Mono3DLane: Efficient Monocular 3D Lane Detection via 2D Prior Anchors

Anonymous submission

Abstract

Accurately detecting 3D lanes using monocular cameras presents significant challenges. Previous methods have faced practical limitations due to complex spatial transformations and rigid learning of lane-aware features. In this paper, we propose Mono3DLane, an innovative framework that constructs 3D lanes directly from monocular images without requiring view conversion. To efficiently model 3D spatial relationships from front-view (FV) images, we incorporate Spatial Hierarchy-aware Attention and carefully designed 2D prior anchors, which directly operate on FV features to generate 3D lane representations. Moreover, to mitigate the inherent distortion in monocular images, we develop a lightweight view context model (VCM) to align raw visual features and refine 3D lane parameters using a cross-layer dynamic query strategy. The effectiveness of the proposed components is thoroughly discussed, and experimental results demonstrate that our approach achieves state-of-the-art performance on OpenLane and ONCE-3DLanes datasets.

Introduction

3D lane detection is crucial for autonomous driving, extracting structural and traffic information from the road in a three-dimensional context, thereby facilitating environmental interaction and route planning for self-driving vehicles (Williams et al. 2022). Owing to the cost-effectiveness of sensors and the richness of visual data in color detail, monocular 3D lane detection has emerged as a significant research area within autonomous driving, garnering considerable attention from both industry and academia (Garnett et al. 2019; Guo et al. 2020; Bai et al. 2023).

Predicting 3D lanes directly from 2D images poses significant challenges due to the inherent size variance and depth ambiguity of monocular visuals. Consequently, it is vital to extract beneficial features from the front-view (FV) image to represent the three-dimensional structure of the lane (Luo et al. 2022). According to the source of feature extraction, 3D lane detection methods can be divided into two categories: bird-eye-view (BEV)-based and FV-based.

BEV-based methods (Efrat et al. 2020a; Bai et al. 2023; Luo et al. 2023) employ inverse perspective mapping (IPM) to warp images from FV space to BEV, estimating height values from BEV-derived features to construct 3D lane representations, as shown in Fig. 1(a). Although intuitive, IPM depends heavily on the assumption of flat ground, which often contradicts real road conditions. PersFormer (Chen et al. 2022) attempts to introduce deformable attention to relieve this issue. Nonetheless, the separation of view transformation and lane height estimation leads to error accumulation and increased complexity. Given these shortcomings, several studies have explored FV-based lane detection (Yan et al. 2022; Huang et al. 2023). Anchor3DLane (Huang et al. 2023) utilizes defined dense 3D surrogate anchors and sampled FV features to model 3D lanes. While this approach circumvents the need for view transformation, it faces challenges: (i) FV features extracted by conventional components lack depth perception, (ii) sampling accuracy is affected by the limitations of monocular images, and (iii) the use of dense 3D anchors extends post-processing time.

To address these issues, we introduce the concept of integrated front-view (IFV) and propose Mono3DLane, a novel network that predicts 3D lanes directly using sparse 2D prior anchors in front-view space. As shown in Fig. 1(b), we develop a Spatial Hierarchy-aware Attention to seam-

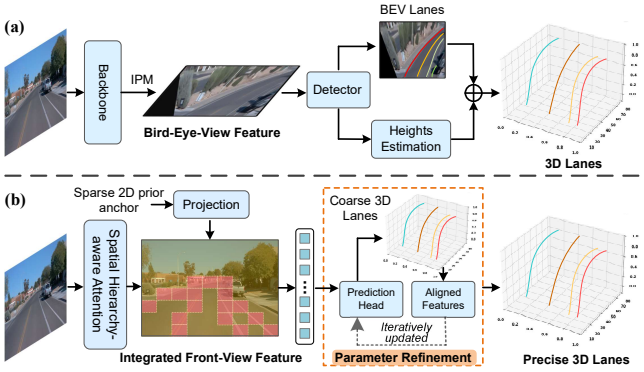

Figure 1: (a) Previous methods primarily perform 3D lane detection on warped BEV features, relying on separate processing detectors to derive the final 3D lane representation. (b) In contrast, our Mono3DLane bypasses view conversion and separate detection steps, directly modeling 3D lanes from front-view images. By employing Spatial Hierarchy-aware Attention and a dynamic refinement strategy, our network effectively leverages the spatial information within features to achieve superior 3D lane detection.

lessly integrate cross-regional lane dependencies and adaptively learn 3D spatial values, providing the prior anchors with the lane-aware features to construct initial 3D lanes. This method narrows the boundaries of 2D and 3D detection tasks, significantly reducing post-processing time.

Furthermore, we introduce a lightweight View Context Model (VCM) to address the geometric distortions inherent in monocular images, and we design a parameter refinement strategy that integrates CNNs and Transformers. This strategy dynamically updates query content by leveraging multi-scale feature layers rich in both semantic and local information, enabling iterative 3D lane detection in a coarse-to-fine manner. Our end-to-end framework maintains efficiency while maximizing information extraction from monocular images. Experimental results demonstrate that our proposed method achieves state-of-the-art performance on the OpenLane and ONCE3DLanes benchmarks.

Our main contributions are the following:

- We propose **Mono3DLane**, an end-to-end 3D lane detection framework that directly predicts 3D lanes using sparse 2D prior anchors in the front-view space. Mono3DLane eliminates the need for complex visual transformations while maintaining high efficiency.
- We introduce the concept of integrated front-view (IFV) and develop a Spatial Hierarchy-aware Attention mechanism that enhances lane-aware features and adaptively learns 3D spatial values, facilitating the bridge between 2D and 3D lane detection.
- We present a cross-layer dynamic refinement strategy to iteratively and accurately regress 3D lane positions. Additionally, the proposed View Context Model (VCM) aligns multi-scale visual features and can be seamlessly integrated into other networks.

Related Work

3D Lane Detection in Bird's-Eye-View. These methods initially transform the front-view image into BEV space by using IPM and subsequently detect lanes based on the resulting BEV features (Liu et al. 2022a,b; Wang et al. 2023a). 3D-LaneNet+ (Efrat et al. 2020b) constructs the shape of lanes in predefined grid cells based on the assumption of straight line segments. This method enhances both detection accuracy and efficiency. However, IPM heavily relies on the assumption of flat ground, potentially resulting in misalignment between lanes represented in BEV space and their corresponding 3D positions on uneven ground. To address this limitation, Performer (Chen et al. 2022) employs deformable attention to generate fine-grained BEV features more adaptively and robustly. However, the improvement in detection accuracy is limited by the assumption of flat ground and the absence of height values in BEV features.

3D Lane Detection in Front-View. Another approach involves direct prediction of 3D lanes from front-view (FV) features. These methods utilize common feature extraction components employed in 2D visual perception tasks to obtain FV features. Then, the parameters necessary for constructing a 3D lane are inferred by the obtained FV features. SALAD (Yan et al. 2022) decomposes the 3D lane detection task into 2D lane segmentation and dense depth estimation. It predicts 2D lanes using the segmenting head and combines the estimated depth information. However, this method lacks a structured representation of 3D lanes, and its performance still lags behind state-of-the-art methods. Anchor3DLane (Huang et al. 2023) projects the dense anchor defined in 3D space onto the FV to obtain sampling features, and lanes are constructed based on these features and camera parameters. While FV-based methods avoid view transformation, the inherent lack of 3D spatial information and limitations of FV compromise detection accuracy. In contrast, we introduce the integrated front-view (IFV) feature that emphasizes crucial areas and incorporates 3D values from the input monocular image. Instead of densely defined 3D anchors, we utilize sparse 2D prior anchors combined with dynamic queries innovatively to generate robust 3D representations directly from IFV.

Spatial self-attention. Self-attention has proven effective in modeling global range dependencies (Devlin et al. 2018) and is widely used in pixel token relational modeling across various visual tasks (Vaswani et al. 2017; Carion et al. 2020). Most spatial self-attention methods employ absolute or relative positional embedding (Child et al. 2019; Chu et al. 2021), with the adaptation of 3D position embeddings on 2D planes being critical for decoding 3D spatial relationships. Additionally, several studies have aimed to enhance efficiency (Ramachandran et al. 2019; Katharopoulos et al. 2020), focusing on regional attention (Liu et al. 2021). Known as shifted window attention (Liu et al. 2022c), this method highlights key regions of objects efficiently through iterative processes. In lane detection tasks, where lanes typically occupy only a small portion of an image, employing a lane-aware region for effective feature sampling is advantageous (Tabelini et al. 2021; Qu et al. 2021; Li et al. 2023). Building on this foundation, this paper introduces Spatial Hierarchy-aware Attention, which integrates essential lane-aware regions and spatial metrics from an image.

Method

The overall architecture of Mono3DLane is illustrated in Fig. 2. Given a batch of front-view images $\boldsymbol{I} \in \mathbb{R}^{B\times 3\times H_i\times W_i}$, a Feature Pyramid Network (FPN) (Lin et al. 2017) is employed to extract multi-scale visual features $\{\mathbf{F_0}, \mathbf{F_1}, \mathbf{F_2}\}$. Unlike methods that construct lane topology from BEV or FV alone, our method aims to fully exploit the feature context containing 3D spatial information without space transformation. High-level features encapsulate rich semantic content, while low-level features excel at capturing fine edge details, enhancing precise object delineation (Chen et al. 2021; Zhuang et al. 2023). Utilizing these insights, the Integrated Front-View (IFV) $\mathbf{F_2^{'}} \in \mathbb{R}^{C\times H_I\times W_I}$ is extracted from the highest level $\mathbf{F_2}$, maintaining consistent dimensions. The view context model (VCM) aligns visual features at lower levels $\mathbf{F_0}$ and $\mathbf{F_1}$, which are then fed into the dynamic query during the refinement iteration.

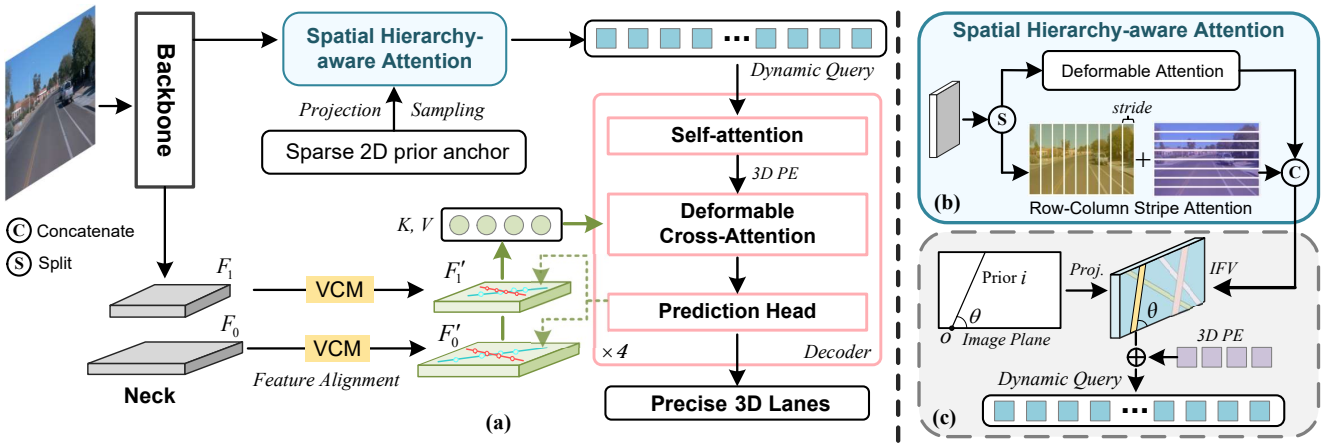

Figure 2: The overall architecture of Mono3DLane, as shown in part (a). The front-view image is initially processed by a CNN-based backbone to obtain multi-scale feature map layers, which are then aligned using the View Context Model (VCM). Subsequently, the 2D prior anchors are projected onto the Integrated Front-View (IFV) generated by Spatial Hierarchy-aware Attention (b), and the initial query content is generated through a sampling process of this projection (c). During the refinement stage of the decoder stack, the prediction head directly generates 3D lanes and projects them back onto the aligned features to update the query before the next decoding step, ultimately producing an accurate 3D lane representation.

Lane and Anchor Representation

3D Lane Representation. Our pipeline utilizes two primary coordinate systems: the camera coordinate system, aligned with the front-view image, and the ground coordinate system, which is used for annotating 3D lanes. Following Chen et al. (Chen et al. 2022), in the ground coordinate system, the 3D lane is defined by a sequence of 3D points, with N points uniformly sampled along the $\mathbf{y}$-coordinates $Y = \{y^k\}_{k=1}^{N}$. As the $\mathbf{y}$-coordinate remains constant, the identity of a 3D lane depends primarily on its $\mathbf{x}$ and $\mathbf{z}$ coordinates. Specifically, the i-th 3D lane is represented as $\mathbf{L}_i = \{\mathbf{p}_i^k\}_{k=1}^{N}$, where each k-th point is denoted by $\mathbf{p}_i^k = (x_i^k, y^k, z_i^k, v_i^k)$, and v_i^k indicates the visibility of $\mathbf{p}_i^k$.

2D Prior Anchor. Lanes are characterized by slender shapes with well-defined shape priors, making predefined lane priors instrumental in initializing query content and accelerating model convergence. Anchor3DLane (Huang et al. 2023) utilizes dense anchors defined in 3D space to predict lanes, which results in a significant number of invalid predictions and diminishes the benefits of using priors knowledge. In contrast, our 2D prior anchors are strategically placed within the front-view image in the camera coordinate system. Specifically, a 2D prior anchor is a virtual ray originating from one of the image borders (excluding the top border) and directed at an angle θ, as illustrated in Fig. 2(c). This placement aligns more closely with the actual distribution of lanes. Similar to the 3D lane representation, a prior anchor is depicted as a collection of 2D points, evenly distributed along the $\mathbf{y}$-axis, denoted as $P = \{(x_0^p, y_0^p), \cdots, (x_{N-1}^p, y_{N-1}^p)\}$. The initial number of prior anchors is N_p. Sampling is guided by these preset prior anchors, which provide the query content with valuable lane prior knowledge derived from the front-view image. To detect 3D lane, the network must estimate five components: (1) foreground and background probabilities, (2) the lane's category, (3) the visibility v_i^k of each point within the lane, (4) the offset $\mathbf{\Delta}\boldsymbol{x}_i$ for each point along the $\mathbf{x}$-axis, and (5) the offset $\mathbf{\Delta}\boldsymbol{z}_i$ for each point along the $\mathbf{y}$-axis.

Spatial Hierarchy-aware Attention

Motivation. Instead of using complex spatial transformations in previous methods, we aim to implicitly model intra-lane and inter-lane relationships directly from FV features. The inherent challenges of detecting 3D lanes from monocular images include the lack of explicit 3D value and the complexity of lane topology. To this end, we introduce Spatial Hierarchy-aware Attention during the feature learning stage. This mechanism is specifically designed to capture relationships within lane-aware regional features and effectively consolidate 3D spatial information.

Attention Structure. As shown in Fig. 2(b), Spatial Hierarchy-aware Attention incorporates the deformable attention (Zhu et al. 2020) and the novel Row-Column Stripe Attention proposed in this paper. The input FV feature map is evenly split along the channel dimension and subsequently concatenated after parallel processing in the two attention modules. Stripe attention is designed to capture regional objects through hierarchical stripe iterations. Concurrently, deformable attention extends beyond adapting to the lane local edge (Chen et al. 2022) and incorporates a custom-designed dynamic Adaptive 3D Position Embedding.

Row-Column Stripe Attention. We use intersecting vertical and horizontal rectangular stripes to encompass various lane topology types, as illustrated in Fig. 4, to pool lane-aware feature. Additionally, the space complexity of self-attention $\mathcal{O}\left(M^2\right)$ can be reduced by processing features aggregated by row-column stripes and adjusting the number of tokens M. By sequentially pooling the original features along the horizontal and vertical axes with a stride size of S, we generate a condensed summary of the row-column stripe

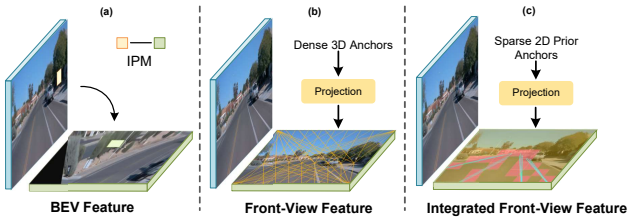

Figure 3: Comparison of lane-aware learning methods. Our method (c) aligns with prior knowledge, circumvents the inherent limitations of IPM (a), and eliminates the predefined complexity associated with dense 3D anchors (b).

features. This summary encapsulates compact dimensional information from the feature map and serves as a foundation for subsequent similarity comparisons. Taking the row stripe attention calculation as an example, let the number of stripes be L and $\mathbf{A} \in \mathbb{R}^{L\times d}$ be the intermediary token of stripe aggregation, and the query-to-stripe operation is as follows:

$$\mathbf{O_{q2s}} = \text{Softmax}\left(\mathbf{A}\cdot\mathbf{K}_s^T/\sqrt{d}\right)\cdot\mathbf{V}_s \tag{1}$$

where $L \ll M, \mathbf{K}_s, \mathbf{V}_s \in \mathbb{R}^{M\times d}$ are the key and value matrices, and d denote the dimension of one token. L is only related to the stride size and feature map height. $\mathbf{O_{q2s}}$ explores the relationship between query (pixels in stripes) and stripes. Then the stripe-to-value operation is as follows:

$$\mathbf{O_{s2v}} = \text{Softmax}\left(\mathbf{Q}_s\cdot\mathbf{A}^T/\sqrt{d}\right)\cdot\mathbf{Z} \tag{2}$$

where $\mathbf{Q}_s \in \mathbb{R}^{M\times d}$ and $\mathbf{Z} \in \mathbb{R}^{L\times d}$ are the query matrices and intermediate feature from $\mathbf{O_{q2s}}$, respectively. The $\mathbf{O_{s2v}}$ expands the size of the feature $\mathbf{Z}$ and recovers the region information in $\mathbf{V}_s$. The column stripe attention mirrors this process. The concurrent application of row and column attention forms the core of Row-Column Stripe Attention. As a result, the space complexity is reduced to $\mathcal{O}(ML)$ and iterates once Row-Column Stripe Attention time complexity is traditional self-attention $\frac{1}{S^2}$. The choice of the stride size of this attention is discussed in the ablation study.

Adaptive 3D Position Embedding. Intuitively, regions of the image that feature curb structures hold critical spatial information essential for modeling 3D lanes. To harness this information, we introduce the Adaptive 3D Position Embedding, trained to determine height values using ground-truth (GT) labels. This embedding is seamlessly integrated into the deformable attention mechanism to encode areas of the feature map that are critical for representing 3D lanes. The 3D position embedding of a feature map, with height H_f and width W_f, is denoted by $\mathbf{E}_{uv} \in \mathbb{R}^{H_f\times W_f\times C}$. We initially establish a 2D coordinate grid $\mathbf{G} \in \mathbb{R}^{H_f\times W_f\times 3}$ in camera coordinate system. During the training phase, The GT coordinates are projected onto $\mathbf{G}$ through a projection matrix. Each projection point on $\mathbf{G}$, expressed as $p_i = (u_i \times h_i, v_i \times h_i, h_i)$, correlates with the pixel coordinates (u_i, v_i) in the FV feature map and incorporates the corresponding height value h_i. Finally, the data derived from grid sampling is processed through a linear layer, creating a matrix for height value distributions $\mathbf{H}_{uv} \in \mathbb{R}^{H_f\times W_f\times C}$ learned from GT height label. In the inference phase, $\mathbf{H}_{uv}$ serves to estimate the height value near the prior anchor. The adaptive 3D position embedding is obtained as follows:

$$\mathbf{E}_{uv} = \left[\mathbf{H}_{uv} + (\mathbf{G}\mathbf{W}_1 + \mathbf{b_1})\right]\mathbf{W}_2 + \mathbf{b_2} \tag{3}$$

where $(\mathbf{G}\mathbf{W}_1 + \mathbf{b_1})$ denotes the learnable 2D position embedding in the grid. The $\mathbf{W}_1 \in \mathbb{R}^{3\times C}$, $\mathbf{b_1} \in \mathbb{R}^C$, $\mathbf{W}_2 \in \mathbb{R}^{C\times C}$ and $\mathbf{b_2} \in \mathbb{R}^C$ are learnable weights. Flattening $\mathbf{E}_{uv}$ can generate the 3D position embedding vector $\mathbf{PE} \in \mathbb{R}^{(H_f\times W_f)\times C}$ input attention.

Projection and Refinement

Instead of using redundant features as queries like previous methods (Chen et al. 2022; Huang et al. 2023), our proposed prior anchor-based learning approach is more explicit and efficient. It provides dynamic queries with greater flexibility in depicting lanes and subtly models both intra-lane and inter-lane relationships. Initially, the prior anchor points defined in the FV image are scaled projections mapped onto the IFV feature for the respective proposals. For each $y_i = 0, 1, 2, \ldots, N-1$, there is a corresponding single $\mathbf{x}$-coordinate:

$$x_j = \left[\frac{1}{\tan\theta}\left(y_j - y_{orig}/\delta_b\right) + x_{orig}/\delta_b\right] \tag{4}$$

where (x_{orig}, y_{orig}) represents the origin point and δ_b denotes the scaled of backbone stride. Each prior anchor i, containing N points, has its corresponding query generated through bilinear interpolation sampling. Consequently, all queries are aggregated to initialize dynamic queries $\mathbf{Q} \in \mathbb{R}^{(N_p\times N)\times C}$. The initial dynamic query is processed by the prediction head of the decoder to yield preliminary 3D lane parameters (*e.g.*, $\boldsymbol{\Delta x}$, $\boldsymbol{\Delta z}$).

Refinement structure. During the refinement stage, camera intrinsic parameters are used to project these preliminary 3D lane points from the ground coordinate system onto the aligned features, and the query context is dynamically updated based on the sampling information. This process is illustrated in Fig. 2(a). The projection process of the k-th point in the j-th predict lane is as follows:

$$\begin{bmatrix}\tilde{u}_j^k\\ \tilde{v}_j^k\\ d_j^k\end{bmatrix} = \mathbf{K_c}\mathbf{T}\begin{bmatrix}x_j^k\\ y_j^k\\ z_j^k\\ 1\end{bmatrix} \tag{5}$$

where $\mathbf{K_c} \in \mathbb{R}^{3\times 3}$ denotes camera intrinsic parameters, $\mathbf{T} \in \mathbb{R}^{3\times 4}$ denotes the transform matrix from the ground coordinate to camera coordinate. Subsequently, the 2D coordinates $\left(u_j^k, v_j^k\right) = \left(s_w\cdot\frac{\tilde{u}_j^k}{d_j^k}, s_h\cdot\frac{\tilde{v}_j^k}{d_j^k}\right)$ are determined using scale factors $s_w = W_f/W_i$ and $s_h = H_f/H_i$. Once again, the key and value are obtained by sampling the features around the projection point through bilinear interpolation, and the query is dynamically updated to generate a more accurate lane representation.

Non-local Based View Context Model

The inherent scale ambiguity and distortion in monocular views directly impact the accurate regression of lane positions (Chen et al. 2022; Wang et al. 2023b). To address this, we have developed the lightweight View Context Model (VCM) to realign low-level, multi-scale visual features, thereby enhancing detection accuracy.

VCM redefines feature context by calculating non-local relationships (Wang et al. 2018) between adapted reference features and features needing alignment. Specifically, deformable convolutions (Dai et al. 2017) are first applied to the FV features to generate intermediate reference features, corresponding to the context that requires correction. Subsequently, the original FV feature and the reference feature serve as inputs for non-local similarity calculation. This process captures the relationship between every pixel pair in the flattened FV feature and the flattened reference feature. As a result, instead of relying on fixed mapping, the weights learned by VCM offer flexible adaptability to variations caused by different defects. Further model details will be discussed in the supplementary material.

Decoder and Losses

Following the standard Transformer (Katharopoulos et al. 2020), we build a multi-layer stacked decoder. In each layer, dynamic queries $\mathbf{Q} \in \mathbb{R}^{(N_p \times N) \times C}$ and adaptive 3D position embedding $\mathbf{PE} \in \mathbb{R}^{(H_f \times W_f) \times C}$ are fed into a standard deformable cross-attention module (Zhu et al. 2020) as follow:

$$\mathbf{Q}_l = \text{DeformAttn}\left(\mathbf{Q}_{l-1}, \mathbf{X}^T + \mathbf{PE}^T, \mathbf{X}^T + \mathbf{PE}^T\right) \quad (6)$$

where $\mathbf{X} \in \mathbb{R}^{C \times (H_f \times W_f)}$ represents the features extracted by re-projecting the predicted 3D lane back to the plane, while l denotes the layer index. By stacking decoders, Mono3DLane progressively refines the predicted parameters to enhance lane detection accuracy.

Prediction Head. The prediction head comprises a classification head and a regression head. We employ a feed-forward network (FFN) (Vaswani et al. 2017) to predict parameters. For the lane 3D positions estimation, the regression head can be formulated as:

$$[\boldsymbol{\Delta x}, \boldsymbol{\Delta z}, \boldsymbol{v}] = \text{FFN}_{reg}(\mathbf{Q}) \quad (7)$$

where $\boldsymbol{\Delta x}, \boldsymbol{\Delta z} \in \mathbb{R}^{N_p \times N \times 1}$ denote the offsets corresponding prior anchor. $\boldsymbol{v} \in \mathbb{R}^{N_p \times N \times 1}$ denotes the visibility of each point. For the lane category, the classification head is as follows:

$$\mathcal{C} = \text{FFN}_{cls}(\mathbf{Q}) \quad (8)$$

where $\mathcal{C} \in \mathbb{R}^{N_p \times K}$ denotes the class-logits and K is the number of possible classes. Lanes classified as background are discarded. We adopt a hard-matching strategy of calculating the distance to GT label (Huang et al. 2023) for 3D lane label assignment. During the inference phase, we use Non-Maximum Suppression (NMS) to keep a reasonable number of proposals.

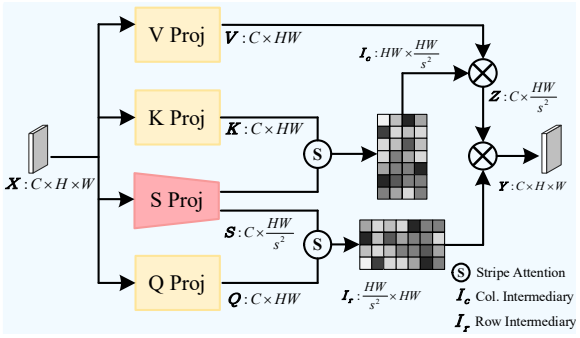

Figure 4: Illustration of Row-Column Stripe Attention. Our proposed attention mechanism is both simple and efficient, enabling hierarchical attention to regional areas enriched with lane-aware features.

Total Loss. Given matched ground truth labels for the lane queries, we calculate the corresponding loss for each matched pair. The overall loss of our Mono3DLane model includes both classification loss and regression loss:

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{reg} \quad (9)$$

where $\mathcal{L}_{cls}$ is the focal loss with $\gamma = 2.0$ and $\alpha = 0.25$, same as (Bai et al. 2023; Chen et al. 2022). $\mathcal{L}_{reg}$ is the smooth-l_1 loss for each prediction point in the $\mathbf{x}$-axis and $\mathbf{z}$-axis directions.

Experiments

We evaluate our method on the two most popular 3D lane detection benchmarks: OpenLane (Chen et al. 2022) and Once-3DLanes (Yan et al. 2022).

Datasets and Evaluation Metrics

OpenLane (Chen et al. 2022) is a large-scale real-world 3D lane detection benchmarks. It includes 160k training images and 40K validation images, with accompanying camera intrinsics. The validation set consists of six different scenarios, including curve, extreme weather, night, etc. This complex scene greatly tests the performance of the detector.

Once-3DLanes (Yan et al. 2022) comprises a total of 200k frames and over 880K lanes are annotated. Compared with OpenLane, it only demands detectors to localize the lanes but not classify them. It covers diverse temporal conditions, encompassing mornings, noons, nights, etc.

Evaluation Metrics. We follow official evaluation metrics and adopt the F1 score for the evaluation of regression and classification. Specifically, a prediction is considered a true positive if over 75% of its points' distances to ground-truth points are less than the maximum allowed distance of 1.5 meters. In addition, x/z errors are calculated in the near (0-40m) and far (40-100m) ranges, respectively. For ONCE-3DLanes, we utilize the Intersection over Union (IoU) to perform lane matching in the top view. If the computed IoU value surpasses the defined threshold of 0.3, we employ a unilateral Chamfer Distance (CD) metric to quantify the error in matching the curves.

Method	Up& Down	Curve	Extreme Weather	Night	Intersection	Merge& Split	All
3D-LaneNet (Garnett et al. 2019)	40.8	46.5	47.5	41.5	32.1	41.7	44.1
GenLaneNet (Guo et al. 2020)	25.4	33.5	28.1	18.7	21.4	31.0	32.3
PersFormer (Chen et al. 2022)	42.4	55.6	48.6	46.6	40.0	50.7	50.5
Anchor3DLane (Huang et al. 2023)	46.7	57.2	52.5	47.8	45.4	51.2	53.7
Curveformer++ (Bai et al. 2024)	48.1	59.0	51.2	48.5	**51.2**	48.1	52.5
Mono3DLane	50.1	59.9	53.9	50.9	46.9	52.9	56.3
Mono3DLane†	**52.0**	**61.5**	**55.2**	**52.9**	48.4	**53.8**	**57.8**

Table 1: Comparison with state-of-the-art methods on OpenLane under different scenarios. † denotes using ResNet50 backbone.

Method	F1(%)↑	X err.near(m)↓	X err.far(m)↓	Z err.near(m)↓	Z err.far(m)↓	FPS↑
3D-LaneNet (Garnett et al. 2019)	44.1	0.479	0.572	0.367	0.443	-
GenLaneNet (Guo et al. 2020)	32.3	0.591	0.684	0.411	0.521	54
PersFormer (Chen et al. 2022)	50.5	0.485	0.553	0.364	0.431	21
Anchor3DLane (Huang et al. 2023)	53.7	0.276	0.311	0.107	0.138	-
Curveformer++ (Bai et al. 2024)	52.5	0.333	0.805	0.186	0.687	20
Mono3DLane	56.3	0.266	0.285	0.083	0.112	**82**
Mono3DLane†	**57.8**	**0.245**	**0.292**	**0.079**	**0.110**	71

Table 2: Comparison with state-of-the-art methods on OpenLane validation set. We report the F1 score, regression errors in the **x**-axis and **z**-axis directions, and FPS. FPS was measured on one NVIDIA 3090 GPU and based on Pytorch framework.

Implementation Details

We adopt ResNet-18 and ResNet-50 (He et al. 2016) as our backbones and all input shapes are resized to 360×480. Similar to (Huang et al. 2023), we perform data augmentation methods including random rotation and random horizontal flips. The starting positions x_{orig} of prior anchors are evenly placed along the **x**-axis with an interval of 1.3m, and the $\theta \in \{0°, \pm1°, \pm2°, \pm5°\}$. We set the number of prior anchors $N_p = 192$, and the sampling points of anchor $N = 20$. For Spatial Hierarchy-aware Attention, the stride is 8. The shape of the IFV feature map is $45 \times 60 \times 256$. For the decoder, we employ deformable cross-attention (Zhu et al. 2020) with 4 attention heads, 8 sample points, and 256-D embeddings. In the training process, we use Adam optimizer with an initial learning rate of $1e^{-4}$ and step learning rate decay with weight decay set as $1e^{-4}$. The λ_{cls} and λ_{reg} are set to 1.5 and 1. More details about our Mono3DLane are included in supplementary materials.

Quantitative Results

Performance on OpenLane. Tab. 1 shows the results under six challenging scenarios. Mono3DLane significantly enhances performance in all scenarios, particularly in the Up & Down scenario, demonstrating the powerful advantage of 3D space perception. In Tab. 2, Mono3DLane achieves a new state-of-the-art with a 57.8% F1 score. Utilizing ResNet-18 as the backbone, our method achieves a 56.3% F1 and 82 FPS, outperforming other methods in both speed and accuracy. Notably, our method also records the lowest location errors in the **x**-axis and **z**-axis directions, indicating more precise regression of 3D lane locations.

We present the qualitative results in Fig. 5. The performance of BEV-based methods, such as PersFormer(Chen et al. 2022), is significantly compromised in challenging scenes. In contrast, our method successfully reconstructs the 3D lane structure even under low-light conditions, demonstrating the effectiveness and robustness of Mono3DLane.

Method	F1(%)↑	Prec.(%)↑	Rec.(%)↑	CD Error(m)↓
3D-LaneNet	44.73	61.46	35.16	0.127
Gen-LaneNet	45.59	63.95	35.42	0.121
SALAD	64.07	75.90	55.42	0.098
PersFormer	74.33	80.30	69.18	0.074
Anchor3DLane	74.87	80.85	69.71	0.060
Mono3DLane	77.13	81.66	71.25	0.055
Mono3DLane†	**77.84**	**81.95**	**71.72**	**0.051**

Table 3: Comparison with state-of-the-art methods on ONCE-3DLanes validation set. We report the F-score, precision (Prec.), recall (Rec.), and chamfer distance (CD) errors.

Performance on Once-3DLanes. The comparison results for the Once-3DLanes are presented in Tab. 3. Mono3DLane achieves a new state-of-the-art performance with a 77.84% F1 score. Notably, the highest precision and recall scores and the lowest CD error indicate the effectiveness of our coarse-to-fine lane prediction strategy, indicating its potential to be applied in real-world scenarios.

Ablation Study

To validate the effectiveness of our design modules, we conducted several ablation experiments on the OpenLane validation set. All experiments are based on the ResNet-18 backbone. Further details concerning module structure, and extra ablation studies (including 3D position embedding) are provided in our Appendix.

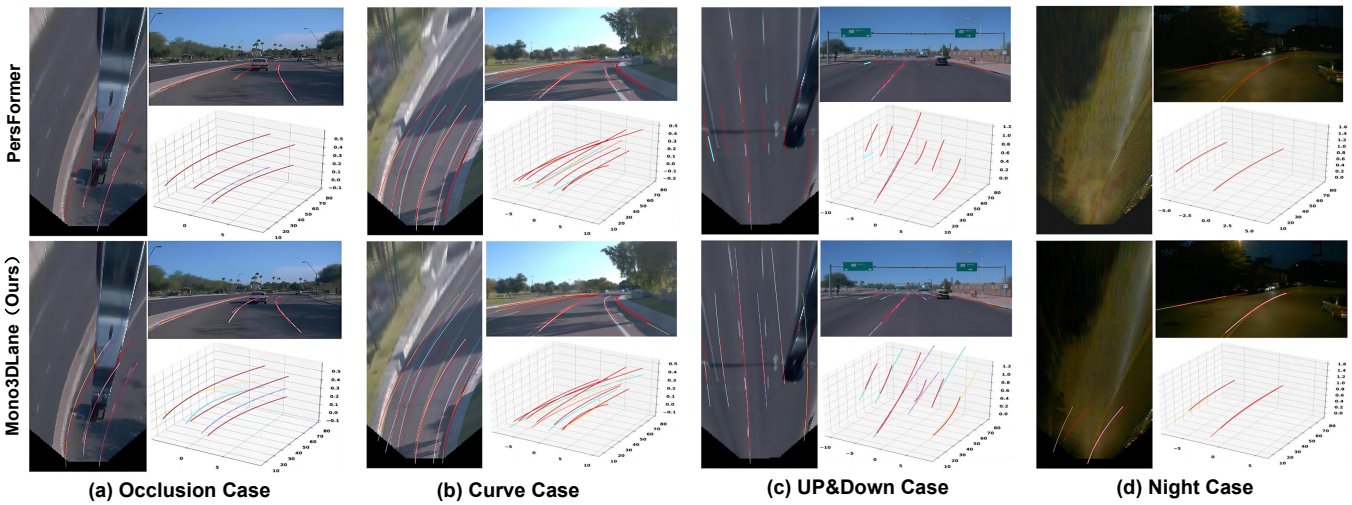

Figure 5: Qualitative evaluation on Openlane validation set. (a) → (d) denotes four challenging scenarios. Results are presented in front-view, BEV and 3D space. Here, different color lanes indicate specific categories and the red lanes indicate ground truth.

VCM	Attention (Our)	F1(%)↑	X error (m)↓		Z error (m)↓	
			near	*far*	*near*	*far*
		48.6	0.504	0.542	0.109	0.175
✓		50.1	0.311	0.358	0.102	0.129
	✓	55.2	0.342	0.381	0.108	0.131
✓	✓	**56.3**	**0.266**	**0.285**	**0.083**	**0.112**

Table 4: Effects of each component in our method.

Overall Ablation Study. Tab. 4 shows the overall ablation study results. We progressively integrate the VCM and Spatial Hierarchy-aware Attention into the baseline while maintaining a consistent refinement strategy. The VCM notably reduces detection errors and Spatial Hierarchy-aware Attention significantly boosts the F1, confirming its effectiveness in capturing spatial information and lane-aware features.

Effect of Refinement Sequence. The key and value pairs dynamically collect and update queries across aligned feature map layers at varying depths, using a descending acquisition sequence to refine parameters. The outcomes of different refinement sequences are detailed in Tab.5. These results show that the sequence $\mathbf{F_0} \rightarrow \mathbf{F_1} \rightarrow \mathbf{F_2}$ more effectively constructs the lane representation, aligning with the coarse-to-fine prediction methodology.

2D Prior Anchors vs. Dense 3D Anchors. We compare the impact of the proposed 2D prior anchors with dense 3D anchors (Huang et al. 2023), as shown in Tab. 6. The 3D anchor definition is complex and collects redundant features, offering minimal performance gains. In contrast, our method, which initializes only 192 prior anchors, is consistent with realistic lane distribution, achieving commendable performance while enhancing inference speed.

Effects of Stride Size. The ablation experiments with different stride sizes for Row-Column Stripe Attention, as shown in Tab. 6, reveal that reducing the stride size and increasing the stripe density enhances performance. A stride size of 8 strikes an optimal balance between performance and speed.

Settings	F1(%)↑	Night↑	Curve↑	X err.far(m)↓	Z err.far(m)↓
F_0	52.8	48.2	59.5	0.387	0.151
F_1	52.2	48.3	59.2	0.383	0.144
F_2	52.5	48.2	59.1	0.381	0.142
$F_0 \rightarrow F_0$	53.2	48.4	59.7	0.343	0.138
$F_0 \rightarrow F_0 \rightarrow F_0$	53.4	49.3	60.1	0.331	0.133
$F_2 \rightarrow F_1 \rightarrow F_0$	53.8	50.1	60.4	0.324	0.135
$F_0 \rightarrow F_1 \rightarrow F_2$	**56.3**	**52.9**	**61.5**	**0.285**	**0.112**

Table 5: Ablation studies of different refinement sequences. F_i is the aligned feature map layer.

Method	Anchors	Stride	F1(%)↑	X err.far(m)↓	Z err.far(m)↓	FPS↑
w/o PA	4431	10	53.1	0.312	0.129	55
		8	53.8	0.306	0.123	41
w/ PA	192	10	55.2	0.297	0.125	**91**
		8	56.3	**0.285**	0.112	82
		6	**56.4**	0.286	**0.110**	74

Table 6: Ablation study on prior anchors and effect of the stride size. PA: 2D prior anchors

Conclusion

In this work, we introduce Mono3DLane, an innovative framework for 3D lane detection that efficiently predicts 3D lanes directly from monocular images. The Spatial Hierarchy-aware Attention plays a crucial role in extracting lane-aware features and aggregating 3D spatial information. By employing 2D prior anchors, our method offers a more intuitive and efficient approach to generating 3D lanes. To overcome geometric distortions, we incorporate a lightweight view context model that aligns monocular visual features and refines parameters using a coarse-to-fine strategy, substantially improving lane positioning accuracy. Experimental results demonstrate that our method outperforms existing approaches on two 3D lane detection benchmarks with a streamlined architecture.

References

Bai, Y.; Chen, Z.; Fu, Z.; Peng, L.; Liang, P.; and Cheng, E. 2023. Curveformer: 3d lane detection by curve propagation with curve queries and attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 7062–7068. IEEE.

Bai, Y.; Chen, Z.; Liang, P.; and Cheng, E. 2024. CurveFormer++: 3D Lane Detection by Curve Propagation with Temporal Curve Queries and Attention. *arXiv preprint arXiv:2402.06423*.

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

Chen, L.; Sima, C.; Li, Y.; Zheng, Z.; Xu, J.; Geng, X.; Li, H.; He, C.; Shi, J.; Qiao, Y.; et al. 2022. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*, 550–567. Springer.

Chen, Z.; Yang, C.; Li, Q.; Zhao, F.; Zha, Z.-J.; and Wu, F. 2021. Disentangle your dense object detector. In *Proceedings of the 29th ACM international conference on multimedia*, 4939–4948.

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34: 9355–9366.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 764–773.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Efrat, N.; Bluvstein, M.; Garnett, N.; Levi, D.; Oron, S.; and Shlomo, B. E. 2020a. Semi-local 3d lane detection and uncertainty estimation. *arXiv preprint arXiv:2003.05257*.

Efrat, N.; Bluvstein, M.; Oron, S.; Levi, D.; Garnett, N.; and Shlomo, B. E. 2020b. 3d-lanenet+: Anchor free lane detection using a semi-local representation.

Garnett, N.; Cohen, R.; Pe'er, T.; Lahav, R.; and Levi, D. 2019. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2921–2930.

Guo, Y.; Chen, G.; Zhao, P.; Zhang, W.; Miao, J.; Wang, J.; and Choe, T. E. 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*, 666–681. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, S.; Shen, Z.; Huang, Z.; Ding, Z.-h.; Dai, J.; Han, J.; Wang, N.; and Liu, S. 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.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, 5156–5165. PMLR.

Li, Y.; Fan, Y.; Xiang, X.; Demandolx, D.; Ranjan, R.; Timofte, R.; and Van Gool, L. 2023. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18278–18289.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022a. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, Q.; Wang, T.; Zhang, X.; and Sun, J. 2022b. PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images. *arXiv preprint arXiv:2206.01256*.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. 2022c. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12009–12019.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Luo, Y.; Yan, X.; Zheng, C.; Zheng, C.; Mei, S.; Kun, T.; Cui, S.; and Li, Z. 2022. Mˆ 2-3DLaneNet: Multi-Modal 3D Lane Detection. *arXiv preprint arXiv:2209.05996*.

Luo, Y.; Zheng, C.; Yan, X.; Kun, T.; Zheng, C.; Cui, S.; and Li, Z. 2023. LATR: 3D Lane Detection from Monocular Images with Transformer. *arXiv preprint arXiv:2308.04583*.

Qu, Z.; Jin, H.; Zhou, Y.; Yang, Z.; and Zhang, W. 2021. Focus on local: Detecting lane marker from bottom up via key point. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14122–14130.

Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; and Shlens, J. 2019. Stand-alone self-attention in vision models. *Advances in neural information processing systems*, 32.

Tabelini, L.; Berriel, R.; Paixao, T. M.; Badue, C.; De Souza, A. F.; and Oliveira-Santos, T. 2021. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 294–302.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Li, T.; Li, Y.; Chen, L.; Sima, C.; Liu, Z.; Wang, Y.; Jiang, S.; Jia, P.; Wang, B.; Wen, F.; Xu, H.; Luo, P.; Yan, J.; Zhang, W.; and Li, H. 2023a. OpenLane-V2: A Topology Reasoning Benchmark for Scene Understanding in Autonomous Driving. *arXiv preprint arXiv:2304.10440*.

Wang, R.; Qin, J.; Li, K.; Li, Y.; Cao, D.; and Xu, J. 2023b. BEV-LaneDet: An Efficient 3D Lane Detection Based on Virtual Camera via Key-Points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1002–1011.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Williams, K. R.; Schlossman, R.; Whitten, D.; Ingram, J.; Musuvathy, S.; Pagan, J.; Williams, K. A.; Green, S.; Patel, A.; Mazumdar, A.; et al. 2022. Trajectory planning with deep reinforcement learning in high-level action spaces. *IEEE Transactions on Aerospace and Electronic Systems*.

Yan, F.; Nie, M.; Cai, X.; Han, J.; Xu, H.; Yang, Z.; Ye, C.; Fu, Y.; Mi, M. B.; and Zhang, L. 2022. Once-3dlanes: Building monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17143–17152.

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

Zhuang, J.; Qin, Z.; Yu, H.; and Chen, X. 2023. Task-Specific Context Decoupling for Object Detection. *arXiv preprint arXiv:2303.01047*.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (yes)

Does this paper make theoretical contributions? (yes)
If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (yes)
- All novel claims are stated formally (e.g., in theorem statements). (yes)
- Proofs of all novel claims are included. (yes)
- Proof sketches or intuitions are given for complex and/or novel results. (yes)
- Appropriate citations to theoretical tools used are given. (yes)
- All theoretical claims are demonstrated empirically to hold. (yes)
- All experimental code used to eliminate or disprove claims is included. (yes)

Does this paper rely on one or more datasets? (yes)
If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets (yes)
- All novel datasets introduced in this paper are included in a data appendix. (NA)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (NA)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (NA)

Does this paper include computational experiments? (yes)
If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (yes).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (yes)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)

- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (yes)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes)
- This paper states the number of algorithm runs used to compute each reported result. (yes)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (yes)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)