, D. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, "Laneaf: Robust multi-lane detection with affinity fields," *IEEE Robotics and Automation Letters*, vol. 6, pp. 7477–7484, 2021.

[2] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1013–1021, 2019.

[3] L. Liu, X. Chen, S. Zhu, and P. Tan, "Condlanenet: a top-to-down lane detection framework based on conditional convolution," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3753–3762, 2021.

[4] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, and T. E. Choe, "Gen-lanenet: A generalized and scalable approach for 3d lane detection," in *ECCV*, 2020.

[5] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3d-lanenet: End-to-end 3d multiple lane detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2921–2930, 2019.

[6] L. F. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, and J. Yan, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," *ArXiv*, vol. abs/2203.11089, 2022.

[7] F. Yan, M.-J. Nie, X. Cai, J. Han, H. Xu, Z. Yang, C. Ye, Y. Fu, M. B. Mi, and L. Zhang, "Once-3dlanes: Building monocular 3d lane detection," *ArXiv*, vol. abs/2205.00301, 2022.

[8] J. Mao, M. Niu, C. Jiang, H. Liang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: Once dataset," *ArXiv*, vol. abs/2106.11037, 2021.

[9] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI*, 2017.

[10] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending," in *ECCV*, 2020.

[11] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1965–1973, 2017.

[12] B. Roberts, S. Kaltwang, S. Samangooei, M. Pender-Bare, K. Tertikas, and J. Redford, "A dataset for lane instance segmentation in urban environments," *ArXiv*, vol. abs/1807.01347, 2018.

[13] C. Li, J. Shi, Y. Wang, and G. Cheng, "Reconstruct from top view: A 3d lane detection approach based on geometry structure prior," *ArXiv*, vol. abs/2206.10098, 2022.

[14] R. Liu, D. Chen, T. Liu, Z. Xiong, and Z. Yuan, "Learning to predict 3d lane shape and camera pose from a single image via geometry constraints," *ArXiv*, vol. abs/2112.15351, 2022.

[15] J. Jung and S.-H. Bae, "Real-time road lane detection in urban areas using lidar data," *Electronics*, 2018.

[16] M. Thuy and F. P. León, "Lane detection and tracking based on lidar data," *Metrology and Measurement Systems*, vol. 17, pp. 311–321, 2010.

[17] A. Y. Hata and D. F. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 584–589, 2014.

[18] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," *2008 IEEE Intelligent Vehicles Symposium*, pp. 1137–1142, 2008.

[19] N. Efrat, M. Bluvstein, S. Oron, D. Levi, N. Garnett, and B. E. Shlomo, "3d-lanenet+: Anchor free lane detection using a semi-local representation," *ArXiv*, vol. abs/2011.01535, 2020.

[20] Y. Jin, X. Ren, F. Chen, and W. Zhang, "Robust monocular 3d lane detection with dual attention," in *ICIP*, 2021.

[21] N. Efrat, M. Bluvstein, N. Garnett, D. Levi, S. Oron, and B. E. Shlomo, "Semi-local 3d lane detection and uncertainty estimation," *ArXiv*, vol. abs/2003.05257, 2020.

[22] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun, "Deep multi-sensor lane detection," in *IROS*, 2018.

[23] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar-camera fusion for road detection using fully convolutional neural networks," *Robotics Auton. Syst.*, vol. 111, pp. 125–131, 2019.

[24] X. N. Zhang, Z. Li, X. Gao, D. Jin, and J. Li, "Channel attention in lidar-camera fusion for lane line segmentation," *Pattern Recognit.*, vol. 118, p. 108020, 2021.

[25] Z. Qin, J. Wang, and Y. Lu, "Weakly supervised 3d object detection from point clouds," *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[26] Q. Meng, W. Wang, T. Zhou, J. Shen, L. V. Gool, and D. Dai, "Weakly supervised 3d object detection from lidar point cloud," in *ECCV*, 2020.

[27] Z. Liu, X. Qi, and C.-W. Fu, "One thing one click: A self-training approach for weakly supervised 3d semantic segmentation," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1726–1736, 2021.

[28] Y. Wei, S.-C. Su, J. Lu, and J. Zhou, "Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4348–4354, 2021.

[29] X. Xu, Y. Wang, Y. Zheng, Y. Rao, J. Lu, and J. Zhou, "Back to reality: Weakly-supervised 3d object detection with shape-guided label enhancement," *ArXiv*, vol. abs/2203.05238, 2022.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.