RETR: A Structure-Preserving RGB-Event Transformer for Robust 3D Lane Detection

Jingtao Dong, Hao Zhuang, Hao Yang, Liyuan Pan†, Wei Liang
Beijing Institute of Technology, Beijing, China
{jingtao.dong, hao.zhuang}@bit.edu.cn
{hao.yang, liyuan.pan, liangwei}@bit.edu.cn

Abstract

Robust 3D lane detection requires accurate metric road geometry recovery from visual inputs, yet conventional RGB-based approaches suffer from fragile lane evidence under adverse illumination, motion blur, and long-range perspective compression. Event cameras offer complementary high-temporal-resolution, high-dynamic-range structural cues that are robust to these challenges, but their sparse, motion-dependent responses cannot be directly fused into consistent 3D geometry. In this paper, we present the first attempt to introduce event cameras into 3D lane detection. To enable systematic research, we first build two multimodal benchmarks with metric 3D lane annotations: DSEC-3DLD (real-world sequences) and Ev-OpenLane (large-scale simulated sequences). We further propose RETR, a structure preserving transformer that converts complementary RGB and event observations into coherent 3D lane geometry. RETR first aligns modality consistent evidence through Reciprocal Context Flow Fusion, then preserves thin and uncertain lane structures with Uncertainty Aware Structural Consolidation, and finally decodes ordered lane hypotheses using a Geometry State Decoder with proposal conditioned initialization, reference conditioned query evolution, and diversified geometric inquiry. Extensive experiments show that RETR achieves state-of-the-art performance on both benchmarks, especially showing strong improvements in challenging lighting and complex road geometry scenarios. Code and dataset will be available at `https://github.com/duskswell/RETR`.

1 Introduction

Reliable 3D lane detection is a core perception task for autonomous driving. It aims to recover metric lane geometry, including lateral position, longitudinal layout, and elevation, together with lane visibility and continuity. Recent methods have advanced this task through stronger geometric representations [14], camera-to-ground transformations and BEV construction [7, 62], anchor-based lane modeling [19], and query-based decoding with transformer-based long-range reasoning [35, 33]. However, their predictions still depend largely on the lane evidence available in RGB frames. This evidence is often fragile in real driving scenes: lane markings are thin, low-contrast, and strongly compressed by perspective, especially at long range. Under adverse illumination and motion conditions, lane evidence can become weak or structurally incomplete, forcing the model to recover 3D geometry from unreliable visual cues.

The broader challenge of perception from degraded observations has motivated research on adverse-weather removal [60], defocus deblurring [59], and fog-invariant action recognition [27]. For metric 3D lane detection, however, improving visual appearance must also preserve the geometric evidence needed for localization. A recent study of GPT-Image for image restoration [61] reports that

†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

perceptually plausible outputs can deviate from the original image geometry and perspective. This distinction motivates our focus on acquiring complementary structural observations and preserving their geometric information for 3D lane recovery.

Event cameras provide a promising complementary modality. Different from frame-based cameras that capture intensity at fixed intervals, event cameras asynchronously record local brightness changes with microsecond-level temporal resolution and extremely high dynamic range. They produce boundary-aware responses that remain stable under fast motion, over-exposure, under-exposure, and low-contrast conditions [6, 25, 37]. These responses are particularly relevant to lane perception, where distant markings, road boundaries, and weak contrast transitions may disappear in conventional frames. Yet event signals are sparse, motion-dependent and structurally different from RGB features [11, 5]. Therefore, a simple fusion strategy, such as feature concatenation, can easily dilute useful event responses or amplify unreliable activations, leading to geometrically inconsistent lane predictions.

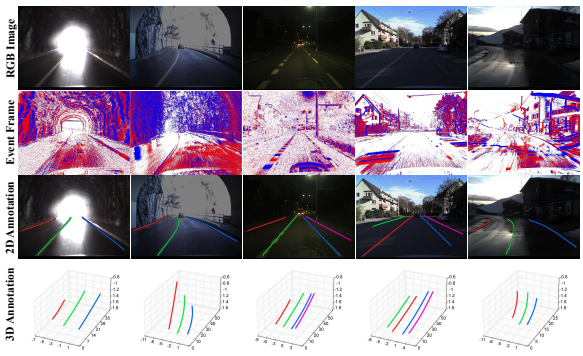


Figure 1: Overview of our DSEC-3DLD dataset. Each column presents one synchronized driving sample, including the RGB image, event frame, 2D lane annotation, and metric 3D lane annotation. The selected examples cover challenging real-world conditions such as tunnels, low illumination and strong exposure changes.

These observations motivate a structure preserving view of 3D lane detection with RGB and event inputs. The goal is not merely to introduce an additional event branch, but to transform complementary sensing evidence into coherent metric lane geometry. This requires a full pipeline that aligns modality specific cues, preserves weak but important lane structures during feature aggregation, and decodes the resulting representation as ordered 3D lane hypotheses. To this end, we propose RETR, a structure preserving transformer for robust 3D lane detection. Rather than treating RGB and event inputs as features to be simply merged, our approach organizes the prediction process around a progressive transformation from complementary sensing evidence to metric lane geometry. It promotes modality consistent responses and bidirectional context exchange across RGB and event features through Reciprocal Context Flow Fusion (RCF), while Uncertainty Aware Structural Consolidation (UASC) retains thin, uncertain, and event induced lane evidence during multi scale aggregation. The consolidated representation is further decoded by the Geometry State Decoder (GSD), which models lanes as ordered geometric entities using proposal conditioned initialization, reference conditioned query evolution, and diversified geometric inquiry. In this way, our approach turns complementary RGB and event observations into stable 3D lane geometry.

A key obstacle to studying this setting is the lack of dedicated benchmarks. Existing 3D lane datasets are primarily built with conventional frame cameras, making them insufficient for evaluating the role of event cameras in 3D lane detection. To the best of our knowledge, we are the first to introduce event cameras into 3D lane detection. We further establish two multimodal benchmarks with image and event inputs: DSEC-3DLD, a real event camera dataset with synchronized RGB and event inputs and metric 3D lane annotations, with representative examples shown in Fig. 1, and Ev-OpenLane, a large scale benchmark with simulated event information for scalable training and evaluation. Experiments on both benchmarks show that RETR achieves state-of-the-art performance, validating the effectiveness of our structure preserving design and demonstrating the importance of event cameras for robust 3D lane detection.

Our contributions are summarized as follows:

- We introduce event-based 3D lane detection and establish two multimodal benchmarks, DSEC-3DLD and Ev-OpenLane, providing synchronized image and event inputs together with metric 3D lane annotations. To the best of our knowledge, this is the first work to study event cameras for 3D lane detection.

- We propose RETR, a structure preserving transformer that transforms complementary RGB and event observations into coherent 3D lane geometry.
- We develop a reliability-to-geometry pipeline that integrates RCF, UASC and GSD to align cross-modal lane evidence, preserve sparse structural cues, and decode coherent 3D lanes through proposal guided geometry state evolution.

2 Related Work

3D lane detection. Advances in deep learning have substantially expanded visual understanding capabilities [29, 15–17, 28]. In 3D lane detection, recent methods have improved geometric reasoning through explicit spatial transformation or implicit structure modeling. Explicit approaches project image features into bird's-eye-view space using inverse perspective mapping or depth-aware lifting [7, 62, 14, 24, 46, 47, 13, 36, 34], sometimes further incorporating depth cues, geometric constraints, height-aware representations, or slope modeling [58, 14, 45, 44]. In contrast, implicit methods directly infer 3D lane structures with queries [30, 35, 33], 3D anchors [19], or BEV keypoint grouping [55, 40]. Despite these advances, existing methods still derive their lane evidence primarily from frame-based intensity images, making them vulnerable to severe illumination degradation such as overexposure, nighttime underexposure, headlight glare, and abrupt lighting transitions.

Event-based vision for robust perception. Event cameras asynchronously capture intensity changes with microsecond latency and high dynamic range, making them attractive for perception in challenging conditions. Their applications, datasets, and practical challenges in automotive sensing are reviewed by Shariff et al. [49]. They have been successfully applied to object detection [38, 65, 66, 26], segmentation [1, 38, 69, 50], optical flow [41, 10, 32, 68], and image restoration [42, 51, 21, 64]. In autonomous driving, event-based datasets such as DSEC [12] and MVSEC [67] provide synchronized event, frame, and geometric measurements for downstream multimodal perception. NSAVP [4] further combines event, thermal, monochrome, and RGB observations and benchmarks place recognition under varied lighting conditions. More recently, event signals have also been explored for 3D object detection [9, 22]. For lane perception, DET [8] provides a high-resolution event-based lane extraction benchmark with pixel-level annotations. While DET addresses image-plane lane extraction, our work targets metric 3D lane geometry from RGB-event observations.

Lane datasets. Benchmark development has similarly progressed from 2D lane datasets such as TuSimple [54], CULane [43], and LLAMAS [3] to 3D benchmarks such as Apollo-Sim-3D [14], ONCE-3DLanes [58], and OpenLane [7]. These datasets have greatly advanced geometric lane perception, but they are primarily built on conventional frame cameras. As a result, they provide limited support for studying multimodal robustness under adverse conditions. Our work is motivated by this gap: we study RGB-event fusion for 3D lane detection and build a benchmark setting tailored to robust 3D road topology understanding in challenging environments.

3 Method

Overview. Our approach builds on LATR [35] with a dual-stream design for RGB and event inputs. Given a synchronized RGB image and an event representation aggregated within a short temporal window, two modality-specific encoders extract hierarchical features $\{\mathbf{F}_r^l\}_{l=1}^L$ and $\{\mathbf{F}_e^l\}_{l=1}^L$, where l indexes the feature stage and L is the number of selected stages. RETR first aligns and exchanges RGB and event features through RCF. The resulting multi-scale features are then consolidated by UASC to preserve weak lane structures before decoding. Finally, GSD transforms the consolidated representation into ordered 3D lane hypotheses, including lane existence, anchor-wise geometry, and visibility.

3.1 Reciprocal Context Flow Fusion

Let $\mathbf{F}_r^l, \mathbf{F}_e^l \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$ denote the RGB and event features at stage l, where B is the batch size, C_l is the channel dimension, and H_l, W_l are the spatial resolution. RGB features encode dense appearance and semantic context, while event features capture temporal changes and high-frequency structures. Because these modalities respond differently to lane evidence, naive concatenation can mix redundant or misaligned responses. RCF therefore starts by mapping both modalities into a

shared stage-wise feature space before co-activation enhancement and bidirectional context exchange:

$$\bar{\mathbf{F}}_r^l = \psi_r^l(\mathbf{F}_r^l), \qquad \bar{\mathbf{F}}_e^l = \psi_e^l(\mathbf{F}_e^l), \tag{1}$$

where ψ_r^l and ψ_e^l are 1×1 convolutional projections. A co-activation response is then computed by element-wise multiplication:

$$\mathbf{M}^l = \bar{\mathbf{F}}_r^l \odot \bar{\mathbf{F}}_e^l. \tag{2}$$

The response $\mathbf{M}^l$ emphasizes spatial locations that are supported by both RGB appearance and event structures. Rather than using it as a hard mask, we inject it into both streams by residual addition:

$$\mathbf{H}_r^l = \bar{\mathbf{F}}_r^l + \mathbf{M}^l, \qquad \mathbf{H}_e^l = \bar{\mathbf{F}}_e^l + \mathbf{M}^l. \tag{3}$$

This produces co-activated RGB and event features while keeping the two modalities distinguishable.

To exchange complementary information, RCF uses a Cross-Attention Block (CAB). Given a target feature $\mathbf{X}$ and a guidance feature $\mathbf{Y}$, CAB is implemented as

$$\mathrm{CAB}(\mathbf{X} \mid \mathbf{Y}) = \mathbf{X} + \rho\left(\left[\mathbf{X}, \mathrm{Unflat}\left(\mathrm{softmax}\left(\frac{(W_q\mathbf{X})(W_k\mathbf{Y})^\top}{\sqrt{d}}\right) W_v\mathbf{Y}\right)\right]\right). \tag{4}$$

Before attention, $\mathbf{X}$ and $\mathbf{Y}$ are flattened over spatial positions. The target feature provides the query through W_q, while the guidance feature provides keys and values through W_k and W_v. The operator $\mathrm{Unflat}(\cdot)$ restores the spatial layout after attention, d is the key dimension, and $\rho(\cdot)$ is a Conv-BN-ReLU fusion block. CAB therefore transports context from the guidance modality and fuses it back into the target modality.

RCF applies CAB in both directions, generating an event-guided RGB feature $\mathbf{Y}_r^l = \mathrm{CAB}(\mathbf{H}_r^l \mid \mathbf{H}_e^l)$ and an RGB-guided event feature $\mathbf{Y}_e^l = \mathrm{CAB}(\mathbf{H}_e^l \mid \mathbf{H}_r^l)$. The two features are concatenated along the channel dimension and compressed by a 1×1 Conv-BN-ReLU projection to obtain the fused stage feature $\mathbf{Z}^l$. The fused hierarchy $\{\mathbf{Z}^l\}_{l=1}^L$ is then passed to UASC.

3.2 Uncertainty-Aware Structural Consolidation

Given the fused hierarchy $\{\mathbf{Z}^l\}_{l=1}^L$, UASC produces a compact representation while preserving thin and uncertain lane evidence. Each fused stage is first projected to a common channel dimension and resized to a shared spatial resolution:

$$\hat{\mathbf{Z}}^l = \mathrm{Align}\left(\phi_l(\mathbf{Z}^l)\right), \qquad l = 1, \ldots, L. \tag{5}$$

The projection ϕ_l is implemented by a 1×1 convolution, and $\mathrm{Align}(\cdot)$ denotes bilinear resizing. The aligned features are concatenated along the channel dimension to form the aligned multi-scale feature $\mathbf{Z}_a$.

UASC contains two branches with different responsibilities. The structural branch, denoted as STR, preserves lane geometry from both all-scale evidence and high-resolution details. It first fuses all aligned stages to obtain a scale-complete representation and also processes the high-resolution stages to retain local lane structures:

$$\mathbf{F}_{\mathrm{STR}} = \mathcal{B}_{\mathrm{STR}}\left(\left[\mathcal{B}_{\mathrm{ms}}(\mathbf{Z}_a), \mathcal{B}_{\mathrm{hr}}\left([\hat{\mathbf{Z}}^1, \ldots, \hat{\mathbf{Z}}^{L_h}]\right)\right]\right). \tag{6}$$

$\mathcal{B}_{\mathrm{ms}}$ fuses all aligned scales using Conv-BN-ReLU layers, while $\mathcal{B}_{\mathrm{hr}}$ processes the first L_h high-resolution stages. The output $\mathbf{F}_{\mathrm{STR}}$ therefore keeps global scale coverage and fine spatial details, which is important for thin lane markings and sparse event responses.

The context branch, denoted as CTX, complements STR by modeling road-level layout and long-range lane continuity. It uses parallel dilated convolutions with different receptive fields:

$$\mathbf{F}_{\mathrm{CTX}} = \mathcal{B}_{\mathrm{CTX}}\left([\mathcal{D}_1(\mathbf{Z}_a), \mathcal{D}_2(\mathbf{Z}_a), \mathcal{D}_5(\mathbf{Z}_a), \mathcal{D}_9(\mathbf{Z}_a)]\right). \tag{7}$$

Each $\mathcal{D}_s$ is a 3×3 dilated Conv-BN-ReLU block with dilation rate s, and $\mathcal{B}_{\mathrm{CTX}}$ integrates the multi-range responses. By aggregating context at multiple dilation scales, CTX provides layout-level continuity cues for weak or partially missing lane evidence while keeping the spatial resolution unchanged.

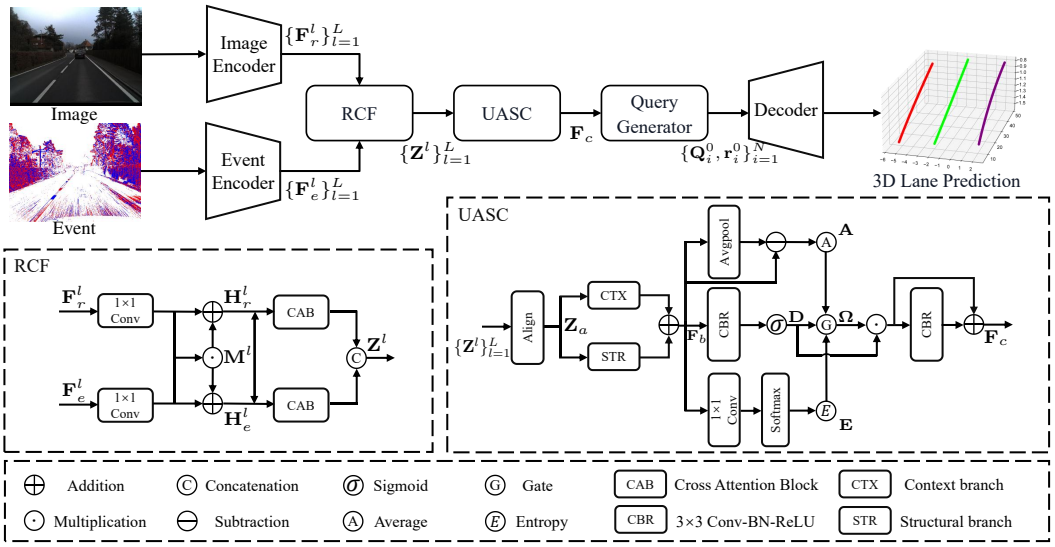


Figure 2: **Overall architecture of RETR.** RETR converts synchronized RGB and event observations into metric 3D lane geometry through a structure-preserving RGB-event pipeline. Modality-specific encoders first extract multi-scale RGB and event features $\{\mathbf{F}_r^l\}_{l=1}^L$ and $\{\mathbf{F}_e^l\}_{l=1}^L$. RCF aligns both modalities at each scale by using a co-activation response $\mathbf{M}^l$ to emphasize jointly supported lane evidence and bidirectional Cross Attention Blocks (CAB) to exchange complementary context, producing fused features $\{\mathbf{Z}^l\}_{l=1}^L$. UASC aggregates the fused hierarchy into $\mathbf{F}_c$ while preserving thin and uncertain lane structures. It combines context and structural branches and gates detail refinement using high-frequency response $\mathbf{A}$, learned detail cue $\mathbf{D}$, and entropy-based uncertainty $\mathbf{E}$. The query generator and geometry decoder finally predict ordered 3D lane hypotheses, including lane existence, 3D coordinates, and visibility.

The STR and CTX outputs are then coupled to form a structure-context base feature $\mathbf{F}_b$. This coupling combines the detail-preserving response from STR with the layout-aware continuity response from CTX, providing a stronger base representation for the subsequent uncertainty-guided refinement.

To identify regions requiring structural refinement, UASC computes three cues from $\mathbf{F}_b$:

$$\begin{aligned} \mathbf{E} &= \mathcal{H}\left(\operatorname{softmax}\left(h_c(\mathbf{F}_b)\right)\right), \\ \mathbf{A} &= \operatorname{Avg}_c\left(\left|\mathbf{F}_b - \operatorname{AvgPool}(\mathbf{F}_b)\right|\right), \\ \mathbf{D} &= \sigma\left(h_d(\mathbf{F}_b)\right). \end{aligned} \tag{8}$$

The entropy cue $\mathbf{E}$ is computed from the channel-wise probability distribution predicted by the classification projection h_c, the high-frequency cue $\mathbf{A}$ measures the channel-averaged residual between $\mathbf{F}_b$ and its spatially averaged version, and the learned detail cue $\mathbf{D}$ is predicted by the detail head h_d. These cues are then fused into a refinement gate:

$$\mathbf{\Omega} = \sigma\left(h_g\left([\mathbf{E}, \mathbf{A}, \mathbf{D}]\right)\right). \tag{9}$$

The gating head h_g is implemented with lightweight convolutional layers. The gate $\mathbf{\Omega}$ indicates the spatial regions where the base feature requires additional structural refinement. The gated detail cue is first converted into a structural refinement feature:

$$\mathbf{F}_{\text{ref}} = \mathcal{B}_{\text{ref}}\left([\mathbf{F}_b, \mathbf{\Omega} \odot \mathbf{D}]\right). \tag{10}$$

Here, $\mathbf{F}_{\text{ref}}$ is a refinement feature predicted from the base feature and the gated detail cue.

$$\mathbf{F}_c = \mathcal{B}_o\left([\mathbf{F}_b, \mathbf{F}_b + \mathbf{\Omega} \odot \mathbf{F}_{\text{ref}}]\right). \tag{11}$$

The final consolidated feature $\mathbf{F}_c$ is obtained by residually injecting $\mathbf{F}_{\text{ref}}$ into $\mathbf{F}_b$ through $\mathbf{\Omega}$, followed by the output projection $\mathcal{B}_o$. This residual design prevents uncertain detail cues from replacing the base representation and only enhances regions selected by the refinement gate.

3.3 Geometry State Decoder

Given the consolidated feature $\mathbf{F}_c$, GSD transforms the fused lane evidence into ordered 3D lane hypotheses. As shown in Fig. 2, the query generator maps $\mathbf{F}_c$ to N initial lane states $\{\mathbf{Q}_i^0, \mathbf{r}_i^0\}_{i=1}^N$ by first extracting proposal descriptors $\{\mathbf{s}_i\}_{i=1}^N$ and then expanding each descriptor into ordered anchor tokens. For proposal i, $\mathbf{Q}_i^0 \in \mathbb{R}^{A\times C}$ is an ordered anchor-token sequence associated with predefined longitudinal positions, and $\mathbf{r}_i^0 \in [0,1]^{A\times 2}$ provides the initial lateral-height reference for these anchors. The decoder then evolves these proposal-level lane states to predict lane existence, 3D geometry, and visibility:

$$\mathbf{Q}_i^0 = \mathrm{MLP}_q\left(\mathbf{s}_i \oplus \mathbf{p}_{1:A}\right), \qquad \mathbf{r}_i^0 = \sigma\left(\mathrm{MLP}_r\left(\mathbf{Q}_i^0\right)\right). \tag{12}$$

Here, $\mathbf{s}_i \in \mathbb{R}^C$ is the proposal descriptor extracted from $\mathbf{F}_c$, $\mathbf{p}_{1:A}$ are learned anchor embeddings associated with predefined longitudinal positions, and $\oplus$ denotes anchor-wise broadcasting. The initialized token sequence $\mathbf{Q}_i^0$ represents proposal i over ordered lane anchors, while $\mathbf{r}_i^0$ provides its initial normalized 3D reference state. In this way, each proposal is initialized as a structured lane state instead of an independent query vector.

At decoder layer k, GSD conditions each anchor token on its current 3D reference state. For anchor a of proposal i, the current reference state is converted into a metric 3D point and used to construct a geometry-conditioned positional embedding:

$$\mathbf{u}_{i,a}^k = [x(r_{i,a,x}^k), y_a, z(r_{i,a,z}^k)], \qquad \tilde{\mathbf{p}}_{i,a}^k = \mathbf{p}_a + \beta\Phi\left(\mathbf{u}_{i,a}^k\right). \tag{13}$$

y_a is the fixed longitudinal position of anchor a, while $x(\cdot)$ and $z(\cdot)$ map the normalized reference values to metric lateral and height coordinates. $\Phi(\cdot)$ encodes the current 3D reference point, and β controls the strength of the geometry-conditioned positional offset. This design makes the query position evolve with the predicted lane geometry instead of relying only on fixed learned anchor embeddings.

To improve geometry-aware reasoning, each decoder layer contains M independent inquiry branches:

$$(\mathbf{Q}_m^k, \mathbf{h}_m^k) = \mathcal{G}_m^k\left(\mathbf{Q}^{k-1} + \tilde{\mathbf{P}}^k, \mathbf{F}_c, \mathbf{r}^k\right), \qquad m = 1, \ldots, M. \tag{14}$$

$\tilde{\mathbf{P}}^k$ collects all dynamic positional embeddings $\tilde{\mathbf{p}}_{i,a}^k$. Each branch $\mathcal{G}_m^k$ contains anchor self-attention, reference-conditioned cross-attention to $\mathbf{F}_c$, and a feed-forward network. $\mathbf{Q}_m^k \in \mathbb{R}^{N\times A\times C}$ contains anchor-level states, while $\mathbf{h}_m^k \in \mathbb{R}^{N\times C}$ contains lane-level descriptors obtained from the anchor tokens. The multiple branches provide diverse geometric inquiries over the same lane proposal, which helps distinguish nearby lanes and maintain coherent lane structures.

The branch outputs are projected and normalized to update the anchor-level state $\mathbf{Q}^k$ and lane-level state $\mathbf{h}^k$, which are used for reference refinement and final lane prediction. After K layers, GSD predicts lane existence, anchor-wise 3D geometry, and visibility while maintaining proposal identity and ordered lane structure throughout decoding.

3.4 Loss

Following LATR [35], we train RETR with a weighted combination of a 3D lane loss, a lane proposal loss, and an auxiliary 2D lane loss:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{3D}} + \lambda_{\mathrm{prop}}\mathcal{L}_{\mathrm{prop}} + \lambda_{\mathrm{2D}}\mathcal{L}_{\mathrm{2D}}, \tag{15}$$

where $\lambda_{\mathrm{prop}} = 5.0$ and $\lambda_{\mathrm{2D}} = 1.0$.

The 3D lane loss $\mathcal{L}_{\mathrm{3D}}$ supervises lane existence, anchor-wise lateral and height coordinates, and visibility. It also includes a projection consistency term. The proposal loss $\mathcal{L}_{\mathrm{prop}}$ supervises lane proposal generation using classification, mask, Dice, and objectness terms. The auxiliary 2D loss $\mathcal{L}_{\mathrm{2D}}$ provides image-plane supervision through lane segmentation, overlap, and instance embedding.

4 Experiments

This work makes the first attempt to study 3D lane detection with event cameras. To validate the effectiveness of our method, we introduce two complementary event-based 3D lane datasets: DSEC-3DLD with real events and Ev-OpenLane with simulated events. Details of the annotation and dataset generation pipelines are provided in the supplementary material.

Table 1: Comparison with state-of-the-art 3D lane detection methods on DSEC-3DLD. We report overall F1-score (F1), precision (P), recall (R), centerline distance error (CD), and scenario-wise F1-score . Scenario columns denote Straight (Str.), Curved (Cur.), Intersection (Int.), Merge/Split (M&S), Normal (Nor.), Overexposed (Ove.), Underexposed (Und.), and Up/Down road geometry (U/D). Best and second-best results are shown in **bold** and underlined, respectively.

Method	Overall				Scenario-wise F1-Score (%)							
	F1↑	P↑	R↑	CD↓	Str.↑	Cur.↑	Int.↑	M&S↑	Nor.↑	Ove.↑	Und.↑	U/D↑
PersFormer [7]	50.1	48.7	51.5	0.127	51.5	48.3	48.8	58.9	53.3	31.3	41.8	34.5
SparsePers [62]	50.2	48.1	52.3	0.134	51.6	48.8	46.1	59.2	52.2	35.2	45.4	45.6
Anchor3DLane [19]	54.7	57.9	51.8	0.127	56.6	53.8	51.2	50.7	57.3	28.3	49.0	45.7
3DLaneDT [56]	53.7	55.2	52.2	0.130	54.1	54.2	51.2	53.3	56.2	24.9	48.8	43.5
Anchor3DLanePP [20]	61.8	67.1	57.4	0.121	63.8	59.9	59.3	65.6	64.4	39.7	56.2	48.6
LATR [35]	69.2	71.1	67.3	0.107	70.7	67.2	68.9	70.7	72.3	49.0	60.7	61.9
DV-3DLane [33]	69.4	72.5	66.6	0.105	70.5	68.2	68.9	72.9	72.7	52.6	59.0	58.4
MRDALane [39]	69.3	70.5	68.0	0.110	70.4	69.3	65.9	69.4	72.1	56.7	60.8	56.8
RETR (ours)	**75.6**	**76.3**	**75.0**	**0.094**	**75.8**	**76.0**	**73.7**	**76.8**	**77.4**	**59.6**	**72.1**	**71.7**

Table 2: Comparison with state-of-the-art 3D lane detection methods on Ev-OpenLane. We report F1-score (%), near/far X-error, and near/far Z-error under the standard OpenLane evaluation protocol. Higher F-score is better, while lower X/Z errors indicate more accurate 3D localization. Best and second-best results are shown in **bold** and underlined, respectively.

Method	F1-score(%)↑	X-error (near)↓	X-error (far)↓	Z-error (near)↓	Z-error (far)↓
PersFormer [7]	50.5	0.485	0.553	0.364	0.431
Anchor3DLane [19]	53.1	0.300	0.311	0.103	0.139
BEVLaneDet [55]	58.4	0.309	0.659	0.244	0.631
LaneCPP [47]	60.3	0.264	0.310	0.077	0.117
LATR [35]	61.9	0.219	0.259	0.075	0.104
HeightLane [45]	62.7	0.240	0.266	0.116	0.165
RFTR [23]	61.8	0.341	0.450	0.073	0.107
PVALane [63]	62.7	0.232	0.259	0.092	0.118
SC-Lane [44]	64.3	0.227	0.251	0.088	0.128
RETR (ours)	**66.7**	**0.203**	**0.230**	**0.071**	**0.101**

4.1 Datasets Establishment and Evaluation Metrics

Ev-OpenLane is built upon OpenLane [7], a comprehensive large-scale 3D lane detection benchmark based on the Waymo dataset [52]. It contains 1000 segments and 200K frames captured under diverse weather, terrain, and brightness conditions at a resolution of 1280×1920. We augment the original image frames with synchronized synthetic event data, while preserving 880K lane annotations across 14 categories, thereby providing a scalable benchmark for 3D lane detection with synchronized images and events.

DSEC-3DLD is built upon DSEC dataset [12], a real-world driving dataset captured with synchronized standard cameras, event cameras, and LiDAR sensors. Rather than using DSEC only as a source of multimodal recordings, we convert it into a metric RGB-event 3D lane detection benchmark. The resulting dataset contains synchronized image and event observations together with 3D lane annotations in autonomous-driving scenes, covering diverse road layouts, elevation changes, and challenging illumination conditions. This setting allows us to evaluate whether event sensing can improve 3D lane perception under real sensor noise, motion, exposure variation, and sparse high-frequency lane evidence.

To construct metric 3D lane annotations, we follow a multi-stage annotation and verification pipeline. We first annotate lane structures on the synchronized camera views, where lane markings are traced in the image plane with lane-level instance identities. These 2D lane traces are then associated with metric depth cues derived from calibrated LiDAR observations. Using the camera-LiDAR calibration, we lift the annotated image-plane lanes into the ego-coordinate system and fit ordered 3D lane curves along predefined longitudinal anchor positions. To improve annotation quality, the

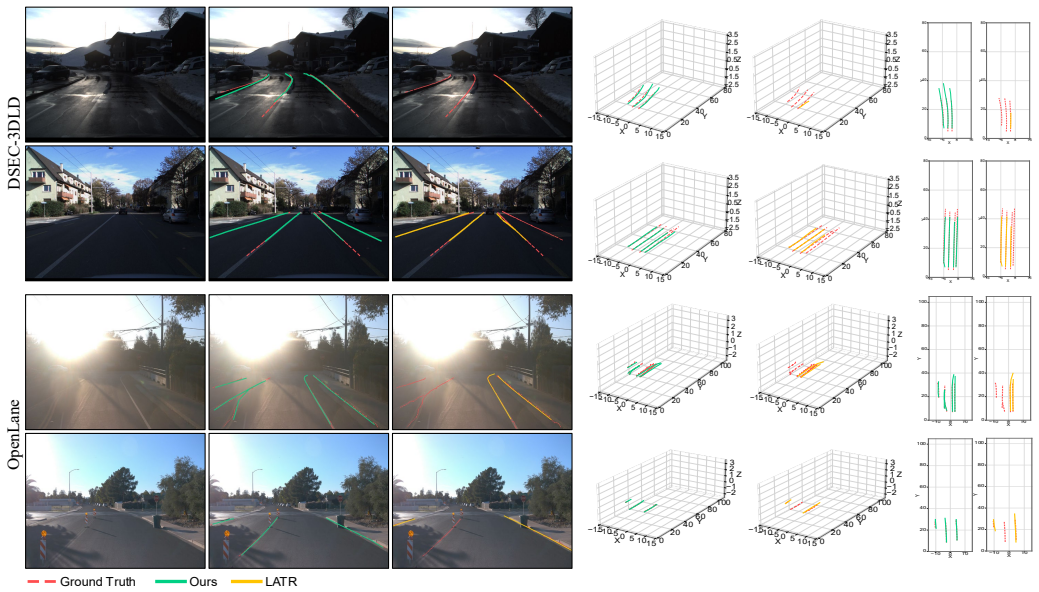


Figure 3: Qualitative comparison on DSEC-3DLD and Ev-OpenLane. We compare RETR with the RGB-only LATR baseline under challenging illumination and road-geometry conditions. Each example shows image-plane lane predictions on the left and corresponding 3D geometry and top-view projections on the right. Red, green, and yellow denote ground truth, RETR, and LATR, respectively. Zoom in for details.

lifted lanes are further refined through geometric consistency checks, including curve continuity, road-surface compatibility, camera-LiDAR projection consistency, and image-event alignment. Lane segments that are highly ambiguous, severely occluded, or cannot be reliably localized in 3D are removed from the annotations. The resulting annotation pipeline provides lane-level existence labels, anchor-wise 3D coordinates, and visibility states, which are used for both training and evaluation. Additional details of the annotation procedure are provided in Supplementary Material A.

Evaluation Metrics. Ev-OpenLane evaluation protocol computes the F1-score by measuring the Euclidean distance between uniformly sampled lane points together with a range-IoU criterion. In addition, the x- and z-errors are reported in the near range $(0\text{–}40\,\mathrm{m})$ and far range $(40\text{–}100\,\mathrm{m})$ to assess the geometric accuracy of 3D lane predictions [7, 14]. DSEC-3DLD follow the distance-based evaluation setting of ONCE-3DLanes [58]. Predicted and ground-truth 3D lane curves are matched according to their centerline distance (CD) [58], and Precision, Recall and F1-score are computed based on the matched lanes.

4.2 Implementation Details

For Ev-OpenLane, all input images are resized to 720×960, and ResNet-50 [18] is adopted as the default backbone. The model is trained for 24 epochs with an initial learning rate of 2×10^{-4}. For DSEC-3DLD, we use an input resolution of 480×640 with ResNet-50 [18] backbone and train for 24 epochs with an initial learning rate of 1×10^{-4}. All models are optimized using AdamW [31] with a weight decay of 0.01 and gradient clipping of 20. All comparison methods and ablation variants follow the same training protocol unless otherwise stated.

4.3 Main Results

Results on DSEC-3DLD. Tab. 1 reports the comparison on DSEC-3DLD. Our approach achieves the best overall performance, with an F1 score of 75.6%, precision of 76.3%, recall of 75.0%, and the lowest centerline distance of 0.094. Compared with the RGB only LATR baseline, it improves the overall F1 score by 6.4 points and reduces the centerline distance from 0.107 to 0.094. The gains are more pronounced under challenging conditions, with 10.6 and 11.4 F1 point improvements in overexposed and underexposed scenes, respectively. On up and down roads, it further improves the F1 score by 9.8 points, indicating stronger 3D geometric robustness under large height variations.

These results show that our reliability based fusion, structural consolidation, and geometry aware decoding effectively exploit the event modality for robust and accurate 3D lane detection.

Results on Ev-OpenLane. Tab. 2 shows the results on Ev-OpenLane under the standard OpenLane evaluation protocol. Our approach achieves an F1 score of 66.7%, outperforming the previous best method by 2.4 points and improving over the RGB only LATR baseline by 4.8 points. It also achieves the lowest localization errors. Compared with SC-Lane, the near and far x errors are reduced from 0.227/0.251 to 0.203/0.230, while the near and far z errors are reduced from 0.088/0.128 to 0.071/0.101. Together with the results on DSEC-3DLD, these improvements demonstrate the effectiveness and scalability of our approach on large-scale 3D lane detection benchmarks. Tab. 5 further reports scenario-wise results.

Qualitative analysis. Fig. 3 shows that our approach gives more accurate predictions than the RGB only LATR baseline under extreme lighting conditions. When strong glare or overexposure makes lane markings faint in the image, LATR often produces missing or fragmented lanes. Our method recovers more complete lane boundaries and maintains better consistency in 3D space. The visual comparisons support the quantitative results, showing that the event modality helps our model remain reliable when RGB appearance becomes degraded.

4.4 Ablation Studies and Analysis

Component analysis. Tabs. 3 and 9 examine the contributions of the proposed components and fusion strategy. Compared with the RGB-only LATR baseline, naive fusion of RGB and event inputs improves the F1-score from 69.2% to 71.8% on DSEC-3DLD and from 61.9% to 63.6% on Ev-OpenLane, showing that the event modality provides useful complementary information. Replacing naive fusion with RCF further improves the performance to 72.9% and 64.5%, respectively. Adding UASC brings the F1-score to 74.3% on DSEC-3DLD and 65.7% on Ev-OpenLane, indicating the benefit of preserving structural lane evidence after multi-scale aggregation. With the full pipeline, our method achieves 75.6% and 66.7% F1-score on the two datasets, outperforming the RGB-only baseline by 6.4 and 4.8 points, respectively.

Table 3: Comparison of component variants in RETR. Results are reported as F1-score (%) on DSEC-3DLD and Ev-OpenLane.

Component	RCF	UASC	GSD	F1-score (%)↑	
				DSEC	Ev-Open
RGB only	✗	✗	✗	69.2	61.9
RCF	✓	✗	✗	72.9	64.5
RCF + UASC	✓	✓	✗	74.3	65.7
Ours	✓	✓	✓	**75.6**	**66.7**

Effect of structural consolidation. Tab. 4 evaluates the contributions of structural consolidation and uncertainty-guided gating. Without UASC, the model achieves F1-scores of 74.1% and 65.2% on DSEC-3DLD and Ev-OpenLane, respectively. Introducing the structural and contextual branches without the gate improves these results to 75.1% and 66.3%, yielding gains of 1.0 and 1.1 percentage points. These improvements support the effectiveness of combining structural details with multi-scale context. Adding the uncertainty gate further improves the F1-scores to 75.6% and 66.7%, providing additional gains of 0.5 and 0.4 percentage points. The complete UASC therefore improves over the variant without UASC by 1.5 percentage points on both datasets, supporting the joint use of structural consolidation and adaptive refinement.

Table 4: Comparison of structural consolidation variants in UASC.

Variant	S&C	Gate	F1-score (%)↑	
			DSEC-3DLD	Ev-OpenLane
w/o UASC	✗	✗	74.1	65.2
w/o Gate	✓	✗	75.1	66.3
Ours	✓	✓	**75.6**	**66.7**

Structural stages in UASC. Tab. 6 evaluates the number of high-resolution structural stages L_h on DSEC-3DLD, with all other settings unchanged. Increasing L_h from one to two improves the F1-score from 74.8% to 75.6%, supporting the use of multi-level structural features. Increasing L_h to three yields 75.0%, with no further improvement. We therefore adopt $L_h = 2$, which achieves the best performance among the evaluated configurations.

Refinement cues in UASC. Tab. 7 examines the contributions of the entropy cue $\mathbf{E}$, high-frequency cue $\mathbf{A}$, and learned detail cue $\mathbf{D}$. Removing these cues individually reduces the F1-score by 0.9, 0.4, and 0.7 percentage points, respectively. The three cues characterize complementary aspects of the feature representation: $\mathbf{E}$ measures ambiguity in the projected feature distribution, $\mathbf{A}$ captures local high-frequency variations, and $\mathbf{D}$ provides learned detail responses. Together, they allow the

Table 5: Scenario-wise quantitative comparison on the Ev-OpenLane validation set using F1-score. The best result in each column is highlighted in bold, and the second-best result is underlined.

Method	All	Up & Down	Curve	Extreme Weather	Night	Intersection	Merge & Split
BEVLaneDet [55]	58.4	48.7	63.1	53.4	53.4	50.3	53.7
LATR [35]	61.9	<u>55.2</u>	68.2	57.1	55.4	52.3	61.5
HeightLane [45]	62.7	53.6	69.3	55.4	54.6	54.1	61.1
PVALane [63]	62.7	54.1	67.3	**62.0**	<u>57.2</u>	53.4	60.0
SC-Lane [44]	<u>64.3</u>	54.6	<u>70.7</u>	58.1	56.5	<u>56.1</u>	<u>63.0</u>
RETR (Ours)	**66.7**	**60.0**	**73.1**	<u>59.8</u>	**60.5**	**58.9**	**65.9**

Table 6: Effect of structural stages in UASC.

L_h	1	2	3
F1-score (%)	74.8	**75.6**	75.0

Table 7: Effect of refinement cues in UASC.

Cues	$\mathbf{E}$+$\mathbf{A}$	$\mathbf{E}$+$\mathbf{D}$	$\mathbf{A}$+$\mathbf{D}$	All
F1-score (%)	74.9	75.2	74.7	**75.6**

Table 8: Effect of inquiry branches in GSD.

M	1	2	3
F1-score (%)	74.5	**75.6**	75.1

refinement gate to account for both feature ambiguity and local structural evidence. Removing $\mathbf{E}$ causes the largest performance drop, from 75.6% to 74.7%, highlighting the value of ambiguity information beyond detail cues alone. None of the evaluated two-cue combinations matches the complete configuration, supporting the complementary roles of all three cues in uncertainty-guided structural refinement.

Inquiry branches in GSD. Tab. 8 evaluates the number of inquiry branches M on DSEC-3DLD. Increasing M from one to two improves the F1-score from 74.5% to 75.6%. The branches independently process the same lane states through geometry-conditioned queries, and their outputs are aggregated to update the anchor representations. The 1.1-percentage-point improvement supports the effectiveness of combining multiple geometric inquiries. Increasing M to three yields 75.1%, without improving on the two-branch configuration. We therefore use $M = 2$ in the default model.

Effect of fusion strategy. Tab. 9 compares RGB-event fusion strategies under the same backbone and decoder, isolating the effect of cross-modal interaction. Direct concatenation improves over the RGB-only baseline, confirming that event observations provide complementary lane cues. However, it treats RGB and event features as homogeneous channels and does not explicitly address their appearance-structure discrepancy. Among the evaluated fusion strategies, RCF achieves the best performance on both benchmarks, improving over concatenation by 1.1 and 0.9 percentage points on DSEC-3DLD and Ev-OpenLane, respectively. This validates our design that effective fusion should first identify modality-consistent lane evidence and then exchange complementary context bidirectionally.

Table 9: Comparison of image and event fusion strategies.

Method	F1-score (%)↑	
	DSEC-3DLD	Ev-OpenLane
Concatenation	71.8	63.6
Cross Attention	72.3	63.4
CBAM [57]	72.1	63.8
RCF	**72.9**	**64.5**

5 Conclusion

We introduced RETR, a structure preserving transformer for 3D lane detection with RGB and event inputs. To the best of our knowledge, this work is the first to bring event cameras into 3D lane detection and to establish dedicated multimodal benchmarks for this setting, including DSEC-3DLD with real event data and Ev-OpenLane with simulated event observations. Our results show that aligning complementary RGB-event evidence, preserving structural details, and decoding ordered lane geometry together improve 3D lane detection across both benchmarks. RETR follows this principle by aligning modality-specific evidence through Reciprocal Context Flow Fusion, preserving thin and uncertain lane structures with Uncertainty Aware Structural Consolidation, and decoding ordered lane hypotheses with the Geometry State Decoder. Extensive experiments on both benchmarks show that RETR achieves state-of-the-art performance and strong robustness under challenging illumination and road geometry, validating the effectiveness of our structure preserving design and demonstrating the importance of event cameras for robust 3D lane detection.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (62302045) and the Fundamental Research Funds for the Central Universities.

References

[1] Inigo Alonso and Ana C Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Luca Bartolomei, Matteo Poggi, Andrea Conti, and Stefano Mattoccia. Lidar-event stereo fusion with hallucinations. In *European Conference on Computer Vision*, pages 125–145. Springer, 2024.

[3] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane markers using maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[4] Spencer Carmichael, Austin Buchan, Mani Ramanagopal, Radhika Ravi, Ram Vasudevan, and Katherine A. Skinner. Dataset and benchmark: Novel sensors for autonomous vehicle perception. *International Journal of Robotics Research*, 44(3):355–365, 2025.

[5] Bharatesh Chakravarthi, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermuller, and Yezhou Yang. Recent event camera innovations: A survey. In *European Conference on Computer Vision Workshops*, pages 342–376, 2025.

[6] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.

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

[8] Wensheng Cheng, Hao Luo, Wen Yang, Lei Yu, Shoushun Chen, and Wei Li. Det: A high-resolution dvs dataset for lane extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[9] Hoonhee Cho, Jae-young Kang, Youngho Kim, and Kuk-Jin Yoon. Ev-3dod: Pushing the temporal boundaries of 3d object detection with event cameras. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 27197–27210, 2025.

[10] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 525–533, 2022.

[11] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020.

[12] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[13] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *European Conference on Computer Vision Workshops*, pages 0–0, 2018.

[14] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *European Conference on Computer Vision*, pages 666–681. Springer, 2020.

[15] Yudong Han, Qingpei Guo, Liyuan Pan, Liu Liu, Yu Guan, and Ming Yang. DynFocus: Dynamic cooperative network empowers LLMs with video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8512–8522, 2025.

[16] Yudong Han, Yong Wang, Zaiquan Yang, Liang Lin, Chongyang Tao, Xiangxiang Chu, and Liyuan Pan. Beyond entropy: Self-diagnostic multi-role token optimization for video reasoning. *arXiv preprint arXiv:2610.03400*, 2026.

[17] Yudong Han, Yong Wang, Zaiquan Yang, Zhen Qu, Liyuan Pan, and Xiangxiang Chu. Visual enhanced depth scaling for multimodal latent reasoning. *arXiv preprint arXiv:2604.10500*,

2026.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

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

[20] Shaofei Huang, Zhenwei Shen, Zehao Huang, Yue Liao, Jizhong Han, Naiyan Wang, and Si Liu. Anchor3dlane++: 3d lane detection via sample-adaptive sparse 3d anchor regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[21] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020.

[22] Jae-Young Kang, Hoonhee Cho, and Kuk-Jin Yoon. Unleashing the temporal potential of stereo event cameras for continuous-time 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6869–6881, 2025.

[23] Chunliang Li, Wencheng Han, Junbo Yin, Sanyuan Zhao, and Jianbing Shen. Repvf: A unified vector fields representation for multi-task 3d perception. In *European Conference on Computer Vision*, pages 273–292. Springer, 2024.

[24] Chenguang Li, Jia Shi, Ya Wang, and Guangliang Cheng. Reconstruct from bev: A 3d lane detection approach based on geometry structure prior. *arXiv preprint arXiv:2206.10098*, 2022.

[25] Jianing Li, Siwei Dong, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Event-based vision enhanced: A joint detection framework in autonomous driving. In *2019 IEEE International Conference on Multimedia and Expo*, pages 1396–1401. IEEE, 2019.

[26] Xinxu Lin, Mingxuan Liu, Kezhuo Liu, and Hong Chen. Spike-slr: an energy-efficient parallel spiking transformer for event-based sign language recognition. In *35th British Machine Vision Conference*, 2024.

[27] Enqi Liu, Liyuan Pan, Zhi Gao, Lingzhi Li, and Qing Li. Seeing through fog: Towards fog-invariant action recognition. *arXiv preprint arXiv:2605.20645*, 2026.

[28] Enqi Liu, Liyuan Pan, Zhi Gao, Yan Yang, Chenrui Shi, Yang Liu, Jingrong Wu, and Qing Li. Benchmarking and improving GUI agents in high-dynamic environments. *arXiv preprint arXiv:2604.25380*, 2026.

[29] Enqi Liu, Liyuan Pan, Yan Yang, Yiran Zhong, Zhijing Wu, Xinxiao Wu, and Liu Liu. Storyboard-guided alignment for fine-grained video action recognition. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[32] Xinglong Luo, Ao Luo, Zhengning Wang, Chunyu Lin, Bing Zeng, and Shuaicheng Liu. Efficient meshflow and optical flow estimation from event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19198–19207, 2024.

[33] Yueru Luo, Shuguang Cui, and Zhen Li. Dv-3dlane: End-to-end multi-modal 3d lane detection with dual-view representation. *arXiv preprint arXiv:2406.16072*, 2024.

[34] Yueru Luo, Xu Yan, Chaoda Zheng, Chao Zheng, Shuqi Mei, Tang Kun, Shuguang Cui, and Zhen Li. M^2-3DLaneNet: Exploring Multi-Modal 3D Lane Detection. *arXiv preprint arXiv:2209.05996*, 2022.

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

[36] Hanspeter A Mallot, Heinrich H Bülthoff, James J Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64(3):177–185, 1991.

[37] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.

[38] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on*

Intelligent Robots and Systems, pages 1–9. IEEE, 2018.

[39] Qirui Ning, Jinlai Zhang, Yuhang Xie, Kaifeng Liu, Kai Gao, Bin Chen, Gengbiao Chen, Qing Fan, Hui Liu, and Ronghua Du. Multi-resolution context augmentation and dual channel attention for 3d lane detection. *IEEE Internet of Things Journal*, 2025.

[40] Halil İbrahim Öztürk, Muhammet Esat Kalfaoğlu, and Ozsel Kilinc. Glane3d: Detecting lanes with graph of 3d keypoints. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 27508–27518, 2025.

[41] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1669–1678. IEEE, 2020.

[42] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019.

[43] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Chaesong Park, Eunbin Seo, Jihyeon Hwang, and Jongwoo Lim. Sc-lane: Slope-aware and consistent road height estimation framework for 3d lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28407–28416, 2025.

[45] Chaesong Park, Eunbin Seo, and Jongwoo Lim. Heightlane: Bev heightmap guided 3d lane detection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1692–1701. IEEE, 2025.

[46] Maximilian Pittner, Alexandru Condurache, and Joel Janai. 3d-splinenet: 3d traffic line detection using parametric spline representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 602–611, 2023.

[47] Maximilian Pittner, Joel Janai, and Alexandru P Condurache. Lanecpp: Continuous 3d lane detection using physical priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10639–10648, 2024.

[48] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982. PMLR, 2018.

[49] Waseem Shariff, Mehdi Sefidgar Dilmaghani, Paul Kielty, Mohamed Moustafa, Joe Lemley, and Peter Corcoran. Event cameras in automotive sensing: A review. *IEEE Access*, 12:51275–51306, 2024.

[50] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7244–7253, 2019.

[51] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *European Conference on Computer Vision*, pages 412–428. Springer, 2022.

[52] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[53] Abhishek Tomy, Anshul Paigwar, Khushdeep S Mann, Alessandro Renzaglia, and Christian Laugier. Fusing event-based and rgb camera for robust object detection in adverse conditions. In *2022 International Conference on Robotics and Automation*, pages 933–939. IEEE, 2022.

[54] TuSimple. Tusimple dataset. `https://cvpr2017.tusimple.ai/`, 2017.

[55] Ruihao Wang, Jian Qin, Kaiying Li, Yaochen Li, Dong Cao, and Jintao Xu. 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*, pages 1002–1011, 2023.

[56] Zhaohui Wang, Junren Sun, Haobo Sun, Shuting Zhang, Haowen Wang, and Bocheng Zhu. 3dlanedt: 3d lane detection via reliable depth and temporal fusion for autonomous driving. *Expert Systems with Applications*, page 131931, 2026.

[57] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, pages 3–19, 2018.

[58] Fan Yan, Ming Nie, Xinyue Cai, Jianhua Han, Hang Xu, Zhen Yang, Chaoqiang Ye, Yanwei Fu, Michael Bi Mi, and Li Zhang. Once-3dlanes: Building monocular 3d lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17143–17152, 2022.

[59] Hao Yang, Liyuan Pan, Yan Yang, Richard Hartley, and Miaomiao Liu. LDP: Language-driven dual-pixel image defocus deblurring network. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 24078–24087, 2024.

[60] Hao Yang, Liyuan Pan, Yan Yang, and Wei Liang. Language-driven all-in-one adverse weather removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24902–24912, 2024.

[61] Hao Yang, Yan Yang, Ruikun Zhang, and Liyuan Pan. A preliminary study on GPT-Image generation model for image restoration. *arXiv preprint arXiv:2505.05621*, 2025.

[62] Chengtang Yao, Lidong Yu, Yuwei Wu, and Yunde Jia. Sparse point guided 3d lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8363–8372, 2023.

[63] Zewen Zheng, Xuemin Zhang, Yongqiang Mou, Xiang Gao, Chengxin Li, Guoheng Huang, Chi-Man Pun, and Xiaochen Yuan. Pvalane: Prior-guided 3d lane detection with view-agnostic feature alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7597–7604, 2024.

[64] Chu Zhou, Minggui Teng, Jin Han, Jinxiu Liang, Chao Xu, Gang Cao, and Boxin Shi. Deblurring low-light images with events. *International Journal of Computer Vision*, 131(5):1284–1298, 2023.

[65] Zhuyun Zhou, Zongwei Wu, Rémi Boutteau, Fan Yang, Cédric Demonceaux, and Dominique Ginhac. Rgb-event fusion for moving object detection in autonomous driving. In *2023 IEEE International Conference on Robotics and Automation*, pages 7808–7815. IEEE, 2023.

[66] Zhuyun Zhou, Zongwei Wu, Danda Pani Paudel, Rémi Boutteau, Fan Yang, Luc Van Gool, Radu Timofte, and Dominique Ginhac. Event-free moving object segmentation from moving ego vehicle. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8960–8965. IEEE, 2024.

[67] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.

[68] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018.

[69] Lin Zhu, Xianzhang Chen, Lizhi Wang, Xiao Wang, Yonghong Tian, and Hua Huang. Continuous-time object segmentation using high temporal resolution event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Supplementary Material

The supplementary material provides dataset construction and annotation details, event representations, model implementation details, additional modality and efficiency analyses, and further qualitative results.

A Additional Dataset and Model Details

A.1 DSEC-3DLD Statistics and Data Splits

DSEC-3DLD comprises 52 driving sequences with 30,156 synchronized RGB-event samples and 82,746 annotated 3D lane instances. We partition the dataset at the sequence level into 33 training, 6 validation, and 13 test sequences, as summarized in Tab. 11. Frames are uniformly sampled at 10 Hz within each sequence. All samples from a given sequence are assigned to the same split to prevent cross-split leakage from temporally adjacent observations within that sequence.

A.2 DSEC-3DLD Annotation Pipeline

DSEC-3DLD is constructed to provide real-world RGB-event data with metric 3D lane annotations. Since existing event-camera driving datasets do not directly provide 3D lane labels, we build a multi-source annotation pipeline that combines image evidence, event observations, calibration information, and geometric cues. The overall pipeline is shown in Fig. 4. It consists of three main stages: data preparation, multi-source annotation, and 3D ground-truth generation.

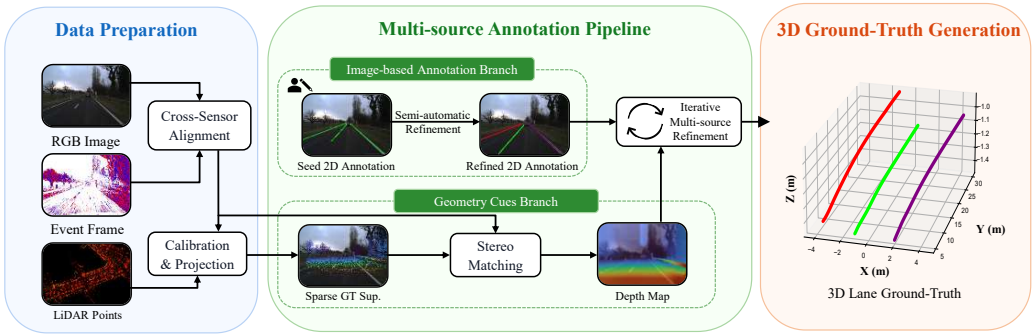


Figure 4: **Annotation pipeline of DSEC-3DLD.** We first prepare synchronized RGB images, event frames, and LiDAR points through cross-sensor alignment and calibration-based projection. Then, we construct a multi-source annotation pipeline with an image-based annotation branch and a geometry-cue branch. The image-based branch provides refined 2D lane annotations, while the geometry branch provides sparse geometric supervision and depth cues. These complementary sources are jointly refined to generate metric 3D lane ground-truth.

Data preparation. For each driving sequence, we first synchronize RGB images and event observations and align them with the available geometric measurements. The event stream is converted into an event frame that preserves polarity and motion-induced structural responses. In parallel, LiDAR points are projected to the camera view using calibrated extrinsic and intrinsic parameters. This cross-sensor alignment step provides a common image coordinate system for RGB appearance, event responses, and sparse 3D geometry.

More specifically, given a LiDAR point $\mathbf{p}_l = [x_l, y_l, z_l, 1]^\top$ in the LiDAR coordinate system, we project it into the camera image plane by

$$\tilde{\mathbf{u}} = \mathbf{K}\mathbf{T}_{c\leftarrow l}\mathbf{p}_l, \qquad \mathbf{u} = \left[\frac{\tilde{u}_x}{\tilde{u}_z}, \frac{\tilde{u}_y}{\tilde{u}_z}\right]^\top, \tag{16}$$

where $\mathbf{K}$ is the camera intrinsic matrix and $\mathbf{T}_{c\leftarrow l}$ denotes the rigid transformation from the LiDAR coordinate system to the camera coordinate system. Points outside the image range or behind the camera are discarded. The projected points provide sparse metric depth cues for lane regions.

Image-based annotation branch. The first branch focuses on image-space lane annotation. Since the RGB camera and event camera have different imaging planes, viewpoints, and resolutions, we first align the RGB frame and the event frame into a unified image coordinate system using camera calibration and homography-based warping [53]. We then perform manual initial annotation on the synchronized and spatially aligned RGB frames to obtain seed 2D lane annotations. These seed annotations define the number of lane instances, their image-space locations, visible extents, and coarse lane topology.

However, relying only on RGB images can be affected by overexposure, underexposure, shadows, motion blur, and weak lane textures. Therefore, we further incorporate the aligned event frames for semi-automatic refinement. Event frames can preserve structural responses along lane boundaries when RGB appearance is degraded, especially under strong illumination changes or locally weak textures. By jointly inspecting RGB appearance and event responses, we correct the initial 2D annotations and obtain more stable refined 2D lane annotations.

Geometry-cue branch. The second branch provides metric geometric constraints for 3D lane annotation. After projecting LiDAR points onto the image plane using the calibrated LiDAR-to-camera transformation, we obtain sparse metric depth measurements around the road surface and lane regions. These sparse depth values are converted into sparse disparity supervision. For a stereo camera with focal length f and baseline B, the disparity corresponding to the projected LiDAR depth d_L^i is computed as

$$s_L^i = \frac{fB}{d_L^i}. \tag{17}$$

The sparse disparity map is then constructed by assigning each projected LiDAR point to its image location:

$$\mathbf{S}_L(\mathbf{u}^i) = s_L^i, \tag{18}$$

where $\mathbf{u}^i$ denotes the projected image coordinate of the i-th LiDAR point.

Using this sparse disparity map as geometric supervision, we employ a state-of-the-art open-source event-based stereo matching algorithm [2] to estimate a dense disparity map:

$$\hat{\mathbf{S}} = \mathcal{M}_{\text{stereo}}\left(\mathbf{I}_L, \mathbf{I}_R, \mathbf{E}_L, \mathbf{E}_R, \mathbf{S}_L\right), \tag{19}$$

where $\mathbf{I}_L$ and $\mathbf{I}_R$ are the stereo RGB frames, $\mathbf{E}_L$ and $\mathbf{E}_R$ are the corresponding event representations, $\mathbf{S}_L$ is the sparse LiDAR-derived disparity supervision, and $\hat{\mathbf{S}}$ is the estimated dense disparity map. The dense depth map is recovered by

$$\hat{\mathbf{D}}(\mathbf{u}) = \frac{fB}{\hat{\mathbf{S}}(\mathbf{u}) + \epsilon}, \tag{20}$$

where ϵ is a small constant for numerical stability.

The estimated dense depth map is used as a geometric prior rather than directly as the final annotation. We sample depth values along each manually refined 2D lane and lift the lane points into metric 3D space using the calibrated camera model. The lifted lanes are further checked with the LiDAR projection and manually inspected in both the image view and the ground-coordinate view. This process converts sparse LiDAR supervision and dense event-stereo geometry into reliable 3D lane ground-truth.

Iterative multi-source refinement. After obtaining the refined 2D lane annotations and dense depth maps, we perform iterative multi-source refinement by jointly checking RGB appearance, event responses, LiDAR projections, and stereo depth estimates. This step verifies both image-space lane localization and metric 3D geometry consistency.

During the generation of 3D labels, we observe that directly fitting 2D lane curves or depth sequences in the image coordinate system with a global high-order function can introduce severe oscillatory artifacts, known as the Runge phenomenon. This issue is particularly evident in the far range, where lane markings are highly compressed in the image plane and small pixel-level errors can be amplified into large 3D localization errors. In addition, depth candidates around lane regions are not perfectly continuous, and global high-order interpolation tends to produce unstable geometry in sparse or noisy regions.

To address this problem, we do not generate final 3D labels by fitting lanes in the image plane. Instead, we first lift each refined 2D lane point into the camera coordinate system using its corresponding depth. For an image-space lane point $\mathbf{q}_j = [u_j, v_j]^\top$ with depth $\hat{d}_j$, the corresponding 3D point in the camera coordinate system is computed as

$$\mathbf{x}_c^j = \hat{d}_j \mathbf{K}^{-1} \left[u_j, v_j, 1\right]^\top , \tag{21}$$

where $\mathbf{K}$ is the camera intrinsic matrix. The point is then transformed into the ground-coordinate system by

$$\mathbf{x}_g^j = \mathbf{T}_{g \leftarrow c} \begin{bmatrix} \mathbf{x}_c^j \\ 1 \end{bmatrix} . \tag{22}$$

Thus, each lifted lane is represented as a metric 3D point sequence

$$\hat{\mathcal{L}} = \{(\hat{x}_j, \hat{y}_j, \hat{z}_j)\}_{j=1}^{N} . \tag{23}$$

We then perform lane-wise refinement in the metric ground-coordinate system rather than global high-order fitting in the image plane. Specifically, we use the longitudinal coordinate $\hat{y}$ as the independent variable and locally smooth or piecewise interpolate the lateral position $\hat{x}$ and height $\hat{z}$. The refined lane is finally resampled at predefined longitudinal positions:

$$\mathcal{L}^{3D} = \{(x(y_k), y_k, z(y_k), m_k)\}_{k=1}^{K} , \qquad y_k \in \mathcal{Y}, \tag{24}$$

where $\mathcal{Y}$ denotes the predefined longitudinal sampling positions and m_k is the validity indicator. By lifting lane points from the image coordinate system to the camera coordinate system, transforming them into the ground-coordinate system, and refining them in metric space, we avoid the Runge phenomenon caused by image-space global fitting and obtain more stable 3D lane annotations with plausible lateral position, height variation, and longitudinal continuity.

Quality control. We perform quality control to ensure reliable 3D lane annotations. For scenario categorization, we first use GPT-4o to assist the preliminary classification of driving scenes, including normal, overexposed, underexposed, curve, and merge/split cases. The generated scenario labels are then manually reviewed, and samples with inaccurate or ambiguous labels are reassigned to the correct categories.

Independent geometric validation. We evaluate 400 frames containing 1,126 annotated lane instances using synchronized geometric measurements withheld from annotation generation and refinement. Depth consistency error (DCE) is the absolute difference between annotated anchor depth and the corresponding independent measurement, excluding out-of-image anchors and invalid measurements. The mean, median, and 95th-percentile DCE are 0.064 m, 0.041 m, and 0.157 m, respectively, supporting the depth consistency of the annotations.

To further assess the reliability of DSEC-3DLD annotations, we introduce an expert scoring protocol. We invite ten expert annotators to evaluate the annotated samples from both image-space and metric 3D perspectives. As shown in Tab. 13, the assessment focuses on three key aspects: 2D annotation accuracy, 3D lane accuracy, and scenario label accuracy. Each criterion is scored on a 1–5 scale, where a higher score indicates better annotation quality. Samples with low scores are corrected or removed from the final dataset. The final annotations are retained only when the image-space lane structure, metric 3D geometry, and scenario labels are all verified.

A.3 Ev-OpenLane Generation

OpenLane is a large-scale 3D lane detection benchmark, but it does not provide event-camera observations. Therefore, it cannot directly support the study of RGB-event multimodal 3D lane detection. To construct a scalable RGB-event 3D lane benchmark, we generate event observations from OpenLane using ESIM [48], a standard open-source event camera simulator, while preserving the original 3D lane annotations.

Directly simulating events from the original OpenLane frames is insufficient because the original sequences have a relatively low frame rate. Large inter-frame displacement may cause abrupt brightness changes and unstable event generation. To address this issue, we first resize the images

Table 10: ESIM simulation settings for generating Ev-OpenLane. OpenLane images are resized to 640×480 and temporally upsampled to 30 fps before event simulation.

Parameter	Setting	Description
Original frame rate	10 fps	Frame rate of the original OpenLane key frames.
Interpolated frame rate	30 fps	Frame rate after temporal interpolation.
Interpolated frame interval	33.33 ms	Temporal interval between adjacent interpolated frames.
Image resolution	640×480	Resolution used for ESIM simulation and event representation.
Input timestamp interval	33.33 ms	Timestamp interval for ESIM input frames.
Data source	Image folder	ESIM reads interpolated frames and timestamps from an image sequence.
ESIM data source flag	`data_source=2`	Image-folder mode in ESIM.
Log image	`use_log_image=1`	Converts input images into the log-intensity domain.
Log epsilon	`log_eps=0.1`	Numerical offset used in log-intensity conversion.
Positive contrast threshold C^+	0.15	Threshold for generating positive events.
Negative contrast threshold C^-	0.15	Threshold for generating negative events.
ESIM publisher frame rate	30 fps	Frame rate used to publish the interpolated input stream.
Exposure time	10.0 ms	Exposure time of the simulated APS frames.
Event accumulation window	100 ms	Events are accumulated between adjacent OpenLane key frames.

to 640×480 and temporally upsample the RGB sequences to 30 fps by frame interpolation. The interpolated frames have a temporal interval of 33.33 ms and are then used as the input to ESIM.

Given the interpolated image sequence, ESIM generates events according to the log-intensity change at each pixel. An event is triggered when the brightness change reaches the positive or negative contrast threshold:

$$\Delta L(\mathbf{x}, t) = L(\mathbf{x}, t) - L(\mathbf{x}, t_{\text{last}}), \qquad |\Delta L(\mathbf{x}, t)| \geq C, \tag{25}$$

where $L(\mathbf{x}, t)$ denotes the log intensity at pixel $\mathbf{x}$ and time t, t_{last} is the timestamp of the last event at the same pixel, and C is the contrast threshold. The event polarity is determined by the sign of $\Delta L(\mathbf{x}, t)$.

We generate the timestamp file required by ESIM according to the 30 fps interpolated frame rate. The main simulation settings are summarized in Tab. 10. The generated event streams are synchronized with the corresponding OpenLane key frames and paired with the original 3D lane annotations, resulting in Ev-OpenLane for large-scale RGB-event 3D lane detection.

By providing complementary real-world and large-scale simulated RGB-event 3D lane benchmarks, we hope DSEC-3DLD and Ev-OpenLane can serve as useful resources for advancing event-camera-based perception in autonomous driving.

Transfer to real events. Training on Ev-OpenLane and evaluating directly on DSEC-3DLD yields an F1-score of 56.4%. Adapting the pretrained model with 10% of the DSEC-3DLD training data improves this to 69.4%, while full DSEC-3DLD training achieves 75.6%. Limited real-event supervision thus substantially improves transfer performance. The remaining gap reflects cross-dataset differences, including sensing conditions and scene distributions, rather than event simulation alone.

A.4 Event Representation and Accumulation Window

For DSEC-3DLD, an RGB frame timestamped at t_k is paired with all valid events in the causal interval $[t_k - 50\,\text{ms}, t_k]$. We spatially align the events with the RGB image and encode them into a polarity-preserving voxel representation. We do not manually select lane-related events, and events after t_k are excluded. The default accumulation window is 50 ms for DSEC-3DLD and 100 ms for Ev-OpenLane.

Accumulation-window ablation. Tab. 15 evaluates the event accumulation duration on DSEC-3DLD while keeping the remaining settings unchanged. The 50-ms window achieves the highest F1-score among the tested durations. Reducing the window to 25 ms or increasing it to 100 ms decreases the F1-score by 0.7 and 0.6 percentage points, respectively.

Table 11: Dataset statistics and sequence-level splits of DSEC-3DLD.

Split	Sequences	Samples	Lane instances
Training	33	19,048	52,422
Validation	6	4,749	12,503
Test	13	6,359	17,821
Total	52	30,156	82,746

Table 12: Default architectural settings of RETR, including UASC and GSD.

Symbol	Description	Value
L_h	High-resolution structural stages in UASC	2
N	Initial lane proposals	12
K_s	Activation maps per proposal	4
A	Ordered anchors per proposal	20
C	Hidden dimension	256
K	Decoder layers	6
M	Inquiry branches per decoder layer	2

A.5 GSD Implementation Details

Proposal descriptor extraction. The instance activation map (IAM) head predicts K_s activation maps for each lane proposal. After sigmoid activation and spatial normalization, each map performs weighted pooling over the consolidated feature $\mathbf{F}_c$. The pooled descriptors are concatenated and projected to obtain the proposal descriptor:

$$\mathbf{s}_i = G\left(W_s\left[\mathrm{Pool}_{i,1}(\mathbf{F}_c); \ldots; \mathrm{Pool}_{i,K_s}(\mathbf{F}_c)\right] + \mathbf{b}_s\right), \tag{26}$$

where $\mathrm{Pool}_{i,j}$ denotes weighted spatial pooling using the j-th normalized activation map of proposal i. The semicolon denotes feature concatenation, W_s and $\mathbf{b}_s$ are projection parameters, and G is the proposal-refinement mapping. The resulting descriptor $\mathbf{s}_i \in \mathbb{R}^C$ is used to initialize the ordered anchor tokens of proposal i.

Ordered anchor initialization. The learned anchor embeddings are implemented as a matrix $\mathbf{P} = [\mathbf{p}_1; \ldots; \mathbf{p}_A] \in \mathbb{R}^{A\times C}$ using `nn.Embedding` with its default initialization. They are shared across proposals and jointly optimized with the network. Each embedding is associated with a predefined longitudinal anchor position. The initial token for anchor a of proposal i is constructed as

$$\mathbf{q}_{i,a}^0 = \mathrm{MLP}_q\left(\mathbf{s}_i + \mathbf{p}_a\right). \tag{27}$$

Here, $\mathbf{s}_i$ is broadcast across the A anchors and added to their respective embeddings. The resulting tokens form $\mathbf{Q}_i^0 \in \mathbb{R}^{A\times C}$.

Inquiry-branch aggregation. At decoder layer k, let $\mathbf{q}_{i,a}^{k,m} \in \mathbb{R}^C$ denote the state of anchor a in proposal i from the m-th branch output $\mathbf{Q}_m^k$. The states from the M branches are concatenated, projected back to dimension C, and normalized:

$$\mathbf{q}_{i,a}^k = \mathrm{LN}\left(W_{\mathrm{agg}}\left[\mathbf{q}_{i,a}^{k,1}; \ldots; \mathbf{q}_{i,a}^{k,M}\right] + \mathbf{b}_{\mathrm{agg}}\right), \tag{28}$$

where $W_{\mathrm{agg}} \in \mathbb{R}^{C\times MC}$, $\mathbf{b}_{\mathrm{agg}} \in \mathbb{R}^C$, and LN denotes layer normalization. The aggregated anchor states form $\mathbf{Q}^k \in \mathbb{R}^{N\times A\times C}$.

Default configuration. Tab. 12 summarizes the default architectural settings of RETR. In addition to the GSD parameters, we report the number of high-resolution structural stages L_h used by UASC.

A.6 Architecture and Event-Modality Contributions

We compare RETR with RGB-only, event-only, and event-perturbed variants on DSEC-3DLD. Tab. 14 summarizes the results, with LATR serving as an RGB-only baseline.

Architectural contribution. RETR-RGB improves the F1-score from 69.2% to 72.5% compared with LATR, yielding a gain of 3.3 percentage points without event input. This result demonstrates an improvement under the RGB-only setting.

Complementary modalities. RETR-Event achieves an F1-score of 64.8%, compared with 72.5% for RETR-RGB. Combining RGB and event observations achieves 75.6%, outperforming both single-modality variants. These results support the use of complementary RGB and event observations for 3D lane detection.

Table 13: Expert assessment of DSEC-3DLD annotations, reporting mean scores (1–5) and quality-control pass rates.

Criterion	Score	Pass (%)
2D annotation	4.62	96.4
3D lane	4.41	93.7
Scenario label	4.70	97.8
Overall	4.57	95.9

Table 14: Comparison of RETR input variants on DSEC-3DLD, including RGB-only, event-only, RGB-event, and noisy-event configurations.

Method	Input	F1-score (%)
LATR [35]	RGB	69.2
RETR-RGB	RGB	72.5
RETR-Event	Event	64.8
RETR-Noisy	RGB + noisy event	74.6
RETR	RGB + event	**75.6**

Table 15: Event window ablation on DSEC-3DLD with other settings fixed.

Accum. window	F1 (%)
25 ms	74.9
50 ms	**75.6**
100 ms	75.0

Table 16: Accuracy and computational cost of RETR and its single-modality variants on DSEC-3DLD. Preprocessing time and network latency are reported separately.

Method	Params (M)	MACs (G)	Preproc. (ms)	Network (ms)	F1-score (%)
RETR-RGB	83.56	117.6	69.9	65.03	72.5
RETR-Event	84.93	119.2	78.5	64.92	64.8
RETR	95.47	148.6	80.9	81.26	**75.6**

Sensitivity to event perturbations. For RETR-Noisy, we retrain the model with zero-mean Gaussian noise applied only to active event-voxel entries, leaving zero-event regions unchanged. The F1-score decreases from 75.6% to 74.6%. This result indicates sensitivity to event perturbations and is consistent with a benefit from reliable event information. Since the perturbation retains part of the event structure, it does not constitute a complete removal of event information.

A.7 Accuracy–Efficiency Tradeoff

All variants are profiled using NVIDIA RTX 4090 hardware. Network latency is averaged over 1,000 runs after 100 warm-up iterations, with CUDA synchronization. Tab. 16 reports parameter counts, multiply–accumulate operations (MACs), preprocessing time, network latency, and F1-score.

Compared with RETR-RGB, the full RGB-event model improves the F1-score by 3.1 percentage points, with 11.91M additional parameters and 31.0G additional MACs. Preprocessing time increases from 69.9 ms to 80.9 ms, while network latency increases from 65.03 ms to 81.26 ms. RETR-Event has a similar parameter count and network latency to RETR-RGB but achieves a lower F1-score. With 11.91M additional parameters compared with RETR-RGB, the full model achieves the best F1-score of 75.6% among the evaluated variants.

B More Qualitative Results and Analysis

In this section, we present additional qualitative results to provide a more comprehensive evaluation of our method. The visualizations show that our model maintains strong performance across diverse illumination conditions and road structures, including overexposed and underexposed scenes, straight and curved roads, as well as complex areas such as intersections and merge-split regions. These results further verify the advantage of our method in challenging scenarios, and also suggest that event cameras are a promising sensing modality for robust 3D lane detection in autonomous driving.

Broader Impacts This work supports robust autonomous-driving perception by advancing RGB-event 3D lane detection and providing benchmarks for event-based multimodal road-structure understanding. These resources may facilitate research under challenging illumination, high dynamic range, and fast-motion conditions.

Limitation and future work. As RGB-Event 3D lane detection is still at an early stage, the availability of real event camera data with metric 3D lane annotations is naturally limited. Our work establishes a first foundation for this setting through dedicated benchmarks and a structure-preserving model. Future research can build upon this foundation by scaling real-world annotations, covering more sensor configurations, and improving deployment efficiency.

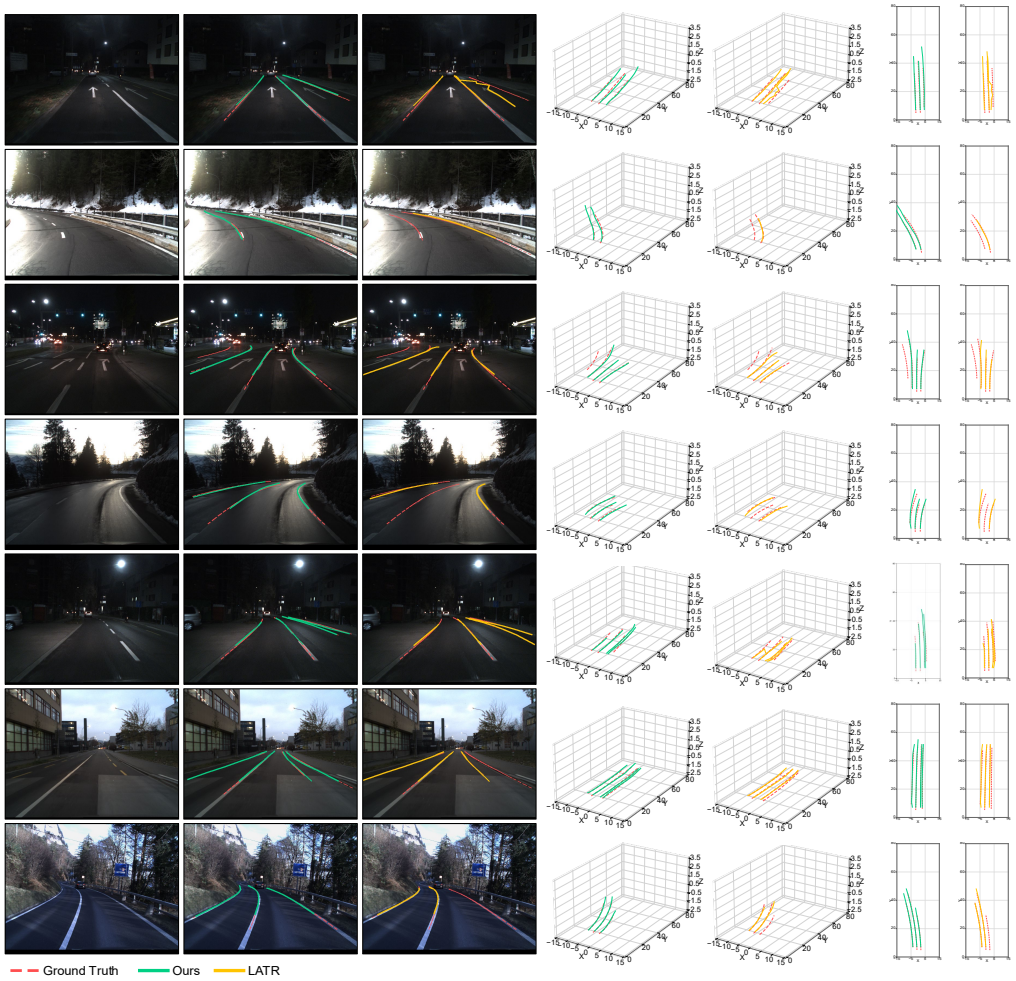


Figure 5: Additional qualitative results on DSEC-3DLD. Red dashed lines denote ground truth, green lines denote our predictions, and yellow lines denote LATR predictions. Our method shows better agreement with the ground truth in both image-space and 3D visualizations, demonstrating robust performance under real-world illumination and road-geometry variations.

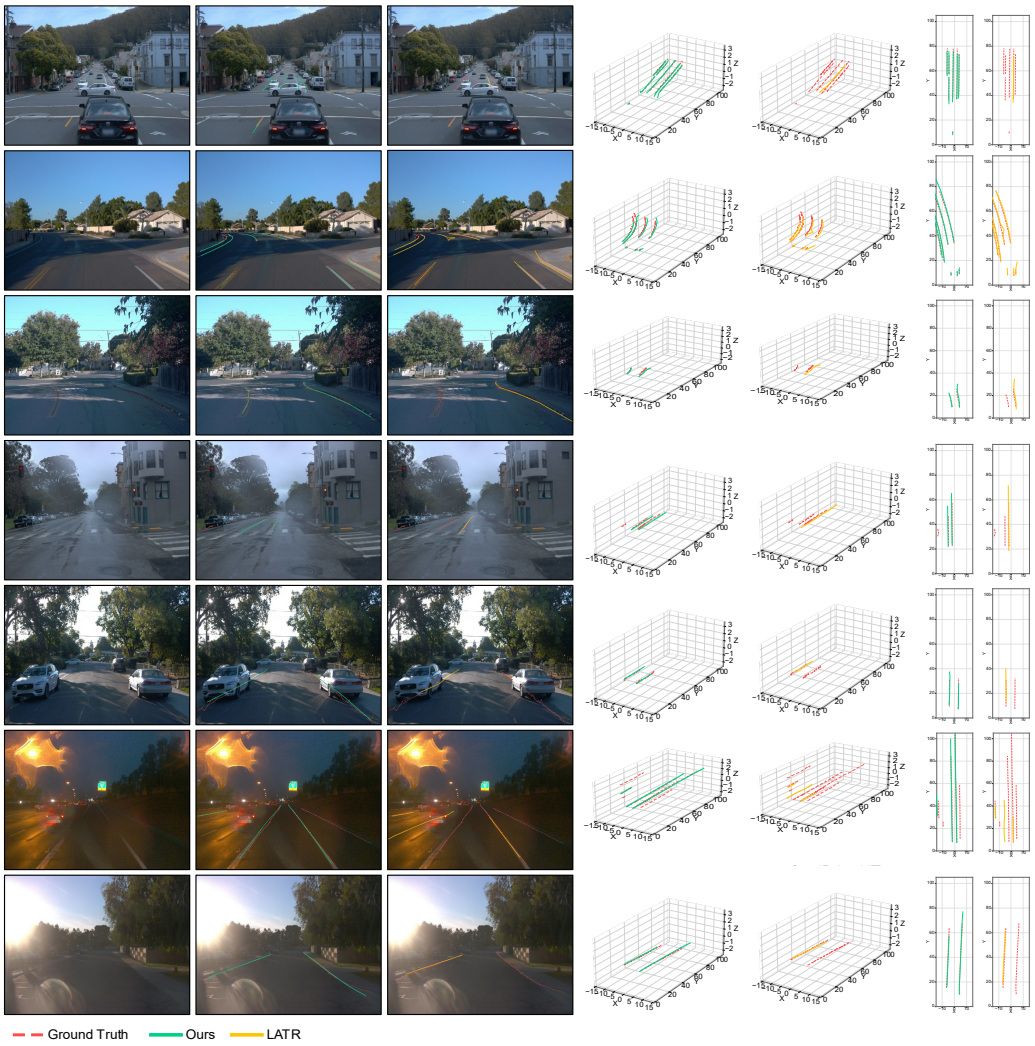


Figure 6: Additional qualitative results on Ev-OpenLane. Red dashed lines denote ground truth, green lines denote our predictions, and yellow lines denote LATR predictions. Our method produces more coherent image-space and 3D lane predictions than LATR across diverse large-scale scenes, showing strong generalization on simulated RGB-event data.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are consistent with the contributions and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the future work section of the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This work does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides experimental details in the main paper and supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: If the paper is accepted, the code and data will be released.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the training and test details in the experimental section and supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report error bars over multiple random seeds due to the high computational cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the compute resources used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses broader impacts in the supplementary material.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped personal data, or assets with obvious dual-use risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the existing assets used in this work and follows their terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper documents the new assets in the main paper and supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.