D^{3}L: Curvature-Constrained Denoising Diffusion Model for 3D Lane Detection

Anonymous Author(s)

Abstract

Monocular 3D lane detection is a challenging task for autonomous driving systems. Recent advances primarily focus on one-step methods for lane detection based on front-view features, which show promising results on straight lanes. However, curved lanes are difficult to handle with one-step prediction, which performs prediction in a single leap without gradual refinement. To address this issue, we propose a novel **D**enoising **D**iffusion Model for 3**D** **L**ane Detection framework (**D^{3}L**). The main idea is to leverage the progressive generation capability of the diffusion model to generate accurate 3D curved lanes, and ensuring lane continuity through curvature constraints. The framework includes three creative components: coarse-to-fine denoiser (CFD), curvature-constrained loss (CCL) and multi-sampling aggregation strategy (MSAS). In CFD, both lane-level and point-level transformer blocks are integrated to accurately denoise 3D lanes, which effectively captures both global and local features. CCL is designed to reduce deviations in lane curvature, resulting in smoother lane continuity. This loss enhances both the accuracy and geometric consistency of lane detection, especially in complex curved scenes. MSAS is proposed to select the optimal lane point-by-point from multiple candidates, thus robustness of the lane prediction is significantly improved. Extensive experiments on two popular 3D lane detection benchmarks demonstrate that our D^{3}L outperforms the state-of-the-art methods.

CCS Concepts

• **Computing methodologies** → **Scene understanding**.

Keywords

3D lane detection, denoising diffusion model, curvature constraint.

ACM Reference Format:

Anonymous Author(s). 2025. D^{3}L: Curvature-Constrained Denoising Diffusion Model for 3D Lane Detection. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

3D lane detection is a critical component of autonomous driving systems. It aims to predict the 3D positions of lanes from front-view (FV) images captured by vision cameras. Accurate and robust perception of 3D lanes is vital for various downstream tasks, including trajectory planning [26] and high-definition map construction [18]. In recent years, the application of deep learning techniques has led to significant advancements in the field of 3D lane detection.

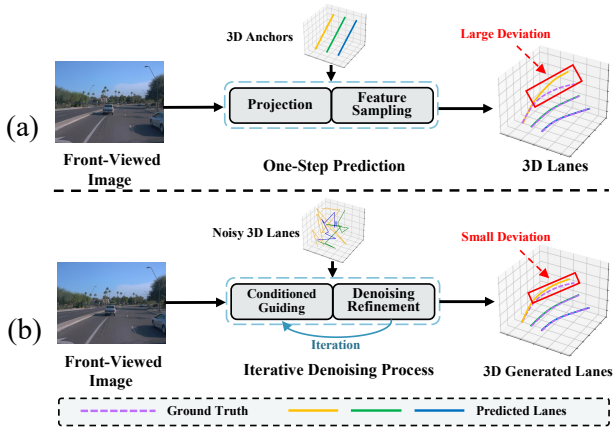


Figure 1: Comparison of previous methods and our proposed method. (a) Previous methods employ 3D anchors projected onto front-view images to extract 2D features, then apply a one-step prediction, which results in large deviations in curved scenes. (b) Our D^{3}L initializes noisy 3D lanes and refines them through an iterative denoising process, resulting in small deviations and improved accuracy.

Current deep learning-based 3D lane detection models are broadly classified into two categories: BEV-based methods and FV-based methods. The former detect 3D lanes by transforming FV features to bird-eye-view (BEV) space using the inverse perspective mapping (IPM), which is effective on flat roads [15][2]. However, IPM's reliance on the flat-ground assumption causes misalignment between FV and BEV spaces, especially in uphill and downhill scenes. The latter directly predict 3D lanes based on the FV images, as shown in Figure 1(a) [11][25]. Specifically, they first define anchors in the 3D space. These anchors are then projected into corresponding 2D points in the FV space using camera parameters. Finally, the 3D lane is predicted in one step based on 2D features obtained by bilinear sampling. However, these methods still encounter difficulties in handling curved lanes, often resulting in large deviations. Through a detailed analysis of the basic mechanism, this limitation can be attributed to two main factors. The one is that one-step prediction produces lane geometry in a single step without gradual refinement. The other is that point-by-point prediction struggles to maintain smooth geometry along curves, leading to further misalignment.

Recently, the Diffusion Model (DM) [10] has gained significant attention for the enhancement of performance through iterative denoising. Specifically, DM has exhibited its superiority in various

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

3D perception tasks, such as 3D human pose estimation [19] and 3D hand pose estimation [4]. Inspired by the success of DM, our work adopts this capability of progressive generation for 3D lane detection. Additionally, a curvature constraint is applied to maintain geometric consistency among lane points, ensuring smooth and realistic lane shapes. Considering these two aspects, our framework integrates diffusion models and curvature constraints to optimize 3D lane generation, as illustrated in Figure 1(b). This process progressively refines noisy 3D lane predictions through iterative denoising and curvature constraints for accurate lane predictions. The framework is general and capable of handling both curved and straight lanes, where straight lanes are a special case with zero curvature. To the best of our knowledge, our work is the first attempt to deploy diffusion models in the 3D lane detection task.

In this paper, we propose a novel framework **D^{3}L**, termed Curvature-Constrained **D**enoising **D**iffusion Model for 3**D** **L**ane Detection. The whole framework includes three creative components: coarse-to-fine denoiser (CFD), curvature-constrained loss (CCL) and multi-sampling aggregation strategy (MSAS). Specifically, in CFD module, noisy 3D lanes are first projected into 2D images for feature sampling, then fused with conditional features to effectively bridge the 2D-3D gap. These fused features are subsequently refined through both lane-level and point-level transformer blocks, enhancing both global structure and local details of the final 3D lane predictions. The CCL is introduced to further enforce curvature consistency in the predicted lanes. It maintains smooth curvature continuity at key points, reducing sharp variations. This ensures that the predicted lanes closely match the true 3D geometry, enhancing overall accuracy. To further improve robustness against prediction noise, MSAS employs a lane point confidence loss to measure proximity between predicted and ground truth points. It then aggregates optimal lane points from multiple candidates, reducing reliance on single predictions and enhancing stability.

Our contributions are summarized as follows:

- A novel diffusion-based framework D^{3}L is proposed for 3D lane detection. By integrating a coarse-to-fine denoiser with both lane-level and point-level transformer blocks, D^{3}L progressively refines predictions and significantly enhances the accuracy of curved lane detection.
- A novel curvature-constrained loss is introduced to regulate the lane's overall shape by enforcing curvature consistency along the lane. This loss minimizes sharp curvature changes, further improving the accuracy of lane predictions.
- A multi-sample aggregation strategy is proposed to select optimal lanes from multiple candidates, reducing randomness and improving prediction stability. Extensive experiments on two datasets show that D^{3}L surpasses the state-of-the-art methods.

2 Related Work

2.1 3D Lane Detection

3D lane detection aims to obtain accurate 3D positions of lanes in real-world scenes. Current methods can be divided into two main types: BEV-based methods [6][15][2], which transform FV images to BEV space, and FV-based methods [11][25], which directly predict lanes from FV images. Due to the good geometric properties of lanes in the BEV perspective, BEV-based methods [6][15][2] attempt to transform the FV image to the BEV space based on IPM. However, IPM's reliance on the flat-ground assumption can cause misalignment between BEV and 3D spaces in rough ground cases. To address this problem, FV-based methods predict 3D lanes directly from the FV image without IPM. Specifically, Anchor3DLane [11] defines anchors in 3D space and directly predicts 3D lane positions by sampling features projected onto 2D images. PVALane [25] further proposes prior-guided 3D anchors projected onto FV and BEV, aligning the two sampled features to predict 3D lanes. However, FV-based methods estimate 3D lane positions in one step by predicting offset values, which limits their flexibility and makes them less suited for accurately modeling curved lanes. In contrast, our approach leverages a diffusion model to progressively learn the distribution of 3D lane points, providing a more adaptable and precise solution for complex lane structures.

2.2 Diffusion Model

Diffusion models [10] are a family of deep generative models, also known as denoising diffusion probabilistic models (DDPMs). The diffusion model reconstructs the original data distribution from a perturbed one by injecting noise over multiple steps and then iteratively denoising to recover the data structure. Recently, they have achieved remarkable success in 3D perception tasks, such as 3D human pose estimation [5][19] and 3D hand pose estimation [4]. Existing diffusion-based approaches for human pose estimation regress 3D keypoint locations from 2D RGB images of the human body. Specifically, DiffPose [5] introduces a conditional heatmap representation of 2D joints to guide the reverse diffusion process, utilizing spatiotemporal features for improved joint localization. D3DP [19] proposes a multi-hypothesis aggregation with joint-wise reprojection to determine the best hypothesis from the diffusion model using the 2D prior. For 3D hand pose estimation, HandDiff [4] employs iterative denoising with joint-wise and local detail conditioning for precise 3D hand pose estimation from hand-shaped image-point clouds. As far as we know, there is currently no work that has applied the diffusion model to 3D lane detection. This paper introduces a novel framework D^{3}L, which explores the potential of diffusion models in 3D lane detection.

3 Method

3.1 Preliminaries

Problem Formulation. Given a front-viewed image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ as input, where H and W denote the height and width of the input image, 3D lane detection aims to predict the 3D position of lanes within it. Lanes are represented by a set of 3D points, denoted as $\mathbf{G} = \{\mathbf{L}_i | i \in 1, ..., N\}$, where N is the number of lanes in the image, and $\mathbf{L}_i$ denotes the i-th lane. Each lane $\mathbf{L}_i = (\mathbf{P}_i, \mathbf{C}_i)$ is composed of a set of points $\mathbf{P}_i = \{(x_i^j, y_i^j, z_i^j, vis_i^j)\}_{j=1}^M$, where M is a predetermined number of output points, and $\mathbf{C}_i$ represents the category of the i-th lane. For each point $\mathbf{P}_i$, the first three elements denote the location of $\mathbf{P}_i$ in 3D space and the last element denotes the visibility of $\mathbf{P}_i$.

Diffusion Model. Diffusion models are a class of latent variable models inspired by concepts from non-equilibrium thermodynamics [21][22]. They are primarily based on two core processes: 1) a

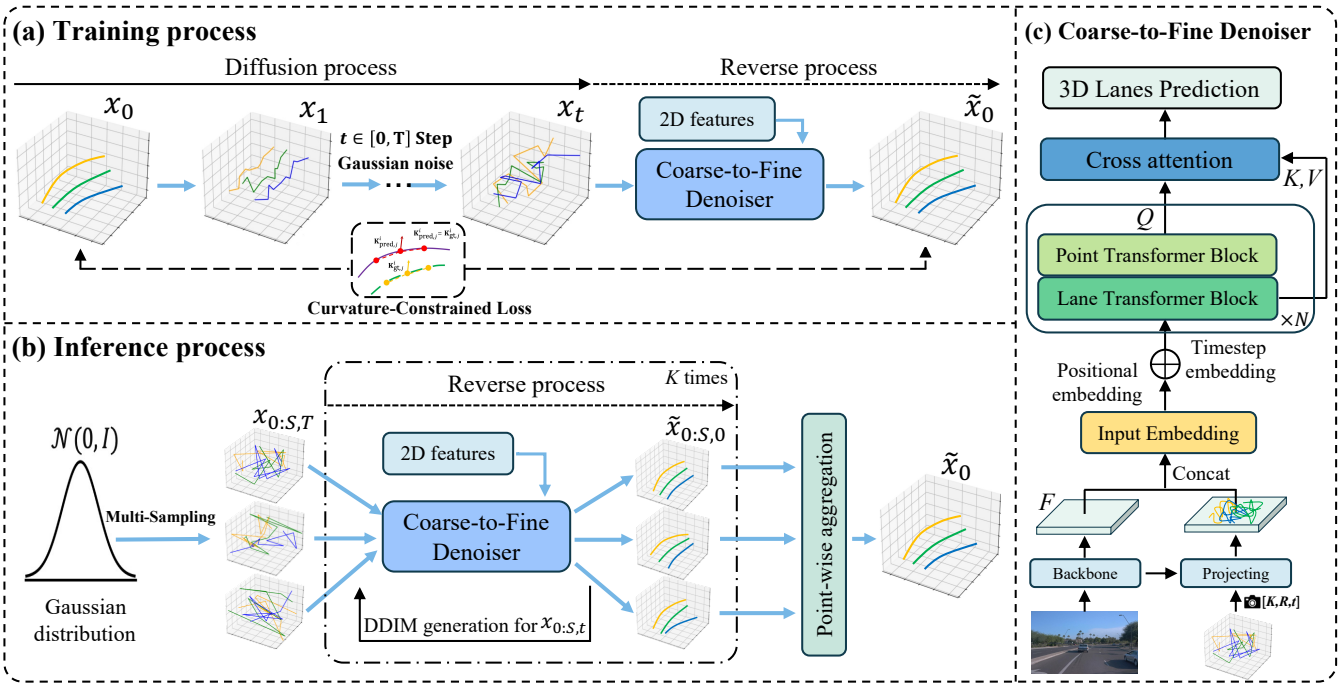


Figure 2: Illustration of our D^{3}L. (a) Training. Gaussian noise is added to the ground truth 3D lanes, resulting in noisy lanes x_t. x_t are then fed into the denoiser conditioned on 2D features to yield the final predictions. (b) Inference. S samples are drawn from a Gaussian distribution to initialize the 3D lanes, which are then used to produce noiseless 3D lanes. The reverse process is iterated K times, refining the results by feeding DDIM-generated 3D lanes with varying noise levels to the denoiser. Finally, the predicted lanes are aggregated point-wise to produce the optimal 3D lanes. (c) Conditioned on front-view features, the Coarse-to-Fine denoiser refines noisy lanes using both lane-level and point-level transformer blocks to generate final 3D lanes.

forward process that gradually adds Gaussian noise to sample data, and 2) a reverse process that learns to invert the forward diffusion. Specifically, the forward process is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{1}$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, 1), \tag{2}$$

where $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t}(1-\beta_s)$ and β_s denotes the noise variance schedule [10]. As shown in Eq. 1 and Eq. 2, the process involves adding noise to the initial data sample $\mathbf{x}_0$, transforming it into a noisy sample $\mathbf{x}_t$ at a specific sampling step t from the set $\{0, 1, ..., T\}$. In the training phase, the neural network $f_\theta(\mathbf{x}_t, t)$ is optimized to predict the noise ϵ by minimizing the L_2 loss, which can be formulated as:

$$\mathcal{L}_{\text{train}} = \|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2 . \tag{3}$$

During the inference phase, the original data $\mathbf{x}_0$ is progressively recovered from the noisy sample $\mathbf{x}_T$ using the trained denoising function f_θ through a sequential and iterative refinement process: $\mathbf{x}_T \rightarrow \mathbf{x}_{T-\Delta} \rightarrow \cdots \rightarrow \mathbf{x}_0$ [20][10].

3.2 Overview of D^{3}L

Our framework is illustrated as Figure 2. During the training process (Figure 2(a)), t-step Gaussian noise is added to the ground truth 3D lanes $\boldsymbol{x}_0$, resulting in the noisy lanes $\boldsymbol{x}_t$. $\boldsymbol{x}_t$ is then fed to the CFD (Figure 2(c)) conditioned on 2D features to yield the final prediction $\tilde{\boldsymbol{x}}_0$. In the inference process (Figure 2(b)), S samples are drawn from a Gaussian distribution to initialize 3D lanes $\boldsymbol{x}_{0:S,T}$, which are utilized to yield the noiseless 3D lane predictions $\tilde{\boldsymbol{x}}_{0:S,0}$. Besides, the above reverse process is iterated K times to refine the final results by sending DDIM-generated 3D lanes $\boldsymbol{x}_{0:S,t}$ with different levels of noise to the CFD. Finally, $\tilde{\boldsymbol{x}}_{0:S,0}$ are aggregated point-wise to produce the optimal 3D lanes $\tilde{\boldsymbol{x}}_0$.

3.3 Coarse-to-Fine Denoiser

In 3D lane detection, fusing 3D coordinates with 2D image features can cause misalignment. Directly merging these disparate data types can result in inaccurate predictions due to the inherent misalignment. To solve this, we project noisy 3D lane points onto a 2D feature map using camera parameters, as shown in Figure 2(c). Specifically, we begin by projecting the noisy 3D lane coordinates $\mathbf{P}_{3D}$ onto the 2D feature map $\mathbf{F}$. Each 3D point $\mathbf{p}_i^j = (x_i^j, y_i^j, z_i^j)$ is transformed using the camera's intrinsic matrix $\mathbf{K}$ and extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$. The combined projection and normalization process is defined as follows:

$$\begin{bmatrix} \hat{u}_i^j \\ \hat{v}_i^j \\ 1 \end{bmatrix} = \frac{1}{z_i^j} \cdot \mathbf{K} \left(\mathbf{R} \cdot \begin{bmatrix} x_i^j \\ y_i^j \\ z_i^j \end{bmatrix} + \mathbf{t} \right), \tag{4}$$

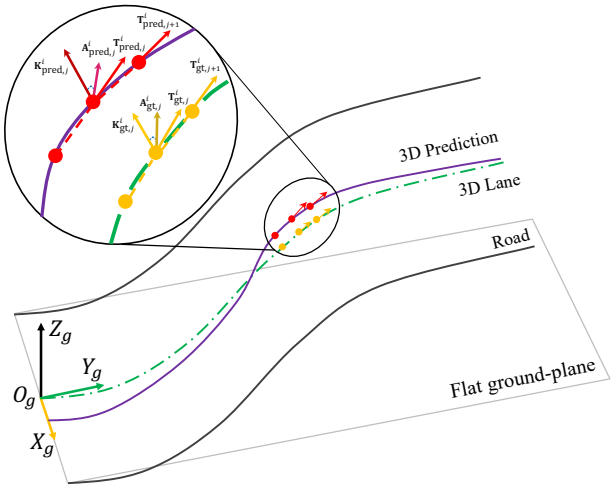


Figure 3: Illustration of Curvature-Constrained Loss in ground coordinate system. Red represents predicted lane points, and orange represents ground truth lane points.

where $\hat{u}_i^j$ and $\hat{v}_i^j$ are the normalized 2D coordinates mapped onto the feature map dimensions. These coordinates are used to sample features from $\mathbf{F}$ via bilinear interpolation. The sampled features are concatenated with the original 2D features to form the input embedding. To enhance the input embedding $\mathbf{E}_{\text{input}}$, positional embeddings $\mathbf{E}_{\text{pos}}$ and timestep embeddings $\mathbf{E}_{\text{time}}$ are added to incorporate spatial and temporal context:

$$\mathbf{E}_{\text{enhanced}} = \mathbf{E}_{\text{input}} + \mathbf{E}_{\text{pos}} + \mathbf{E}_{\text{time}}. \tag{5}$$

The enhanced embedding is then passed through a series of transformer blocks [24] designed to refine the lane-level and point-level features. The first stage involves the Lane Transformer Block, where the self-attention mechanism is used to capture relationships across different lanes. For each lane, the attention mechanism operates as follows:

$$\mathbf{A}_{\text{lane}}^i = \text{softmax}\left(\frac{\mathbf{Q}_{\text{lane}}^i \cdot (\mathbf{K}_{\text{lane}}^i)^\top}{\sqrt{d_k}} + \mathbf{M}_L\right) \cdot \mathbf{V}_{\text{lane}}^i, \tag{6}$$

where $\mathbf{Q}_{\text{lane}}^i$, $\mathbf{K}_{\text{lane}}^i$, and $\mathbf{V}_{\text{lane}}^i$ are the query, key, and value matrices corresponding to the i-th lane, and $\mathbf{M}_L$ is the lane mask matrix, ensuring lane-specific attention. This allows the model to capture the global context across lanes. Subsequently, the refined lane embeddings are processed through the Point Transformer Block, where self-attention is applied at the level of individual points within each lane:

$$\mathbf{A}_{\text{point}}^{i,j} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{point}}^{i,j} \cdot (\mathbf{K}_{\text{point}}^{i,j})^\top}{\sqrt{d_k}} + \mathbf{M}_P\right) \cdot \mathbf{V}_{\text{point}}^{i,j}, \tag{7}$$

where $\mathbf{Q}_{\text{point}}^{i,j}$, $\mathbf{K}_{\text{point}}^{i,j}$, and $\mathbf{V}_{\text{point}}^{i,j}$ correspond to the query, key, and value matrices for the j-th point on the i-th lane, and $\mathbf{M}_P$ is the point mask ensuring point sequence integrity within each lane.

After applying N iterations of the Lane and Point Transformer Blocks, we introduce a Cross-Attention mechanism to integrate the refined lane-level and point-level features. The cross-attention operates by using point-level features as queries and lane-level features as keys and values:

$$\mathbf{Z}_{\text{final}}^{i,j} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{point}}^{i,j} \cdot (\mathbf{K}_{\text{lane}}^i)^\top}{\sqrt{d_k}}\right) \cdot \mathbf{V}_{\text{lane}}^i. \tag{8}$$

Finally, the output $\mathbf{Z}_{\text{final}}$ from the cross-attention mechanism is then passed through a multi-layer perceptron (MLP) to predict the denoised 3D lane coordinates:

$$\mathbf{P}_{3D}^{\text{pred}} = \text{MLP}(\mathbf{Z}_{\text{final}}). \tag{9}$$

3.4 Curvature-Constrained Loss

Previous methods primarily focus on minimizing the Euclidean distance between predicted and ground truth lane points. However, they often overlook the structural continuity and smoothness inherent in lane curves. This limitation becomes particularly pronounced in 3D space, where perspective distortions and road curvature cause lane shapes to vary significantly. To address this issue, we propose a Curvature-Constrained Loss (CCL) , as shown in Figure 3. This loss function is designed to enforce geometric consistency between predicted and ground truth lanes by considering both the shape and continuity of the lanes.

Shape Representation. The shape of a lane in 3D space is inherently defined by the spatial relationship among its constituent points. To capture this relationship, we use the concept of curvature vector, which not only describes the bending of the lane but also encapsulates the directionality of this bending in the 3D space. For each lane point $P_j^i = \left(x_j^i, y_j^i, z_j^i\right)$, we compute the tangent vector $\mathbf{T}_j^i$ and the acceleration vector $\mathbf{A}_j^i$ based on its adjacent points:

$$\mathbf{T}_j^i = \left(\frac{x_{j+1}^i - x_{j-1}^i}{2}, \frac{y_{j+1}^i - y_{j-1}^i}{2}, \frac{z_{j+1}^i - z_{j-1}^i}{2}\right), \tag{10}$$

$$\mathbf{A}_j^i = \begin{pmatrix} x_{j+1}^i - 2x_j^i + x_{j-1}^i \\ y_{j+1}^i - 2y_j^i + y_{j-1}^i \\ z_{j+1}^i - 2z_j^i + z_{j-1}^i \end{pmatrix}. \tag{11}$$

Using these vectors, we derive the curvature vector $\mathbf{K}_j^i$ as follows:

$$\mathbf{K}_j^i = \frac{\mathbf{T}_j^i \times \mathbf{A}_j^i}{|\mathbf{T}_j^i|^3}. \tag{12}$$

This curvature vector K_j^i describes the lane's bending and direction at P_j^i in 3D space.

Loss Calculation. CCL is defined to enforce consistency between the predicted and ground truth curvature vectors for all visible points along the lane. Specifically, for each lane, we compute the difference between the predicted curvature vector $\mathbf{K}_{\text{pred},j}^i$ and the ground truth curvature vector $\mathbf{K}_{\text{gt},j}^i$. CCL $\mathcal{L}_{shape}$ is defined as:

$$\mathcal{L}_{shape} = \frac{1}{\sum_{i=1}^N \sum_{j=1}^M vis_j^i} \sum_{i=1}^N \sum_{j=1}^M vis_j^i \|\mathbf{K}_{\text{pred},j}^i - \mathbf{K}_{\text{gt},j}^i\|^2, \tag{13}$$

where vis_j^i is the visibility indicator for point P_j^i, ensuring that only visible points contribute to the loss calculation. The overall loss is the average of the squared differences in curvature vectors, weighted by the visibility of each point.

Table 1: Comparison with state-of-the-art methods on OpenLane validation set. F1 score is presented for each scene.

Method	All	Up & Down	Curve	Extreme Weather	Night	Intersection	Merge & Split
3D-LaneNet [6]	44.1	40.8	46.5	47.5	41.5	32.1	41.7
Gen-LaneNet [7]	32.3	25.4	33.5	28.1	18.7	21.4	31.0
PersFormer [2]	50.5	42.4	55.6	48.6	46.6	40.0	50.7
CurveFormer [1]	50.5	45.2	56.6	49.7	49.1	42.9	45.4
Anchor3DLane [11]	53.1	45.5	56.2	51.9	47.2	44.2	50.5
LaneCPP [17]	60.3	53.6	64.4	56.7	54.9	52.0	58.7
PVALane-Res18 [25]	61.2	52.6	65.7	59.5	56.5	52.2	58.7
LATR-Lite [16]	61.5	55.2	67.9	57.6	55.1	52.1	60.3
LATR [16]	61.9	55.2	68.2	57.1	55.4	52.3	**61.5**
D^{3}L (Ours)	**62.4**	**56.5**	**68.5**	**59.8**	**56.6**	**52.7**	60.9

Table 2: Comparison with state-of-the-art methods on OpenLane validation set. "Cate Acc" means category accuracy.

Method	F1(%)↑	Cate Acc(%)↑	x err/C(m)↓	x err/F(m)↓	z err/C(m)↓	z err/F(m)↓
3D-LaneNet [6]	44.1	-	0.479	0.572	0.367	0.443
Gen-LaneNet [7]	32.3	-	0.591	0.684	0.411	0.521
PersFormer [2]	50.5	92.3	0.485	0.553	0.364	0.431
CurveFormer [1]	50.5	-	0.340	0.772	0.207	0.651
Anchor3DLane [11]	53.1	90.0	0.300	0.311	0.103	0.139
LaneCPP [17]	60.3	-	0.264	0.310	0.077	0.117
PVALane-Res18 [25]	61.2	**93.0**	0.249	0.263	0.094	0.122
LATR-Lite [16]	61.5	91.9	0.225	0.249	0.073	0.106
LATR [16]	61.9	92.0	0.219	0.259	0.075	0.104
D^{3}L (Ours)	**62.4**	92.2	**0.216**	**0.245**	**0.070**	**0.099**

3.5 Multi-Sampling Aggregation Strategy

In traditional 3D lane detection, a single prediction from the FV image often leads to lane deviation due to inherent uncertainty. To address this issue, we introduce a Multi-Sampling Aggregation Strategy (MSAS), as shown in Figure 2(b). The idea is to improve robustness by considering multiple hypotheses rather than relying on a single prediction, reducing inaccuracies and deviations. In inference, we perform multiple samplings from a Gaussian distribution to generate initial noisy lanes $\boldsymbol{x}_{0:S,T}$. Let S denote the number of samples drawn. For each sample, the noisy input is passed through the CFD conditioned on 2D image features, producing S different sets of predicted lanes $\tilde{\boldsymbol{x}}_{0:S,0}$.

To aggregate these multiple predictions into a single output, we propose a novel lane point confidence score $\hat{c}_{ij}$, which measures the proximity between the predicted lane points p_{ij}^{pred} and the ground truth p_{ij}^{GT}. The confidence score is calculated using the following equation:

$$\hat{c}_{ij} = \exp\left(-\frac{\|p_{ij}^{pred} - p_{ij}^{GT}\|^2}{\sigma^2}\right), \tag{14}$$

where $\|p_{ij}^{pred} - p_{ij}^{GT}\|$ denotes the Euclidean distance between the predicted and ground truth lane points. σ denotes a scaling parameter that controls the sensitivity of the confidence score. The lane point confidence loss $\mathcal{L}_{\text{conf}}$ is then defined as:

$$\mathcal{L}_{conf} = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\hat{c}_{ij} - \exp\left(-\frac{\|p_{ij}^{pred} - p_{ij}^{GT}\|^2}{\sigma^2}\right)\right)^2, \tag{15}$$

where N denotes the total number of lanes and M denotes the total number of points per lane. After calculating the confidence score for each point across S samples, we retain the point with the highest confidence by comparing the same point across different samples. This point-by-point selection method ensures that only the most reliable lane points are preserved, producing a refined and robust final aggregated 3D lane prediction $\tilde{\boldsymbol{x}}_0$. MSAS effectively mitigates the impact of noise and uncertainty inherent in individual predictions, enhancing the accuracy of the 3D lane representation.

3.6 Training and Inference Process

Training. We perform diffusion process that corrupts ground truth lane coordinates to noisy lane coordinates, and train the coarse-to-fine denoiser for lane denoising to reverse this process. The

Table 3: Comparison with other state-of-the-art methods on ApolloSim dataset with three different scenes. "C" and "F" are short for close and far respectively. D^{3}L achieves the best performance in terms of the F1 score across three scenes.

Scene	Method	AP(%)↑	F1(%)↑	x err/C(m)↓	x err/F(m)↓	z err/C(m)↓	z err/F(m)↓
Balanced Scene	3D-LaneNet [6]	89.3	86.4	0.068	0.477	0.015	0.202
	Gen-LaneNet [7]	90.1	88.1	0.061	0.496	0.012	0.214
	CLGo [15]	94.2	91.9	0.061	0.361	0.029	0.250
	PersFormer [2]	-	92.9	0.054	0.356	0.010	0.234
	GP [13]	93.8	91.9	0.049	0.387	0.008	0.213
	Anchor3DLane [11]	97.2	95.6	0.052	0.306	0.015	0.223
	LATR-Lite [16]	97.8	96.5	0.035	0.283	0.012	0.209
	LATR [16]	97.9	96.8	**0.022**	0.253	**0.007**	0.202
	D^{3}L (Ours)	**98.1**	**96.8**	0.030	**0.250**	0.012	**0.201**
Rare Subset	3D-LaneNet [6]	74.6	72.0	0.166	0.855	0.039	0.521
	Gen-LaneNet [7]	79.0	78.0	0.139	0.903	0.030	0.539
	CLGo [15]	88.3	86.1	0.147	0.735	0.071	0.609
	PersFormer [2]	-	87.5	0.107	0.782	0.024	0.602
	GP [13]	85.2	83.7	0.126	0.903	0.023	0.625
	Anchor3DLane [11]	96.9	94.4	0.094	0.693	0.027	0.579
	LATR-Lite [16]	97.2	95.8	0.060	0.618	0.020	0.538
	LATR [16]	97.3	96.1	**0.050**	0.600	**0.015**	0.532
	D^{3}L (Ours)	**97.5**	**96.2**	0.058	**0.598**	0.026	**0.520**
Visual Variations	3D-LaneNet [6]	74.9	72.5	0.115	0.601	0.032	0.230
	Gen-LaneNet [7]	87.2	85.3	0.074	0.538	0.015	0.232
	CLGo [15]	89.2	87.3	0.084	0.464	0.045	0.312
	PersFormer [2]	-	89.6	0.074	0.430	0.015	0.266
	GP [13]	92.1	89.9	0.060	0.446	**0.011**	0.235
	Anchor3DLane [11]	93.6	91.4	0.068	0.367	0.020	0.232
	LATR-Lite [16]	95.6	94.0	0.048	0.352	0.018	0.231
	LATR [16]	**96.6**	95.1	0.045	**0.315**	0.016	0.228
	D^{3}L (Ours)	95.8	**95.3**	**0.042**	0.330	0.022	**0.228**

total loss function of our D^{3}L consists of three parts: the lane point confidence loss $\mathcal{L}_{conf}$, the CCL $\mathcal{L}_{shape}$, and the 3D lane prediction loss $\mathcal{L}_{lane}$. The above can be expressed as:

$$\mathcal{L}_{lane} = w_x\mathcal{L}_x + w_z\mathcal{L}_z + w_v\mathcal{L}_v + w_c\mathcal{L}_c, \tag{16}$$

$$\mathcal{L} = w_s\mathcal{L}_{conf} + w_p\mathcal{L}_{shape} + w_l\mathcal{L}_{lane}, \tag{17}$$

where $w_{[*]}$ represent different loss weights. $\mathcal{L}_x$ and $\mathcal{L}_z$ constrain the predictions of the x and z axes respectively using smooth L1 loss. $\mathcal{L}_v$ denotes the visibility loss of lane points. $\mathcal{L}_c$ is the lane classification loss, which is calculated using the focal loss [14].

Inference. The proposed D^{3}L conducts denoising on noisy 3D lanes sampled from a Gaussian distribution, progressively refining its predictions over multiple sampling steps. For each sampling step, the CFD takes noisy lanes or the predicted lanes of the last sampling step as input and outputs the predicted lanes of the current step.

4 Experiments

4.1 Datasets

We conduct experiments on two popular 3D lane detection benchmarks: OpenLane [3] and ApolloSim [8].

OpenLane is a large-scale real world 3D lane detection benchmark based on the Waymo Open dataset [23]. It includes 200K frames and over 880K carefully annotated lanes in 14 categories. The scenes include highways, urban areas, and residential areas. This dataset contains various weather, terrain, and brightness conditions at a resolution of 1280×1920. Camera intrinsics and extrinsics are provided for each frame.

ApolloSim is a photo-realistic synthetic dataset generated using a game engine, containing over 10.5K images. The dataset comprises three distinct types of scenes: 1) Balanced scenes, 2) Rarely observed scenes, and 3) Scenes with visual variations. It includes diverse terrain structures such as highways, urban areas, residential zones, and downtown scenes, as well as varied lighting conditions across different times of day. Additionally, ApolloSim covers a range of weather conditions, road surface qualities, and traffic or obstacle variations, providing a comprehensive and challenging set of scenes.

4.2 Evaluation Metrics

We utilize the official evaluation metrics to assess our model's performance on the above two datasets. On ApolloSim dataset, we report the results of F1 score, Average Precision (AP), and x/z-errors. The predictions and ground truth lanes are matched using minimum-cost flow, with the pairwise cost defined as the square root of the sum of the pointwise Euclidean distances. A prediction is considered as true positive if over 75% of its points' distances to

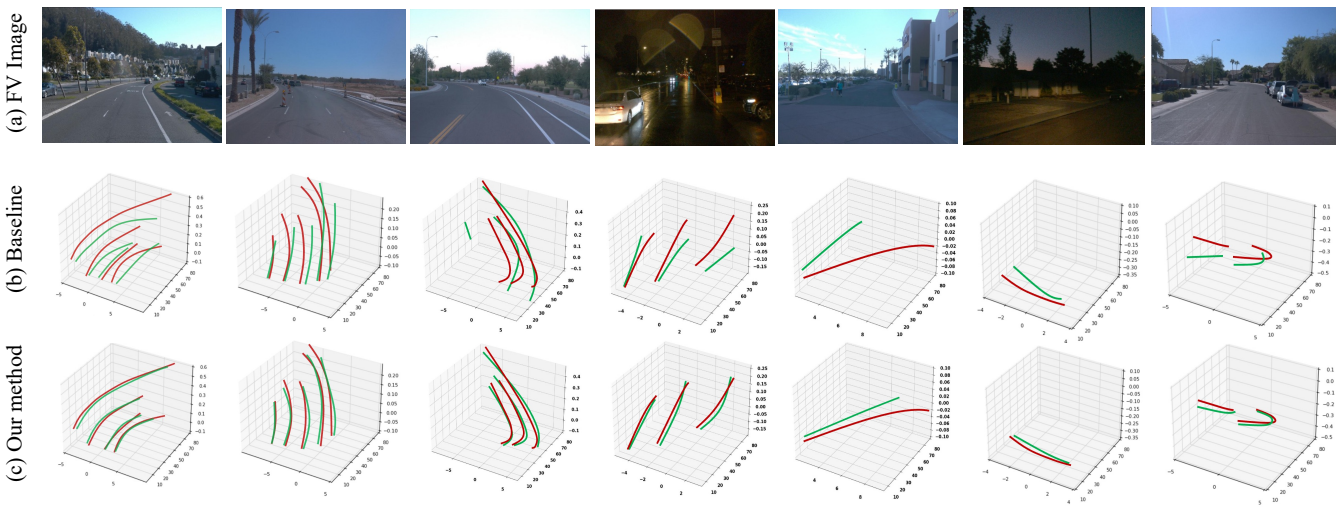


Figure 4: Qualitative results of the proposed D^{3}L and the baseline on OpenLane val set. The red and green lanes indicate the ground truth and prediction in 3D space, respectively.

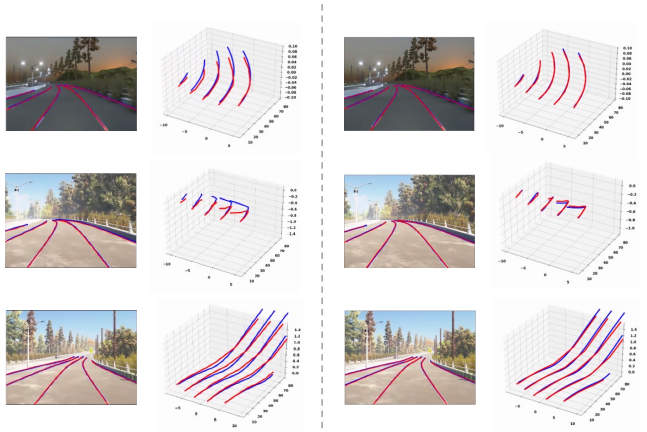
the ground truth points that are less than a threshold of 1.5 meters. Errors are evaluated for ranges from 0 to 40 meters and the far range of 40 to 100 meters along the heading direction. In addition to F1 score and x/z errors, we report category accuracy on OpenLane dataset. This metric calculates the proportion of predictions with correctly identified categories among all true positive predictions.

4.3 Implementation details

We use an input shape of 360 × 480 and adopt ResNet-18 [9] as the backbone of our D^{3}L. All our experiments are trained with the Adam optimizer [12] with a weight decay of 1×10^{-4}. We set the initial learning rate to 1×10^{-4}. Step learning rate decay is used during training. We use batch size 16 and train D^{3}L on two datasets with one NVIDIA RTX 4090 Ti GPU. We train the models for 60,000 iterations on ApolloSim and 100,000 iterations on OpenLane, respectively. More implementation details are provided in the Appendix.

4.4 Comparative Assessment

Quantitative results. In Table 1, we compare with previous methods under different scenes and report F1 score for each scene. Our method consistently improves performance across all scenes. Especially in the curve scene, our method achieves a significant improvement in F1 score compared to other methods, demonstrating the effectiveness of the diffusion model's progressive generation capability in capturing complex 3D lane geometries and enhancing D^{3}L's robustness across diverse real-world scenes. Table 2 shows the experimental results of our method on OpenLane validation set. Our D^{3}L outperforms LATR by 0.5% F1 score improvement. Moreover, our method reduces the x error far and z error far to 0.245 and 0.099, respectively, which is beneficial for the safety of autonomous driving. As shown in Table 3, we present the experimental results under three different split settings of the ApolloSim dataset, including balanced scene, rare subset and visual variations.



(a) Baseline (b) Our method

Figure 5: Qualitative comparison of results between (a) Baseline and (b) our method on the ApolloSim dataset. Blue: Ground-truth. Red: Prediction.

Our D^{3}L outperforms previous methods in F1 score across all the three splits, which shows the superiority of our method. Our D^{3}L also achieves comparable or even lower x/z errors compared with previous methods, especially for x error, highlighting the effectiveness of the diffusion model's progressive generation mechanism in refining lane predictions with enhanced flexibility and precision.

Qualitative results. In Figure 4, we present the detection results during the testing phase to better illustrate the performance of our method. Our D^{3}L demonstrates significantly more accurate detection in curved lanes compared to the Anchor3DLane [11] baseline. These qualitative results further highlight the effectiveness of the diffusion model's iterative denoising process in managing complex lane geometries, particularly in challenging curved scenes. This

Table 4: Performance gain for different contributions of designed modules. "Project" denotes the projection of noisy 3D lanes to 2D features.

Project	CFD	CCL	F1(%)↑	Gain(%)
(baseline)			55.6	+0.0
✓			57.8	+2.2
✓	✓		60.0	+4.4
✓	✓	✓	**62.4**	**+6.8**

Table 5: Per-frame runtime (in seconds) of the proposed D^{3}L.

Encoding	Denoising	Aggregation	Total
4.5×10^{-3}s	6.0×10^{-3}s	2.0×10^{-3}s	1.25×10^{-2}s

capability enables our D^{3}L framework to achieve enhanced robustness and adaptability across various lane structures. Additionally, our model notably reduces detection errors in diverse environments, contributing to greater overall stability. As shown in Figure 5, we compare our proposed D^{3}L framework with Anchor3DLane on the ApolloSim dataset. The results demonstrate that our method provides more accurate and consistent predictions for curved lanes in 3D space. While Anchor3DLane often struggles with curvature handling, D^{3}L achieves smoother lane transitions and closer alignment with the ground truth. This highlights the effectiveness of diffusion models in improving 3D lane prediction accuracy under challenging conditions.

4.5 Ablation Studies

In this section, we present an ablation analysis to validate the effectiveness of the proposed modules, conducting experiments on the OpenLane validation set to evaluate their individual contributions. We also analyze the effect of sample size and denoising steps, and report runtime to show real-time performance.

Effectiveness of designed modules. As shown in Table 4, we establish a denoiser based on standard Transformer Blocks [24] as the baseline. First, by projecting noisy 3D lanes to 2D features, we achieve a 2.2% improvement in F1 score, enhancing the model's ability to capture essential lane features. Then, our designed CFD further improves the F1 score by an additional 2.2%, providing a more refined denoising process that better preserves lane structure. Finally, adding the CCL significantly enhances the F1 score by 2.4%, reinforcing the model's capacity to maintain curvature consistency and improve geometric accuracy.

Number of samples. In Figure 6(a), we present how the model's performance is affected by the number of samples under three different settings. All settings start with the same performance (sampling = 1) and then gradually improve. "P-Best" converges at a sampling number of 10. For both performance and efficiency, we choose 10 as the final sampling number. "P-Avg" shows little variation in overall performance. "P-W.Avg" shows an upward trend but is less effective than "P-Best" because confidence-weighted averaging weakens the impact of selecting the optimal choice.

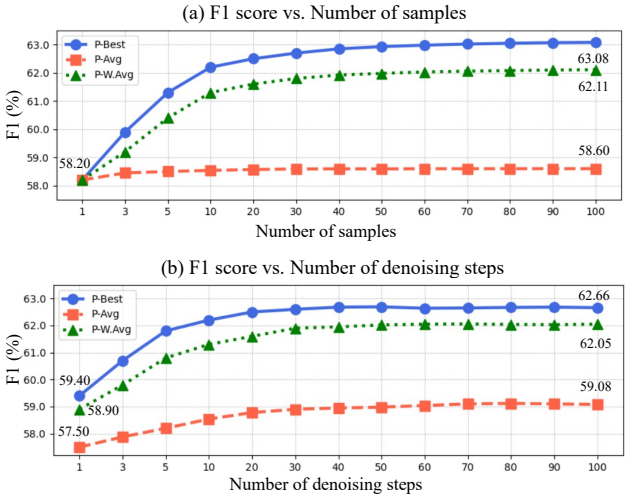


Figure 6: Ablation experiments on the number of (a) samples (denoising steps = 10) and (b) denoising steps (samples = 10). "P-Best" denotes the lane point with the highest confidence. "P-Avg" and "P-W.Avg" denote the average and weighted average of all predicted lane points, respectively.

Number of denoising steps. Figure 6(b) illustrates how performance fluctuates with the number of denoising steps. The three settings show different effects even with a single denoising step. Under the sampling setting of 10, "P-Best" clearly exhibits the best performance. As the number of denoising steps increases, the F1 score continuously improves, but the trend slows down after 10 steps. We set the number of denoising steps to 10 as the final choice in our model. "P-Avg" and "P-W.Avg" also show an overall trend of performance improvement with the increase in denoising steps. Similarly, the model's performance stabilizes after 10 steps.

Runtime. Table 5 lists the runtime for each stage of D^{3}L. The processing speed of D^{3}L is about 80 frames per second, which meets the real-time requirements for autonomous driving. Note that the Denoising stage includes both the initialization of noisy 3D lanes and the iterative refinement process with 10 steps.

5 Conclusion and Future Work

In this paper, we propose a novel diffusion-based framework D^{3}L for 3D lane detection. D^{3}L includes three innovative components: CFD, CCL and MSAS. CFD is introduced to accurately denoise 3D lanes by incorporating both lane-level and point-level transformer blocks, effectively capturing both global and local features. CCL is formulated to minimize lane curvature deviations, improving accuracy and geometric consistency in lane detection. MSAS is designed to select the optimal lane point-by-point from multiple candidates, reducing randomness and enhancing robustness. Experimental results demonstrate that D^{3}L achieved state-of-the-art performance over the existing mainstream methods on popular 3D lane detection benchmarks. Future work includes applying this framework to multi-modal datasets, incorporating both FV images and point cloud data, to further enhance performance.

References

[1] Yifeng Bai, Zhirong Chen, Zhangjie Fu, Lang Peng, Pengpeng Liang, and Erkang Cheng. 2023. Curveformer: 3d lane detection by curve propagation with curve queries and attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7062–7068.

[2] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. 2022. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*. Springer, 550–567.

[3] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. 2022. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*. Springer, 550–567.

[4] Wencan Cheng, Hao Tang, Luc Van Gool, and Jong Hwan Ko. 2024. HandDiff: 3D Hand Pose Estimation with Diffusion on Image-Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2274–2284.

[5] Runyang Feng, Yixing Gao, Tze Ho Elden Tse, Xueqing Ma, and Hyung Jin Chang. 2023. DiffPose: SpatioTemporal diffusion model for video-based human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14861–14872.

[6] Noa Garnett, Rafi Cohen, Tomer Pe'er, Roee Lahav, and Dan Levi. 2019. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2921–2930.

[7] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. 2020. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 666–681.

[8] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. 2020. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 666–681.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[11] Shaofei Huang, Zhenwei Shen, Zehao Huang, Zi-han Ding, Jiao Dai, Jizhong Han, Naiyan Wang, and Si Liu. 2023. Anchor3dlane: Learning to regress 3d anchors for monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17451–17460.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Chenguang Li, Jia Shi, Ya Wang, and Guangliang Cheng. 2022. Reconstruct from BEV: A 3D Lane Detection Approach based on Geometry Structure Prior. *arXiv preprint arXiv:2206.10098* (2022).

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[15] Ruijin Liu, Dapeng Chen, Tie Liu, Zhiliang Xiong, and Zejian Yuan. 2022. Learning to predict 3d lane shape and camera pose from a single image via geometry constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 1765–1772.

[16] Yueru Luo, Chaoda Zheng, Xu Yan, Tang Kun, Chao Zheng, Shuguang Cui, and Zhen Li. 2023. Latr: 3d lane detection from monocular images with transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7941–7952.

[17] Maximilian Pittner, Joel Janai, Alexandru P Condurache, and Alexandru P Condurache. 2024. LaneCPP: Continuous 3D Lane Detection using Physical Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10639–10648.

[18] Tong Qin, Tongqing Chen, Yilun Chen, and Qing Su. 2020. Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot. In *2020 IEEE/RSJ International Conference on intelligent robots and systems (IROS)*. IEEE, 5939–5945.

[19] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. 2023. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14761–14771.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[21] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).

[22] Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models. *Advances in neural information processing systems* 33 (2020), 12438–12448.

[23] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2446–2454.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[25] Zewen Zheng, Xuemin Zhang, Yongqiang Mou, Xiang Gao, Chengxin Li, Guoheng Huang, Chi-Man Pun, and Xiaochen Yuan. 2024. PVALane: Prior-Guided 3D Lane Detection with View-Agnostic Feature Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 7597–7604.

[26] Sheng Zhu and Bilin Aksun-Guvenc. 2020. Trajectory planning of autonomous vehicles based on parameterized control optimization in dynamic on-road environments. *Journal of Intelligent & Robotic Systems* 100, 3 (2020), 1055–1067.