Selection, Not Fusion: Radar-Modulated State Space Models for Radar-Camera Depth Estimation

Zhangcheng Hou
School of Science and Technology
Keio University
Yokohama 223-8522, Japan
zhangcheng@ohtsuki.ics.keio.ac.jp

Tomoaki Ohtsuki
School of Science and Technology
Keio University
Yokohama 223-8522, Japan
ohtsuki@ics.keio.ac.jp

Abstract

Radar camera depth estimation turns a very sparse but metric radar signal into a dense depth map. Existing methods combine radar and image features outside the sequence operator of the backbone, through concatenation, confidence gating, sparse supervision, or graph based extraction. Mamba variants for multimodal fusion also leave the selection mechanism itself unimodal. We argue that selection is the right place for radar to enter. We introduce Radar-Modulated Selection (RMS), which injects radar inside the selective scan of Mamba. Radar adds learned perturbations, initialised at zero, to the step size $\mathbf{\Delta}$ and the readout $\mathbf{C}$. The input projection $\mathbf{B}$ and the state matrix $\mathbf{A}$ never see radar. At initialisation the block equals a pretrained image Mamba, so radar gains influence only where it improves accuracy. The construction also couples the two modalities at linear cost and falls back to the image backbone when radar is missing. We deploy RMS in a Multi-View Scan Pyramid (MVSP) that matches the fusion operator to the reach of radar at each scale. On nuScenes, SemoDepth reduces MAE by 34.0%, 29.9%, and 29.9% over the previous best within 50, 70, and 80 m. It also has the lowest single frame latency among methods with a public release (26.8 ms). A feature blend placed before the scan adds no accuracy on top of RMS, which supports the view that selection inside the scan can replace fusion outside it. When twice as many spurious radar returns as true ones are injected, the MAE@80 of SemoDepth rises by 1.4%, against 33.6% for a Mamba baseline that fuses radar before the scan.

1 Introduction

Dense metric depth is foundational for autonomous driving Geiger et al. [2013], Caesar et al. [2020] and mobile robotics. It supports obstacle avoidance and localisation Adolfsson et al. [2022] as well as 3D object detection and perception in bird's eye view Lin et al. [2024], Chu et al. [2025]. Monocular cameras are cheap, but they recover depth only up to scale Yang et al. [2024] and degrade in rain, fog, and low light. Light detection and ranging (LiDAR) resolves scale, yet its power and cost confine it to premium vehicles. Automotive millimetre wave radar with frequency modulated continuous waves (FMCW) at 77 to 81 GHz bridges this gap. It is inexpensive, robust to weather, directly metric, and already common in modern vehicles Bilik et al. [2019], Han et al. [2023], Caesar et al. [2020]. The signal is very limited, though. A nuScenes radar sweep holds only 40 to 100 returns in the field of view (FoV) of the front camera, three orders of magnitude fewer than LiDAR. The returns also carry angular and height ambiguity, which emerging 4D radar datasets Palffy et al. [2022], Li et al. [2024a] resolve only in part.

Radar camera depth completion Uhrig et al. [2017] has a decade of designs. They include concatenation Lin et al. [2020], Long et al. [2021], Gasperini et al. [2021], Lo and Vandewalle [2021], Singh et al. [2023], confidence aware gating Sun et al. [2024], sparse supervision Li et al. [2024b],

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

graph based structure extraction Wang et al. [2025], and two stage refinement of metric scale Li et al. [2024a]. In every case radar and image features meet at the feature level, outside the sequence operator of the backbone. In parallel, selective state space models such as Mamba Gu and Dao [2023], Dao and Gu [2024], Vim Zhu et al. [2024], VMamba Liu et al. [2024], and MambaDepth Grigore and Popa [2024] have become an efficient alternative to Transformers in vision. Their core is a selection mechanism, in which a step size $\mathbf{\Delta}$ and a readout $\mathbf{C}$ computed from each token decide what the token keeps and emits. Existing Mamba variants for multimodal fusion Xie et al. [2024], Li et al. [2024c] stack the streams of the two modalities before a shared scan, so the selection itself stays unimodal.

We let radar act on selection itself. Radar-Modulated Selection (RMS) injects radar inside the selective scan of Mamba Gu and Dao [2023], and a Multi-View Scan Pyramid (MVSP) matches the fusion operator to the reach of radar at each decoder resolution. The resulting model, SemoDepth, sets a new state of the art on nuScenes Caesar et al. [2020] for mean absolute error (MAE) and root mean square error (RMSE) at every evaluation range. It also has the best MAE on ZJU-4DRadarCam Li et al. [2024a] among methods without a monocular depth prior. We further show that selection inside the scan can replace fusion outside it, since a feature blend before the scan adds no accuracy on top of RMS. Our contributions are threefold.

- Radar-Modulated Selection (RMS). This mechanism injects radar inside the selective scan of Mamba. Radar adds learned modulations, initialised at zero, to two selection parameters. These are the step size $\mathbf{\Delta}$, which sets the memory horizon of each token, and the readout $\mathbf{C}$, which reads out the state. The input projection $\mathbf{B}$ and the state matrix $\mathbf{A}$ never see radar. Feature level fusion operates on static representations. Selection inside the scan instead modulates the recurrence dynamics that govern long range propagation. Radar then shapes not only how local features are composed but also how memory evolves along the sequence. It changes how information is stored and propagated, not merely how it is combined. Three advantages follow that fusion outside the scan cannot offer. (i) Coupling at linear cost. Radar shapes the recurrence at every token at $O(L)$ cost, never the $O(L^2)$ cost of cross attention. (ii) An initialisation that preserves the loss. At step zero the block equals a vanilla Mamba scan, so the pretrained image solution is preserved exactly and radar gradients grow only where they improve accuracy. (iii) A fallback to the image. With $\mathbf{A}$ and $\mathbf{B}$ unimodal, the model reduces to image behaviour under sparse or missing radar without any change to the architecture.
- Multi-View Scan Pyramid (MVSP). This decoder with three tiers matches the fusion operator to the reach of radar at each resolution. The two coarsest levels use RMS over the full image in four scan directions, a linear cost approximation of the 2D receptive field of a Transformer. The middle level uses RMS in windows around radar returns, so scan capacity is spent only where radar supports the image. The two finest levels use feature wise linear modulation (FiLM) at constant cost. Radar stays active at every decoder level, and scan compute concentrates where radar evidence reaches furthest. MVSP lowers MAE@80 by 18% compared with FiLM at every level, so the allocation carries accuracy and is not merely an efficiency measure.
- State of the art on nuScenes and evidence that selection can replace fusion. SemoDepth sets a new state of the art for MAE and RMSE at every evaluation range and lowers MAE by about 30% over the previous best. Controlled ablations show that modulating $\mathbf{\Delta}$ and $\mathbf{C}$ helps, modulating $\mathbf{B}$ does not improve MAE@80, and a blend before the scan adds nothing. A corruption study confirms the fallback property. With radar removed at test time, MAE@80 rises by only 3.3%.

2 Related work

2.1 Radar camera depth estimation

Methods fall into two families. Two stage approaches such as RadarCam-Depth Li et al. [2024a] adapt a pretrained monocular depth model and learn a correction of its metric scale from radar. The image backbone never sees radar. Single stage methods learn dense depth end to end from radar and image. They differ in how radar enters the image pipeline, usually through feature fusion before or after the backbone operator. Early work Lin et al. [2020], Long et al. [2021], Gasperini et al. [2021], Lo and Vandewalle [2021], Singh et al. [2023] concatenates rasterised radar depth with the image

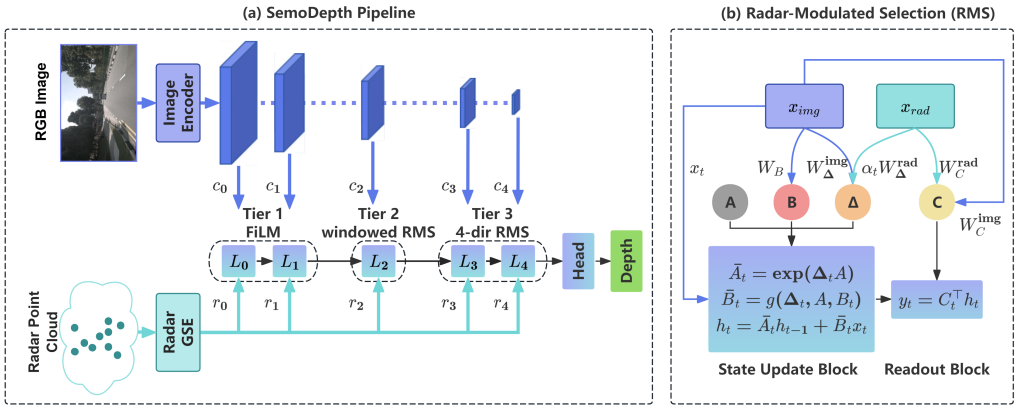


Figure 1: SemoDepth architecture. (a) Pipeline. A ResNet-34 image encoder and a radar graph structure extractor (GSE) built on PCA-GM produce the image pyramid $c_0, \ldots, c_4$ and one radar feature map. Projections (1×1) of this map at each level form the radar pyramid $r_0, \ldots, r_4$. MVSP allocates fusion by resolution. FiLM modulates the two finest levels (Tier 1). Windowed RMS around each projected radar return serves the middle level (Tier 2). RMS over the full image in four directions serves the two coarsest levels (Tier 3). (b) Radar-Modulated Selection (RMS). Radar adds modulations to the step size $\mathbf{\Delta}_t$ and readout $\mathbf{C}_t$ (teal arrows). The input projection $\mathbf{B}_t$ is computed from the image only (blue arrows), and the state matrix $\mathbf{A}$ never sees radar. The state update block applies zero order hold (ZOH) discretization and the recurrence $\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t x_t$. The readout block emits $y_t = \mathbf{C}_t^\top \mathbf{h}_t$. All radar projections start at zero, so the block starts as vanilla Mamba.

channels and lets the network learn the fusion. CaFNet Sun et al. [2024] adds a confidence gate that suppresses uncertain returns before mixing, which addresses the height and azimuth ambiguity of radar. Li et al. Li et al. [2024b] avoid densification and supervise only at pixels that radar observes. TacoDepth Wang et al. [2025] extracts radar structure with a graph network and attends to image features through a pyramid of flash attention centred on radar points.

2.2 Selective state space models in vision

Selective state space models (SSMs), introduced by Mamba Gu and Dao [2023], extend linear time sequence models Gu et al. [2021] by computing the recurrence parameters from each token. A selective gate controls the memory horizon of each token, and a readout projection decides what leaves the hidden state. Vim Zhu et al. [2024], VMamba Liu et al. [2024], and MambaDepth Grigore and Popa [2024] adopt this design as a Transformer replacement for vision. Multimodal variants such as FusionMamba Xie et al. [2024] and MambaDFuse Li et al. [2024c] stack or concatenate features before a shared Mamba block. Their selection parameters $\mathbf{\Delta}$ and $\mathbf{C}$ come from one fused representation and are never conditioned on each modality separately. This limits cross modal control within the selective dynamics that govern long range dependencies. SemoDepth computes $\mathbf{\Delta}$ and $\mathbf{C}$ from both modalities, while the token stream and the input projection $\mathbf{B}$ never see radar.

3 SemoDepth

3.1 Overview

SemoDepth is a single stage depth estimator for radar and camera (Fig. 1). A ResNet-34 encoder pretrained on ImageNet produces five feature maps $c_0, \ldots, c_4$ at strides $1/2$ to $1/32$ of the input. In parallel, a radar graph structure extractor (GSE) that follows TacoDepth Wang et al. [2025] processes up to 512 radar returns per frame. Each return is described only by its geometry $(y/H,\ x/W,\ d/d_{\max})$, where H and W are the image height and width and d is the metric range of the return. The GSE is a PCA-GM graph network with full adjacency Wang et al. [2019], originally developed for deep graph matching, and its output is scattered into a radar feature map. Image and radar streams stay separate until the selective scan. The radar map is downsampled to each decoder resolution and

passed through a 1×1 projection for that level, which yields the radar pyramid $r_0, \ldots, r_4$. Decoder level i fuses c_i with r_i through MVSP (§3.3). A regression head maps the finest decoder feature to metric depth at every pixel.

3.2 Radar-Modulated Selection (RMS)

We first describe RMS conceptually. The selective scan computes, for each token, a step size that controls how long the content of the token persists in the recurrence. It also computes a readout vector that controls what is exposed from the latent state. In a vanilla cross modal Mamba, both come from a single fused stream of image and radar. Radar then enters only as an additional input and never as a control over selection. RMS modulates the step size and the readout directly. At every token, radar contributes a learned correction to each, initialised at zero, while the input projection and the state matrix never see radar. The rest of this section makes the construction precise and defines all symbols of the scan in one place.

Let $\mathbf{x}_{\text{img},t}, \mathbf{x}_{\text{rad},t} \in \mathbb{R}^D$ be the image and radar tokens at position t of a scanned sequence of length L with D channels. A vanilla Mamba block Gu and Dao [2023] runs a selective recurrence on the image stream, separately for each channel. A scalar input x_t, one channel of $\mathbf{x}_{\text{img},t}$, drives a hidden state $\mathbf{h}_t \in \mathbb{R}^N$ and produces a scalar output y_t,

$$\mathbf{h}_t = \bar{\mathbf{A}}_t\, \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t\, x_t, \qquad y_t = \mathbf{C}_t^\top \mathbf{h}_t, \qquad t = 1, \ldots, L. \tag{1}$$

ZOH discretization gives $\bar{\mathbf{A}}_t = \exp(\boldsymbol{\Delta}_t \mathbf{A})$ and $\bar{\mathbf{B}}_t = (\boldsymbol{\Delta}_t \mathbf{A})^{-1}\big(\exp(\boldsymbol{\Delta}_t \mathbf{A}) - \mathbf{I}\big)\, \boldsymbol{\Delta}_t \mathbf{B}_t$. Here $\mathbf{A} \in \mathbb{R}^{D\times N}$ is a trained matrix with negative entries, shared by all tokens, and $\mathbf{I}$ is the identity. Three quantities depend on the token. The step size $\boldsymbol{\Delta}_t \in \mathbb{R}^D$ sets the memory horizon. The input projection $\mathbf{B}_t \in \mathbb{R}^N$ controls how the input enters the state. The readout $\mathbf{C}_t \in \mathbb{R}^N$ selects which part of the state is emitted. Vanilla Mamba computes all three from the image stream with trained linear maps,

$$\boldsymbol{\Delta}_t = \operatorname{softplus}\big(\mathbf{W}_\Delta^{\text{img}}\, \mathbf{x}_{\text{img},t}\big), \qquad \mathbf{B}_t = \mathbf{W}_B\, \mathbf{x}_{\text{img},t}, \qquad \mathbf{C}_t = \mathbf{W}_C^{\text{img}}\, \mathbf{x}_{\text{img},t}. \tag{2}$$

From a control theoretic view, these three parameters play distinct roles. $\mathbf{B}_t$ governs controllability, which is how inputs enter the system. $\mathbf{C}_t$ governs observability, which is the part of the hidden state that is exposed. $\boldsymbol{\Delta}_t$ sets the memory timescale of the recurrence. RMS modulates $\boldsymbol{\Delta}_t$ and $\mathbf{C}_t$ and keeps $\mathbf{B}_t$ driven by the image. This preserves the semantic content of image features. Radar influences temporal credit assignment through $\boldsymbol{\Delta}_t$ and observation through $\mathbf{C}_t$, the two basic degrees of freedom over selection.

3.2.1 Horizon modulation

The horizon pathway adds a gated radar term to the step size before the softplus,

$$\boldsymbol{\Delta}_t = \operatorname{softplus}\big(\mathbf{W}_\Delta^{\text{img}}\, \mathbf{x}_{\text{img},t} + \alpha_t\, \mathbf{W}_\Delta^{\text{rad}}\, \mathbf{x}_{\text{rad},t}\big), \qquad \alpha_t = \sigma\big(\mathbf{W}_g\, \mathbf{x}_{\text{rad},t} + b_g\big). \tag{3}$$

Here σ is the logistic sigmoid, the scalar gate α_t scales every channel, $\mathbf{W}_\Delta^{\text{rad}}$ has the output shape of $\mathbf{W}_\Delta^{\text{img}}$, and $\mathbf{W}_g \in \mathbb{R}^{1\times D}$ and $b_g \in \mathbb{R}$ form the gate. We initialise $\mathbf{W}_\Delta^{\text{rad}}$ and $\mathbf{W}_g$ at zero and set $b_g = -2$, so the gate opens at $\sigma(-2) \approx 0.12$ and the radar term starts at zero. A small step size makes $\bar{\mathbf{A}}_t$ close to the identity and retains the state over long ranges. A large step size resets the state and focuses on the current token Gu and Dao [2023]. The horizon pathway lets radar choose this balance locally. The small initial gate keeps the modulation latent at the start, so training opens it only if it helps.

3.2.2 Readout modulation

The readout pathway adds a radar term to $\mathbf{C}_t$,

$$\mathbf{C}_t = \mathbf{W}_C^{\text{img}}\, \mathbf{x}_{\text{img},t} + \mathbf{W}_C^{\text{rad}}\, \mathbf{x}_{\text{rad},t}, \tag{4}$$

where $\mathbf{W}_C^{\text{rad}}$ is initialised at zero. Horizon modulation changes what the state stores. Readout modulation changes what each token reports through $y_t = \mathbf{C}_t^\top \mathbf{h}_t$ and leaves the stored state untouched. No gate is needed because the projection grows only under gradient.

3.2.3 Joint modulation

The full RMS block applies Eq. (3) and Eq. (4) in one Mamba layer. At initialisation $\mathbf{W}_{\Delta}^{\text{rad}} = \mathbf{W}_{C}^{\text{rad}} = \mathbf{0}$, so

$$\boldsymbol{\Delta}_t\big|_{\text{init}} = \text{softplus}\big(\mathbf{W}_{\Delta}^{\text{img}}\,\mathbf{x}_{\text{img},t}\big), \qquad \mathbf{C}_t\big|_{\text{init}} = \mathbf{W}_{C}^{\text{img}}\,\mathbf{x}_{\text{img},t}. \tag{5}$$

This reproduces the image selection of Eq. (2) exactly. The recurrence equals baseline Mamba at step zero, which we verify numerically.

3.3 Multi-View Scan Pyramid (MVSP)

Applying RMS at every decoder level is inefficient. Scan cost grows linearly with the number of tokens and exceeds 10^5 tokens per direction at the finest levels. The spatial support of radar shrinks with scale, so the heaviest compute would land where radar helps least. MVSP matches the operator at each level to the reach of radar evidence. At the two coarsest levels, RMS runs over the full image in four directions and gathers context across the scene. This approximates a 2D receptive field at linear cost. At the middle level, RMS runs only in windows centred on projected radar returns. At the two finest levels, FiLM modulates features with radar at constant cost per pixel. Every level also ends with a radar FiLM block initialised at identity, which is the only fusion operator at the two finest levels. Scan compute goes where radar evidence reaches furthest.

3.4 Training objective

The objective combines robust supervision from a dense but noisy target with accurate supervision from a clean sparse one. Let $\mathbf{D}_{\text{pred}}$, $\mathbf{D}_{\text{main}}$, and $\mathbf{D}_{\text{sparse}}$ be the prediction, the main target, and an optional clean sparse target. On nuScenes, $\mathbf{D}_{\text{main}} = \mathbf{D}_{\text{acc}}$ is accumulated over 160 LiDAR sweeps and is dense but noisy, and $\mathbf{D}_{\text{sparse}} = \mathbf{D}_{\text{gt}}$ is a single clean sweep. On ZJU-4DRadarCam, $\mathbf{D}_{\text{main}}$ is the interpolated dense depth released with the dataset, and no $\mathbf{D}_{\text{sparse}}$ exists. Depths are normalised in log space to $\tilde{d} \in [-1, 1]$ over $[d_{\min}, d_{\max}]$, and each loss uses only the valid pixels of its target. The objective is

$$\mathcal{L} = \lambda_{\text{log}}\mathcal{L}_{\text{log}} + \lambda_{\text{lin}}\mathcal{L}_{\text{lin}} + \lambda_{\text{grad}}\mathcal{L}_{\text{grad}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}}. \tag{6}$$

$\mathcal{L}_{\text{log}}$ is an L1 loss in log space against $\mathbf{D}_{\text{sparse}}$, or against $\mathbf{D}_{\text{main}}$ when no sparse target exists. $\mathcal{L}_{\text{lin}}$ is a Huber loss ($\delta = 5\,\text{m}$) against $\mathbf{D}_{\text{main}}$. $\mathcal{L}_{\text{grad}}$ is an L1 loss on the x and y gradients of $\tilde{\mathbf{D}}_{\text{pred}} - \tilde{\mathbf{D}}_{\text{main}}$. $\mathcal{L}_{\text{sparse}}$ is an L1 loss in metres against $\mathbf{D}_{\text{sparse}}$, as in the sparse term of TacoDepth Wang et al. [2025]. Both linear losses are divided by $d_{\max}$. Log space is used only on the clean target. Beyond 5 m the Huber loss grows linearly, which bounds the gradient of large residuals so that rare large residuals do not dominate training. We set $\lambda_{\text{log}} = \lambda_{\text{lin}} = \lambda_{\text{sparse}} = 1$ and $\lambda_{\text{grad}} = 0.5$ unless stated otherwise.

4 Experiments

We conduct experiments on the nuScenes Caesar et al. [2020] and ZJU-4DRadarCam Li et al. [2024a] datasets, following prior radar camera depth work Li et al. [2024a], Wang et al. [2025], Sun et al. [2024], Li et al. [2024b], Singh et al. [2023], Lin et al. [2020], Long et al. [2021], Gasperini et al. [2021], Lo and Vandewalle [2021].

4.1 Datasets and evaluation protocols

The nuScenes dataset Caesar et al. [2020] has 1,000 driving scenes with about 40,000 synchronised keyframes from Boston and Singapore, recorded with a vehicle platform carrying a camera, 3D radar, LiDAR, and an inertial measurement unit. We use the standard split of 700, 150, and 150 scenes for training, validation, and test from prior work Li et al. [2024a], Wang et al. [2025], Sun et al. [2024], Li et al. [2024b], Singh et al. [2023]. Front camera images are 900×1600, and each radar frame has 40 to 100 returns in the camera FoV.

ZJU-4DRadarCam Li et al. [2024a] has 33,409 keyframes from a ground robot with a camera, a 4D radar, and a denser LiDAR. We use the official split of 29,312 frames for training and validation and 4,097 for test. As in prior work Li et al. [2024a], Wang et al. [2025], we crop images from 720×1280 to 300×1280 to remove the hood and sky.

Table 1: Comparison with published radar camera depth methods on nuScenes Caesar et al. [2020]. Time is single frame latency, measured on identical hardware for methods with a runnable release. Baseline MAE and RMSE come from the original publications. †Our reimplementation of FusionMamba for depth completion, trained with the supervision, optimiser, and loss of SemoDepth at a matched parameter budget. SemoDepth reports the mean and standard deviation over three seeds. Best per column in bold.

Method	Time (ms)↓	0–50 m		0–70 m		0–80 m	
		MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓
Lin et al. Lin et al. [2020] (IROS'20)	–	2034.9	4316.5	2294.7	5338.2	2371.0	5623.0
RC-PDA Long et al. [2021] (CVPR'21)	145.0	2225.0	4156.5	3326.1	6700.6	3713.6	7692.8
R4Dyn Gasperini et al. [2021] (3DV'21)	–	–	–	–	–	–	6434.0
DORN Lo and Vandewalle [2021] (ICIP'21)	60.9	1926.6	4124.8	2380.6	5252.7	2467.7	5554.3
Singh et al. Singh et al. [2023] (CVPR'23)	40.9	1727.7	3746.8	2073.2	4590.7	2179.3	4898.7
CaFNet Sun et al. [2024] (IROS'24)	44.2	1674.2	3674.5	2010.3	4493.1	2109.8	4765.6
Li et al. Li et al. [2024b] (ECCV'24)	46.4	1524.5	3567.3	1822.9	4303.6	1927.0	4609.6
FusionMamba† Xie et al. [2024] (VI'24)	–	1179.9	3337.2	1483.3	4282.5	1590.5	4640.0
TacoDepth Wang et al. [2025] (CVPR'25)	–	1423.6	3275.8	1712.6	3960.5	1833.4	4150.2
SemoDepth (Ours)	**26.8**	$\mathbf{940.1}_{\pm 4.1}$	$\mathbf{2785.7}_{\pm 2.5}$	$\mathbf{1199.8}_{\pm 1.5}$	$\mathbf{3622.4}_{\pm 16.5}$	$\mathbf{1285.2}_{\pm 0.9}$	$\mathbf{3935.4}_{\pm 21.5}$

Following prior work, we evaluate against sparse LiDAR from a single sweep within 50, 70, and 80 m Li et al. [2024a], Wang et al. [2025], Sun et al. [2024], Li et al. [2024b]. MAE and RMSE are the primary metrics. On ZJU and in the ablations we also report the inverse depth metrics iMAE and iRMSE, computed on $1/\hat{D}$ and $1/D_{\text{gt}}$ in km^{-1}. These metrics complement MAE and RMSE by weighting errors at short range more heavily.

4.2 Implementation details

On nuScenes, $\mathbf{D}_{\text{acc}}$ accumulates $\mathbf{D}_{\text{gt}}$ over 160 nearby LiDAR sweeps following Wang et al. [2025], Li et al. [2024b]. A light filter on the upper image removes accumulation artefacts from this target, and model inputs are unchanged. On ZJU-4DRadarCam, $\mathbf{D}_{\text{acc}}$ is the interpolated dense depth from Li et al. [2024a] without filtering. The radar GSE uses three PCA-GM layers with 64 hidden channels. Each of the two coarsest MVSP levels stacks two RMS layers. The middle level uses windows of size $w = 8$ with Gaussian scatter weights of standard deviation $w/2.5$. We train with Adam at a learning rate of 10^{-4} and the step schedule of TacoDepth Wang et al. [2025], which lowers the rate by 10^{-5} every ten epochs down to half its initial value. The batch size is 12 with bf16 autocast. Each run trains for at most 50 epochs. We evaluate every 2,000 steps, keep the checkpoint with the lowest MAE@80, and end a run once this metric has saturated. The depth range $[d_{\min}, d_{\max}]$ is $[0.5, 120]$ m on nuScenes and $[0.5, 80]$ m on ZJU-4DRadarCam. The ceiling of 120 m keeps the roughly 15% of nuScenes radar returns between 80 and 120 m away from an artificial boundary. Evaluation stops at 80 m on both datasets. Table 1 reports the mean and standard deviation over three training seeds. Other experiments use one seed unless stated otherwise.

4.3 Comparisons with state of the art methods

Table 1 compares SemoDepth with eight published methods on nuScenes under the protocol of TacoDepth Wang et al. [2025], with the same split, 900×1600 input, LiDAR ground truth, and evaluation ranges. Published results are taken from the original papers. Our evaluation script reproduces the published CaFNet result from its public weights (2116 against 2109.8 mm MAE@80). We also reimplement FusionMamba Xie et al. [2024] (marked †) as a modern Mamba fusion baseline. It was proposed for multimodal image fusion and has no radar camera depth result. We adapt its dual encoder backbone (VSSM_Fusion) with cross modal SS2D fusion at every scale. We train it with the supervision, optimiser, loss, and schedule of SemoDepth at a matched budget of about 38 M parameters. Latency is measured for every method with a runnable release on identical hardware (NVIDIA RTX PRO 6000 Blackwell, batch 1, FP32, 900×1600 input, mean over 100 iterations after 10 warmup iterations).

SemoDepth has the lowest MAE and RMSE at every range. Across the published methods in Table 1, MAE@80 fell by about 540 mm in five years, from 2371 mm (Lin et al.) to 1833 mm (TacoDepth). SemoDepth lowers it by another 548 mm in one step. The gain does not come from our supervision configuration. Trained on the raw target accumulated over 160 sweeps without the gradient term, SemoDepth reaches MAE@80 of 1311.9 mm, 28.4% below TacoDepth. FusionMamba† reaches

Table 2: Comparison on ZJU-4DRadarCam Li et al. [2024a]. Baseline numbers are as reported by TacoDepth Wang et al. [2025]. No method here uses a monocular prior. Best per column in bold.

Method	0–50 m		0–70 m		0–80 m	
	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓
DORN Lo and Vandewalle [2021]	2210.2	4129.7	2402.2	4625.2	2447.6	4760.0
Singh et al. Singh et al. [2023]	1785.4	3704.6	1932.7	4137.1	1979.5	4309.3
TacoDepth Wang et al. [2025]	1120.1	2686.7	1181.8	**2906.3**	1201.1	**2990.7**
SemoDepth (Ours)	**1029.2**	**2631.6**	**1111.6**	2946.9	**1137.2**	3053.0

Table 3: Inference efficiency on nuScenes Caesar et al. [2020]. Time follows the protocol of Table 1. TacoDepth has no public release. Params and FLOPs come from the TacoDepth paper for DORN, Singh et al., and TacoDepth. We measure them with thop ($2\times$MACs) for CaFNet and Li et al. The FLOPs of SemoDepth are a lower bound from fvcore. Best per column in bold.

Method	Params (M) ↓	FLOPs (G) ↓	Time (ms) ↓
DORN Lo and Vandewalle [2021] (ICIP'21)	183.4	923.5	60.9
Singh et al. Singh et al. [2023] (CVPR'23)	22.8	502.1	40.9
CaFNet Sun et al. [2024] (IROS'24)	62.2	1426.6	44.2
Li et al. Li et al. [2024b] (ECCV'24)	36.9	**31.0**	46.4
TacoDepth Wang et al. [2025] (CVPR'25)	**13.5**	139.3	–
SemoDepth (Ours)	38.1	271.8	**26.8**

1587.8 mm with the raw configuration and 1590.5 mm with the cleaned one. The gap between the two architectures is 17.4% with raw and 19.2% with cleaned supervision.

Two patterns in Table 1 tie the result to the design claims of §3. MAE gains are more than twice the RMSE gains at every range (34.0% against 15.0% within 50 m, 29.9% against 8.5% within 70 m, and 29.9% against 5.2% within 80 m). MAE reflects the typical pixel, while RMSE stresses rare large errors, so the gain accrues mostly to the typical pixel rather than the worst case. This is the behaviour expected from selection inside the scan. By reshaping $\mathbf{\Delta}$ and $\mathbf{C}$ at every token, RMS lets the metric anchor of each radar return propagate along the whole sequence rather than only near its projected pixel. Such reshaping at the level of tokens is hard to achieve with fusion outside the scan.

The second pattern is the gap to TacoDepth, the most informative comparison across methods. The windowed flash attention of TacoDepth around radar points is architecturally closest to the windowed RMS of our middle tier. A steady MAE margin of about 30% at every range is unlikely to come from windowed processing alone. It is consistent with the two tiers that distinguish MVSP. The first is the scan across the whole scene at the coarsest levels, which has no analogue in the pyramid of TacoDepth. The second is the FiLM at constant cost that keeps radar active at the finest levels, where the pyramid of TacoDepth attenuates it. FusionMamba† gives a second informative comparison. Trained with matched supervision, optimiser, and loss, it reaches MAE@80 of 1591 mm, between TacoDepth and SemoDepth. A Mamba backbone alone closes part of the gap, but a margin of 19.2% MAE@80 remains. With the backbone family fixed and the training pipeline matched, this margin isolates the contribution of selection inside the scan over fusion before it. It is the distinction drawn in §2.2, now visible as a measured gap rather than a design argument. Fig. 2 shows examples at night, by day, and in rain against the three baselines with public code (CaFNet Sun et al. [2024], Li et al. Li et al. [2024b], and Singh et al. Singh et al. [2023]).

On ZJU-4DRadarCam (Table 2), every method uses one radar scan and one image per frame, the same released training target, and the same sparse evaluation. SemoDepth has the best MAE at every range and the best RMSE within 50 m. TacoDepth keeps a small edge on RMSE within 70 m (1.4%) and 80 m (2.1%). Among methods that also consume a monocular depth prior, the best of five RadarCam-Depth configurations Li et al. [2024a] reaches 1183.5 mm MAE@80 and 3229.0 mm RMSE@80, and SemoDepth beats both without such a prior. TacoDepth with a monocular prior remains stronger at 1032.5 mm MAE@80. The RMSE gap to TacoDepth comes from a small far tail. Pixels beyond 70 m make up 0.4% of our test pixels but 7.5% of the squared error, and SemoDepth under predicts them by 7.3 m on average between 70 and 80 m. For the 78.6% of pixels within 20 m, the median error is below 0.3 m. On inverse depth, which weights the far tail less, SemoDepth leads TacoDepth by 18% on iMAE and 8% on iRMSE.

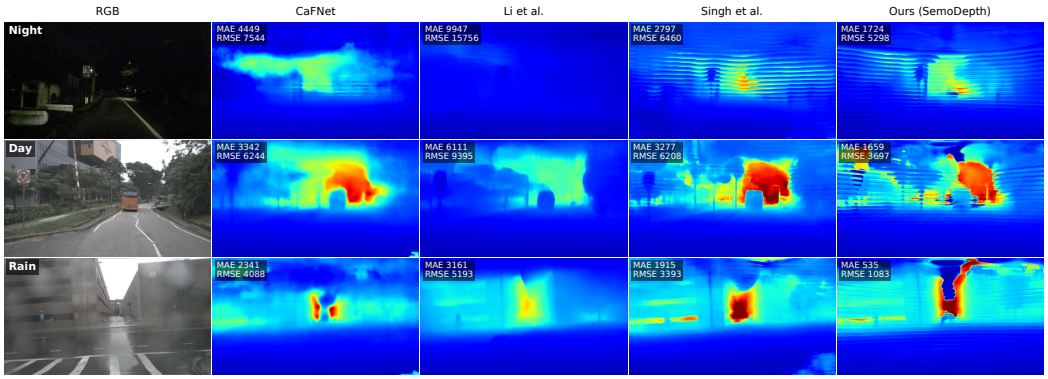


Figure 2: Qualitative comparison on nuScenes Caesar et al. [2020]. Rows show night (frame 5319), day (frame 161), and rain (frame 3088). Columns show the RGB input, CaFNet Sun et al. [2024], Li et al. Li et al. [2024b], Singh et al. Singh et al. [2023], and SemoDepth. Each panel gives the MAE and RMSE of the frame in millimetres against sparse LiDAR within 80 m, and the colour map is clipped at 80 m. SemoDepth has the lowest MAE and RMSE on every frame. The largest margin is in rain, with 535 mm MAE against 1915 mm for the next best baseline.

Table 4: Ablation on nuScenes Caesar et al. [2020]. The upper block changes the decoder. Uniform FiLM replaces the scans of Tiers 2 and 3 with FiLM at every level. Full RMS to level 2 replaces the windowed scan of Tier 2 with the full image scan. The lower block changes which selection parameters radar modulates inside the MVSP decoder. In the row without radar in the scan, radar still reaches the decoder through FiLM. The blend row adds a confidence gated blend in the style of FusionMamba Xie et al. [2024], Li et al. [2024c] before the scan. Best per column in bold.

	0–50 m		0–70 m		0–80 m	
Variant	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓
Uniform FiLM	1094	3075	1437	4166	1564	4668
Full RMS to level 2	**919**	**2728**	1206	3649	1304	4011
No radar in the scan	950	2787	1217	3674	1306	4004
B only	933	2782	1217	3677	1315	4025
Δ only (horizon)	964	2755	1251	3614	1352	3953
C only (readout)	956	2792	1227	3672	1317	4012
Δ+**B**+**C**	967	2802	1267	3748	1360	4092
Δ+**C** with blend before the scan	940	2749	1211	3657	1304	4019
Δ+**C** (joint, main)	943	2784	**1201**	**3611**	**1286**	**3920**

4.4 Inference efficiency

SemoDepth has the lowest latency among methods with a public release (26.8 ms), 1.5 to 2.3 times faster than Singh et al., CaFNet, Li et al., and DORN (Table 3). It has more parameters than TacoDepth and sits in the middle on FLOPs. MVSP keeps scan compute at the coarse levels.

4.5 Ablation studies

Table 4 varies the decoder and the site where radar enters the scan. All rows share the ResNet-34 encoder, the radar GSE, the training recipe of §4.2, and the depth range.

MVSP is critical for accuracy, not only for speed. The Uniform FiLM row replaces the scans of Tiers 2 and 3 with FiLM at every level and keeps every other component fixed. MVSP lowers MAE@80 from 1564 to 1286 mm (18%), so the pyramid plays a central role in accuracy. We also extend the full image scan with four directions down the decoder (Table 5). At level 2 it replaces the windowed scan, and at levels 1 and 0 it replaces FiLM. Each finer level quadruples the number of tokens. Extending the scan to level 2 gains 2.6% MAE within 50 m. It loses 1.4% MAE@80 and 2.3% RMSE@80 and adds 27% latency and 4.6 M parameters, so it is worse on the headline metrics. The scan at fine resolution sharpens nearby structure but dilutes the advantage at long range.

Table 5: Full RMS at finer decoder levels on nuScenes Caesar et al. [2020]. Latency is relative to MVSP, measured at batch 1 and FP32. Levels 1 and 0 are compared on cost only.

Full RMS down to level	Latency	Peak memory (GB)	Params (M)	MAE@80 (mm)	RMSE@80 (mm)
3 (MVSP, main)	1.00×	2.16	38.1	1285.8	3920.2
2	1.27×	2.17	42.7	1303.8	4011.4
1	3.15×	3.59	47.9	–	–
0	10.53×	11.66	53.1	–	–

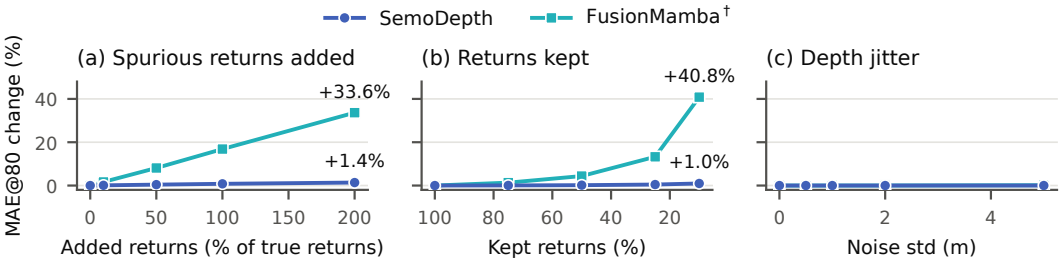


Figure 3: Change in MAE@80 on nuScenes under radar corruption at test time. (a) Spurious returns added, as a share of the true returns. (b) Share of true returns kept. (c) Gaussian noise on the depth of true returns. Both models see identical corrupted inputs. Removing radar entirely costs SemoDepth 3.3% and FusionMamba† 155%.

Levels 1 and 0 multiply latency by 3.1 and 10.5, and level 0 needs 5.4 times the memory. The MVSP allocation is the better balance of accuracy and cost.

Both RMS pathways contribute. Within MVSP, horizon modulation and readout modulation each recover much of the gap between Uniform FiLM and the joint model, so the pathways are not redundant. Their behaviour is complementary. Horizon modulation (§3.2.1) controls how information flows through the recurrence. Readout modulation (§3.2.2) controls what is extracted at the emission step. Their combination is needed for full cross modal expressivity. Joint modulation attains the lowest MAE@80 (1286 mm against 1352 for horizon only and 1317 for readout only) and the lowest RMSE@80 (3920 mm against 3953 and 4012). On RMSE within 50 m, horizon modulation alone edges out the joint model by about 1%, which we attribute to minor optimisation variance.

The input projection should stay driven by the image. With radar only in $\mathbf{B}$, MAE@80 is 1315 mm, within 9 mm of the variant without radar in the scan (1306 mm). Adding $\mathbf{B}$ to $\mathbf{\Delta}$ and $\mathbf{C}$ raises MAE@80 by 5.8% to 1360 mm, worse than no radar in the scan. Radar in $\mathbf{B}$ alone does lower MAE within 50 m, but it does not carry to longer ranges. These results support keeping the input projection driven by the image.

Fusion before the scan adds nothing on top of RMS. The blend row adds a FusionMamba style blend before the scan to the joint model. It reaches 1304 mm MAE@80 against 1286 mm for the joint model and leads only within 50 m. Over three seeds, the blend gives 1305.3 ± 11.5 mm MAE@80 and the joint model gives 1285.2 ± 0.9 mm. The blend is worse on average and an order of magnitude less stable across seeds. Once radar modulates selection inside the scan, an extra pathway before the scan adds no signal at the ranges that drive the headline metrics. This complements the comparison with the full FusionMamba backbone in Table 1.

4.6 Robustness to radar corruption and conditions

We test whether selection filters out corrupted radar. A deterministic procedure corrupts the radar of each nuScenes validation frame with a seed tied to the frame index, so every model sees identical inputs. We compare SemoDepth with FusionMamba†, which shares supervision, loss, recipe, and parameter budget but fuses radar before the scan. Spurious returns are placed at random image positions with random depths between 0.5 and 120 m. Removal keeps a random subset of true returns. Depth noise perturbs the true returns with Gaussian noise (Fig. 3). With twice as many spurious

Table 6: nuScenes validation frames split by condition, within 80 m. A scene counts as night or rain when its description contains that word. Rain overlaps with day and night. Both models are those evaluated in Fig. 3. Best per row and metric in bold.

	SemoDepth			FusionMamba†		
Split (frames)	MAE (mm)	RMSE (mm)	iMAE	MAE (mm)	RMSE (mm)	iMAE
Day (5,282)	**1177.0**	**3731.1**	**3.96**	1530.0	4532.8	5.06
Night (587)	2252.0	5925.5	**7.41**	**2135.2**	**5605.2**	7.83
Rain (1,061)	**1681.7**	**4834.3**	**4.78**	1980.3	5409.6	5.85
All (5,869)	**1284.5**	**3950.6**	**4.30**	1590.5	4640.0	5.34

returns as true ones, SemoDepth loses 1.4% MAE@80 and FusionMamba† loses 33.6%. Keeping 10% of the returns costs 1.0% and 40.8%. Noise with a standard deviation of 5 m costs at most 0.2% for either model, so both depend on where returns are rather than on their exact depth. With radar removed, SemoDepth loses 3.3% and reaches 1327 mm, still below every published method in Table 1. Inside the scans, radar acts only through modulations of $\mathbf{\Delta}$ and $\mathbf{C}$ that start at zero and never enters the token stream or $\mathbf{B}$. Elsewhere it enters through FiLM blocks that start at identity. These narrow, learned channels are consistent with the small effect of corrupted radar.

Table 6 splits the validation frames by condition. The night split follows TacoDepth Wang et al. [2025]. nuScenes contains no fog scenes, so fog cannot be evaluated. SemoDepth leads by 23% MAE@80 by day and by 15% in rain, where the camera degrades and radar does not. At night FusionMamba† is about 5% better on MAE@80 and RMSE@80, while SemoDepth keeps the lower iMAE. The night split holds only 587 frames, and its result varies with the training seed. A second seed of the main configuration reaches 2146 mm MAE@80 at night.

5 Limitations

Four limits qualify our claims. First, at night the camera signal is weak. On the 587 night frames, fusion before the scan can match or beat SemoDepth on linear metrics, and the outcome varies with the seed. Second, supervision on nuScenes comes from LiDAR accumulated over 160 sweeps, which leaves trails behind moving objects. Our filtered target, used with gradient matching, lowers RMSE@80 by 7 to 10% for both architectures relative to the raw configuration, yet some artefacts remain visible in the predicted maps. Third, published methods are compared through their reported numbers. TacoDepth has no public release, so we control for training effects with FusionMamba†, retrained under matched settings with both the raw and the cleaned target. Fourth, SemoDepth under predicts far depth. On ZJU-4DRadarCam the mean bias between 70 and 80 m is -7.3 m, and this tail explains its RMSE gap at 70 and 80 m. We have not tested other sensor pairs, fog, or embedded hardware. Dense depth errors can propagate to planning, so any deployment needs evaluation by scene type and monitoring of inputs that differ from the training data.

6 Conclusion

We argued that radar should enter a Mamba depth model through its selection mechanism rather than its input stream. Radar-Modulated Selection and the Multi-View Scan Pyramid realise this idea. SemoDepth sets a new state of the art on nuScenes for MAE and RMSE at every range and has the lowest latency among methods with a public release. Controlled studies show that modulating $\mathbf{\Delta}$ and $\mathbf{C}$ helps, modulating $\mathbf{B}$ does not improve MAE@80, and a blend before the scan adds nothing. Under spurious, sparse, or missing radar, SemoDepth degrades far less than a Mamba model that fuses radar before the scan. In sequence models whose recurrence depends on the input, the selection mechanism itself can serve as the interface between modalities. We expect this idea to extend to other pairs of sparse and dense sensors, such as LiDAR and camera or radar and thermal imaging, and to tasks beyond depth. Our findings suggest that cross modal learning in sequence models should be reconsidered from the perspective of selection rather than representation. This opens a new direction for multimodal sequence modelling beyond vision.

Acknowledgments and Disclosure of Funding

This work was supported by Japan Science and Technology Agency (JST) Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) Grant Number JPMJAP2326, Japan. The authors declare no competing interests.

References

Daniel Adolfsson, Martin Magnusson, Anas Alhashimi, Achim J Lilienthal, and Henrik Andreasson. Lidar-level localization with radar? the cfear approach to accurate, fast, and robust large-scale radar odometry in diverse environments. *IEEE Transactions on robotics*, 39(2):1476–1495, 2022.

Igal Bilik, Oren Longman, Shahar Villeval, and Joseph Tabrikian. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE signal processing Magazine*, 36(5):20–31, 2019.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

Xiaomeng Chu, Jiajun Deng, Guoliang You, Yifan Duan, Houqiang Li, and Yanyong Zhang. Racformer: Towards high-quality 3d object detection via query-based radar-camera fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17081–17091, 2025.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Ionuţ Grigore and Călin-Adrian Popa. Mambadepth: Enhancing long-range dependency for self-supervised fine-structured monocular depth estimation. *arXiv preprint arXiv:2406.04532*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Zeyu Han, Jiahao Wang, Zikun Xu, Shuocheng Yang, Lei He, Shaobing Xu, Jianqiang Wang, and Keqiang Li. 4d millimeter-wave radar in autonomous driving: A survey. *arXiv preprint arXiv:2306.04242*, 2023.

Han Li, Yukai Ma, Yaqing Gu, Kewei Hu, Yong Liu, and Xingxing Zuo. Radarcam-depth: Radar-camera fusion for depth estimation with learned metric scale. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10665–10672. IEEE, 2024a.

Huadong Li, Minhao Jing, Wang Jin, Shichao Dong, Jiajun Liang, Haoqiang Fan, and Renhe Ji. Sparse beats dense: Rethinking supervision in radar-camera depth completion. In *European Conference on Computer Vision*, pages 127–143. Springer, 2024b.

Zhe Li, Haiwei Pan, Kejia Zhang, Yuhua Wang, and Fengming Yu. Mambadfuse: A mamba-based dual-phase model for multi-modality image fusion. *arXiv preprint arXiv:2404.08406*, 2024c.

Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10233–10240. IEEE, 2020.

Zhiwei Lin, Zhe Liu, Zhongyu Xia, Xinhao Wang, Yongtao Wang, Shengxiang Qi, Yang Dong, Nan Dong, Le Zhang, and Ce Zhu. Rcbevdet: Radar-camera fusion in bird's eye view for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14928–14937, 2024.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

Chen-Chou Lo and Patrick Vandewalle. Depth estimation from monocular images and sparse radar using deep ordinal regression network. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3343–3347. IEEE, 2021.

Yunfei Long, Daniel Morris, Xiaoming Liu, Marcos Castro, Punarjay Chakravarty, and Praveen Narayanan. Radar-camera pixel depth association for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12507–12516, 2021.

Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian FP Kooij, and Dariu M Gavrila. Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022.

Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9275–9285, 2023.

Huawei Sun, Hao Feng, Julius Ott, Lorenzo Servadei, and Robert Wille. Cafnet: A confidence-driven framework for radar camera depth estimation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2734–2740. IEEE, 2024.

Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019.

Yiran Wang, Jiaqi Li, Chaoyi Hong, Ruibo Li, Liusheng Sun, Xiao Song, Zhe Wang, Zhiguo Cao, and Guosheng Lin. Tacodepth: Towards efficient radar-camera depth estimation with one-stage fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10523–10533, 2025.

Xinyu Xie, Yawen Cui, Tao Tan, Xubin Zheng, and Zitong Yu. Fusionmamba: Dynamic feature enhancement for multimodal image fusion with mamba. *Visual Intelligence*, 2(1):37, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A Notation

Table 7 collects the symbols used in the paper.

Table 7: Symbols used in the paper.

Symbol	Meaning
$\mathbf{x}_{\text{img},t},\ \mathbf{x}_{\text{rad},t} \in \mathbb{R}^D$	Image and radar tokens at position t of a scanned sequence
$L,\ D,\ N$	Sequence length, number of channels, state dimension
$x_t,\ y_t$	Scalar input and output of one channel of the scan
$\mathbf{h}_t \in \mathbb{R}^N$	Hidden state of one channel
$\mathbf{A} \in \mathbb{R}^{D \times N}$	State matrix, trained and shared by all tokens
$\bar{\mathbf{A}}_t,\ \bar{\mathbf{B}}_t$	State and input matrices after ZOH discretization
$\boldsymbol{\Delta}_t \in \mathbb{R}^D$	Step size, which sets the memory horizon
$\mathbf{B}_t \in \mathbb{R}^N$	Input projection, computed from the image only
$\mathbf{C}_t \in \mathbb{R}^N$	Readout of the hidden state
$\mathbf{W}_\Delta^{\text{img}},\ \mathbf{W}_B,\ \mathbf{W}_C^{\text{img}}$	Image projections that produce $\boldsymbol{\Delta}_t, \mathbf{B}_t, \mathbf{C}_t$
$\mathbf{W}_\Delta^{\text{rad}},\ \mathbf{W}_C^{\text{rad}}$	Radar projections, initialised at zero
$\alpha_t,\ \mathbf{W}_g,\ b_g$	Radar gate on the step size, its weight and bias ($b_g = -2$)
$\sigma(\cdot)$	Logistic sigmoid
$c_i,\ r_i$	Image and radar features at decoder level i
$H,\ W,\ d$	Image height and width, range of a radar return
$d_{\min},\ d_{\max}$	Depth range of the model
w	Window size of the windowed scan at the middle level
$\mathbf{D}_{\text{pred}},\ \mathbf{D}_{\text{main}},\ \mathbf{D}_{\text{sparse}}$	Prediction, main target, clean sparse target
$\mathbf{D}_{\text{acc}},\ \mathbf{D}_{\text{gt}}$	LiDAR accumulated over 160 sweeps, LiDAR from a single sweep
$\tilde{\mathbf{D}}$	Depth normalised in log space to $[-1, 1]$
$\lambda_{\log},\ \lambda_{\text{lin}},\ \lambda_{\text{grad}},\ \lambda_{\text{sparse}}$	Loss weights

B Additional results on nuScenes

B.1 Day and night against published methods

Table 8 follows the day and night protocol of TacoDepth Wang et al. [2025]. A scene counts as night when its description contains "night", which splits the validation set into $5{,}282$ day and 587 night keyframes. Numbers for Singh et al. and TacoDepth come from TacoDepth. SemoDepth is the main configuration of Table 4, evaluated against sparse LiDAR from a single sweep with predictions clamped to $[0.5, 80]$ m.

Table 8: Day and night results on nuScenes Caesar et al. [2020]. Following TacoDepth Wang et al. [2025], scenes whose description contains "night" form the night split. Best per column within each split in bold.

Scene	Method	0–50 m		0–70 m		0–80 m	
		MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓
Day	Singh et al. Singh et al. [2023] (CVPR'23)	1618.9	3613.0	1924.7	4359.2	2017.9	4632.5
	TacoDepth Wang et al. [2025] (CVPR'25)	1389.5	3227.3	1680.9	3897.1	1782.4	4092.3
	SemoDepth (Ours)	**857.9**	**2606.3**	**1105.2**	**3416.5**	**1190.2**	**3731.1**
Night	Singh et al. Singh et al. [2023] (CVPR'23)	2340.8	4683.8	2863.9	5935.4	3012.9	6338.3
	TacoDepth Wang et al. [2025] (CVPR'25)	**1673.6**	**3631.4**	**1944.8**	**4425.3**	2207.6	**4574.8**
	SemoDepth (Ours)	1709.4	4381.5	2061.9	5358.2	**2146.1**	5621.7
All	Singh et al. Singh et al. [2023] (CVPR'23)	1727.7	3746.8	2073.2	4590.7	2179.3	4898.7
	TacoDepth Wang et al. [2025] (CVPR'25)	1423.6	3275.8	1712.6	3960.5	1833.4	4150.2
	SemoDepth (Ours)	**943.0**	**2783.9**	**1200.9**	**3610.7**	**1285.8**	**3920.2**

By day (90% of frames) SemoDepth wins every column. Its MAE is 33 to 38% below TacoDepth and 41 to 47% below Singh et al. Within 80 m it lowers MAE by 33% (1190 against 1782 mm) and RMSE by 9% (3731 against 4092 mm) relative to TacoDepth. The day split dominates the aggregate numbers of Table 1.

At night (10% of frames) SemoDepth leads Singh et al. in all six columns. Against TacoDepth it has the better MAE@80 (2146 against 2208 mm) but trails at shorter ranges and on RMSE. Another training seed, evaluated in Table 6, reaches 2252 mm MAE@80 at night, which trails TacoDepth. The ranking at night depends on the seed, and we do not claim a lead there. At night the camera

signal is weakest, and large errors that dominate RMSE become more common. A strong attention based fusion competes closely in this regime.

B.2 Complete ablation results

Table 9 extends Table 4 with inverse depth metrics and with the three seed results for the two configurations that carry the claim about fusion before the scan. MVSP lowers iMAE@80 from 5.14 to 4.22 (18%) relative to Uniform FiLM, which matches the MAE@80 result. Radar only in $\mathbf{B}$ gives the lowest iMAE@80 and iRMSE@80. Inverse metrics weight near pixels most, and this variant is also strong within 50 m. It does not help at 70 and 80 m, where it falls behind the joint model and is close to the variant without radar in the scan. Capacity does not predict accuracy here. The variant with $\mathbf{\Delta}$, $\mathbf{B}$, and $\mathbf{C}$ (38.2 M parameters) and full RMS to level 2 (42.7 M) both trail the main model (38.1 M). We did not modulate $\mathbf{A}$ directly. It is a trained constant, and modulating $\mathbf{\Delta}$ already changes the discretized transition $\bar{\mathbf{A}}_t = \exp(\mathbf{\Delta}_t \mathbf{A})$ at every token.

Table 9: Complete ablation on nuScenes Caesar et al. [2020], including iMAE and iRMSE within 80 m. Single seed rows match Table 4. The last two rows give the mean and standard deviation over three seeds. Best per column among single seed rows in bold.

	0–50 m		0–70 m		0–80 m			
Variant	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	iMAE (km^{-1}) ↓	iRMSE (km^{-1}) ↓
Uniform FiLM	1094	3075	1437	4166	1564	4668	5.14	13.44
Full RMS to level 2	**919**	**2728**	1206	3649	1304	4011	4.13	10.74
No radar in the scan	950	2787	1217	3674	1306	4004	4.25	11.01
$\mathbf{B}$ only	933	2782	1217	3677	1315	4025	**3.98**	**10.37**
$\mathbf{\Delta}$ only (horizon)	964	2755	1251	3614	1352	3953	4.41	10.74
$\mathbf{C}$ only (readout)	956	2792	1227	3672	1317	4012	4.36	10.89
$\mathbf{\Delta}$+$\mathbf{B}$+$\mathbf{C}$	967	2802	1267	3748	1360	4092	4.22	10.74
$\mathbf{\Delta}$+$\mathbf{C}$ with blend before the scan	940	2749	1211	3657	1304	4019	4.29	11.03
$\mathbf{\Delta}$+$\mathbf{C}$ (joint, main)	943	2784	**1201**	**3611**	**1286**	**3920**	4.22	10.85
Blend before the scan, 3 seeds	$940.3_{\pm 10.3}$	$2775.3_{\pm 23.6}$	$1214.1_{\pm 10.9}$	$3675.3_{\pm 19.1}$	$1305.3_{\pm 11.5}$	$4017.3_{\pm 6.3}$	$4.29_{\pm 0.05}$	$10.96_{\pm 0.11}$
Joint (main), 3 seeds	$940.1_{\pm 4.1}$	$2785.7_{\pm 2.5}$	$1199.8_{\pm 1.5}$	$3622.4_{\pm 16.5}$	$1285.2_{\pm 0.9}$	$3935.4_{\pm 21.5}$	–	–

B.3 Effect of the supervision configuration

The main SemoDepth result is trained on the filtered target accumulated over 160 sweeps with $\lambda_{\text{grad}} = 0.5$. Table 10 compares this configuration with the raw target and $\lambda_{\text{grad}} = 0$ for the three RMS variants and for FusionMamba†. The two configurations differ in both the target and the gradient term, so the table measures their combined effect. Gradient matching is used only with the cleaned target, because on the raw target it would reinforce the edges of accumulation trails. Each row uses the checkpoint selected by the rule of §4.2.

Table 10: Effect of the supervision configuration on nuScenes Caesar et al. [2020]. Each method appears twice. Raw uses the standard target accumulated over 160 sweeps with $\lambda_{\text{grad}} = 0$. Cleaned uses the filtered target of §4.2 with $\lambda_{\text{grad}} = 0.5$. Bold marks the better regime within each pair. The cleaned joint row is the main configuration.

		0–50 m		0–70 m		0–80 m			
Method	Target	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	MAE (mm) ↓	RMSE (mm) ↓	iMAE (km^{-1}) ↓	iRMSE (km^{-1}) ↓
$\mathbf{\Delta}$ only (horizon)	Raw	**911**	2895	**1204**	3908	**1318**	4393	**4.38**	13.43
	Cleaned	964	**2755**	1251	**3614**	1352	**3953**	4.41	**10.74**
$\mathbf{C}$ only (readout)	Raw	**913**	2914	**1203**	3970	1318	4471	4.54	14.87
	Cleaned	956	**2792**	1227	**3672**	**1317**	**4012**	**4.36**	**10.89**
$\mathbf{\Delta}$+$\mathbf{C}$ (joint, main)	Raw	**900.7**	2849.5	1204.5	3900.3	1311.9	4362.9	**4.17**	12.14
	Cleaned	943.0	**2783.9**	**1200.9**	**3610.7**	**1285.8**	**3920.2**	4.22	**10.85**
FusionMamba†	Raw	**1116.4**	**3329.5**	**1454.3**	4482.8	**1587.8**	4992.7	5.44	15.43
	Cleaned	1179.9	3337.2	1483.3	**4282.5**	1590.5	**4640.0**	**5.34**	**12.90**

The cleaned configuration with gradient matching lowers every RMSE metric for every model relative to the raw configuration without it. RMSE@80 falls by 10.0 to 10.3% for the RMS variants and by 7.1% for FusionMamba†. iRMSE@80 falls by 10.6 to 26.8%. Among the RMS variants, the largest drops occur for the single pathway variants, whose raw iRMSE@80 was highest (14.87 and 13.43). MAE moves much less. The cleaned configuration improves the joint model by 2.0% within 80 m, leaves the readout variant and FusionMamba† unchanged within 0.2%, and costs the horizon variant

2.6%. The removed pixels are few but carry large errors under raw supervision. Squared metrics weight them heavily, while MAE is dominated by the many pixels that were already well supervised. The gap between FusionMamba† and the joint model is 17.4% MAE@80 with the raw target and 19.2% with the cleaned target, so the supervision configuration does not explain the architectural gain.

B.4 Radar corruption protocol

Corruption is applied at test time to the radar input of each nuScenes validation frame. The random generator for frame i is seeded with i, so every model sees identical corrupted inputs. Spurious returns are added at uniformly random image positions with depths drawn uniformly from $[0.5, 120]$ m. Their number is a percentage of the true returns in the frame. Removal keeps a random subset of the true returns, and the removed setting passes no radar at all. Depth noise adds zero mean Gaussian noise to the depth of each true return. We evaluate the SemoDepth and FusionMamba† models of Table 6, both selected by the rule of §4.2. Their clean results match Table 1 within seed variation. Table 11 lists every setting.

Table 11: Radar corruption at test time on nuScenes Caesar et al. [2020], within 80 m. Relative changes are with respect to the clean input of the same model.

	SemoDepth			FusionMamba†		
Setting	MAE (mm)	Change	RMSE (mm)	MAE (mm)	Change	RMSE (mm)
Clean	1284.5	–	3950.6	1590.5	–	4640.0
Spurious, +10%	1285.7	+0.1%	3952.3	1616.9	+1.7%	4678.0
Spurious, +50%	1290.0	+0.4%	3957.9	1719.4	+8.1%	4829.8
Spurious, +100%	1294.8	+0.8%	3965.2	1858.4	+16.8%	5051.9
Spurious, +200%	1302.2	+1.4%	3974.5	2125.6	+33.6%	5481.9
Keep 75%	1285.2	+0.1%	3951.9	1611.7	+1.3%	4678.1
Keep 50%	1286.8	+0.2%	3955.9	1660.1	+4.4%	4766.6
Keep 25%	1290.4	+0.5%	3965.9	1801.6	+13.3%	5032.7
Keep 10%	1296.8	+1.0%	3983.9	2239.7	+40.8%	5864.9
Radar removed	1327.4	+3.3%	4071.9	4055.9	+155.0%	8935.4
Depth noise, 0.5 m	1284.5	0.0%	3950.6	1590.5	0.0%	4640.0
Depth noise, 1 m	1284.5	0.0%	3950.6	1590.6	0.0%	4639.9
Depth noise, 2 m	1284.6	0.0%	3950.6	1590.8	0.0%	4639.7
Depth noise, 5 m	1284.6	0.0%	3950.8	1592.9	+0.2%	4639.9

Both models barely react to depth noise. They respond to whether and where returns appear, not to small changes in their depth. This matches the GSE input, which normalises each return by $d_{\max}$ and encodes it jointly with its image position. Removing radar from SemoDepth gives 1327 mm, close to a model trained without radar in the scan (1306 mm, Table 4). The image pathway of SemoDepth works on its own, as the construction intends.

B.5 Error by depth

Figure 4 and Table 12 break the errors down by ground truth depth, pooling all valid pixels of the evaluation split. On nuScenes SemoDepth has the lower MAE in every 10 m bin. On both datasets the error grows with depth, and far pixels are under predicted. Pixels beyond 50 m make up 5.1% of the nuScenes pixels and 44.9% of the squared error of SemoDepth. On ZJU-4DRadarCam, pixels beyond 70 m are 0.4% of the pixels and 7.5% of the squared error, with a bias of -7.3 m. MAE and RMSE are governed by nearly disjoint sets of pixels. This explains why SemoDepth can lead on MAE while TacoDepth keeps a small RMSE edge at 70 and 80 m on ZJU-4DRadarCam.

B.6 Failure cases

Figure 5 shows where SemoDepth fails worst. We rank all 5,869 validation frames by their MAE@80 and keep the four worst frames from distinct scenes. Frames ranked 2, 3, 4, and 6 belong to the scene of the worst frame, so the selection moves on to the next scene. No frame is chosen by hand. The

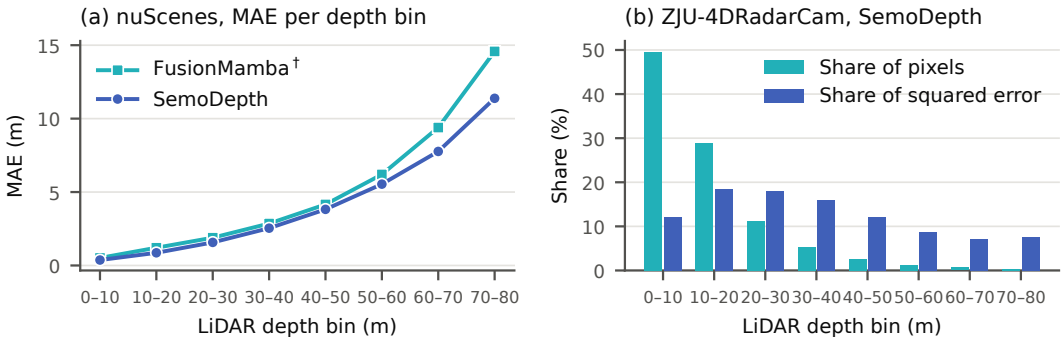


Figure 4: Error by ground truth depth. (a) MAE per 10 m bin on nuScenes for SemoDepth and FusionMamba†. (b) Share of pixels and share of squared error per bin for SemoDepth on ZJU-4DRadarCam.

Table 12: Error by ground truth depth. Pixel and SSE give the share of pixels and of the squared error in each bin. Errors and bias (prediction minus ground truth) are in metres. P50 and P99 are percentiles of the absolute error.

	nuScenes					ZJU-4DRadarCam (SemoDepth)					
Bin (m)	Pixel (%)	MAE ours	MAE FusionMamba†	Bias ours	SSE ours (%)	Pixel (%)	MAE	P50	P99	Bias	SSE (%)
0–10	46.3	0.37	0.51	0.23	10.3	49.6	0.51	0.12	5.9	0.19	12.1
10–20	25.1	0.86	1.21	0.28	13.3	29.0	1.08	0.29	10.9	−0.06	18.4
20–30	11.9	1.57	1.89	−0.19	8.7	11.2	2.00	0.58	17.3	−0.69	18.1
30–40	7.2	2.54	2.86	−0.94	10.3	5.4	2.69	0.84	24.7	−1.50	16.0
40–50	4.4	3.82	4.16	−2.15	12.4	2.6	3.45	1.18	32.2	−2.33	12.1
50–60	2.8	5.54	6.20	−3.92	15.1	1.2	4.33	1.64	41.3	−3.31	8.6
60–70	1.6	7.77	9.39	−6.44	15.6	0.7	5.31	2.04	49.6	−4.62	7.1
70–80	0.7	11.38	14.58	−10.56	14.2	0.4	7.48	2.77	63.0	−7.34	7.5

median frame has an MAE@80 of 1078 mm, and the 95th percentile is 2738 mm. The worst 1% of frames (59 frames, MAE@80 above 4070 mm) come from only 14 scenes. Rain is overrepresented among them, with 40 of the 59 frames against 18% of all frames. Night accounts for 14 of them against 10% of all frames. In these frames 44% of the absolute error comes from pixels beyond 50 m, against 28% over all frames.

In both rain frames (rows 1 and 4) the image is grey and low in contrast. The prediction contains a band of near depth along the horizon just below the radar returns, and distant facades are placed too close. In row 1 a warehouse facade at 60 to 75 m is split into a part at 30 to 50 m and a part saturated at 80 m. In row 2 a tree line that LiDAR and radar place at 40 to 55 m is predicted at 25 to 35 m. In row 3 the depth of a nearby bus spreads into the background beside and above it. In all four frames, more than 90% of the pixels with errors above 20 m are under predictions, which matches the far range bias in Table 12.

C Additional results on ZJU-4DRadarCam

C.1 Inverse depth metrics and methods with a monocular prior

Table 13 gives the inverse depth metrics for the comparison of Table 2. SemoDepth has the best iMAE and iRMSE at every range, about 18% and 8% below TacoDepth. Table 14 adds methods that consume a monocular depth prior at test time. All methods on ZJU-4DRadarCam share one radar scan and one image per frame, the released interpolated training target, and sparse evaluation. SemoDepth beats every one of the five RadarCam-Depth configurations Li et al. [2024a] on MAE and RMSE at every range. The other four configurations reach 1205.1 to 1365.4 mm MAE@80. SemoDepth matches the best RadarCam-Depth configuration on iMAE and iRMSE. The version of TacoDepth that also uses a monocular prior is the strongest entry in the table.

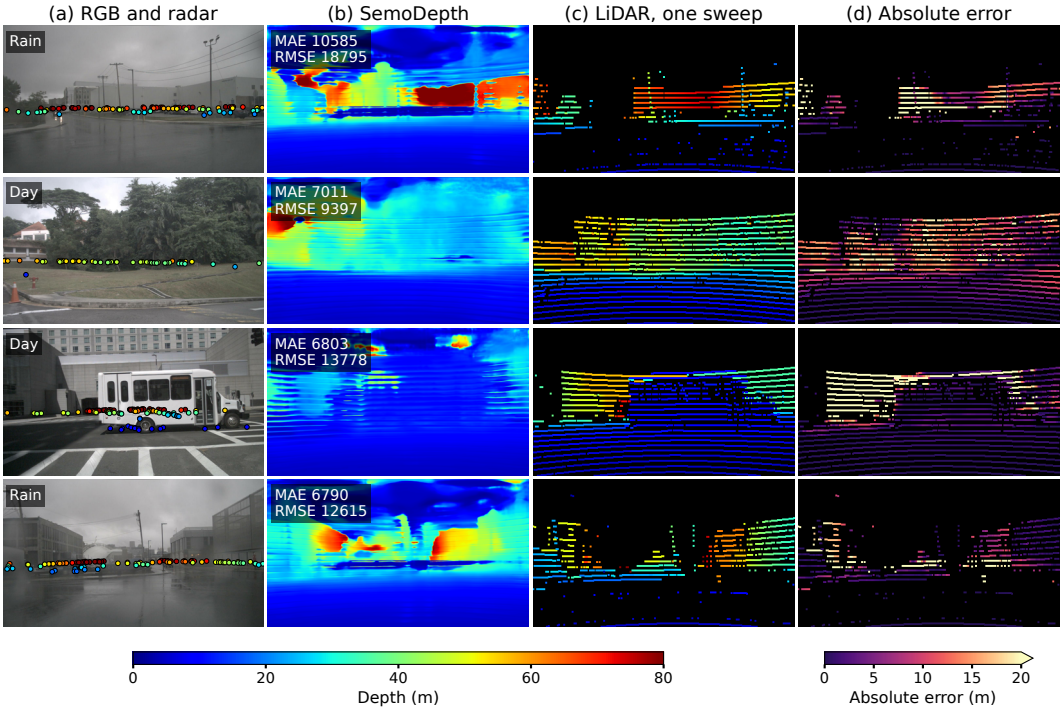


Figure 5: Worst failure cases of SemoDepth on nuScenes Caesar et al. [2020]. The rows are the four frames with the highest MAE@80 among distinct scenes (ranks 1, 5, 7, and 8 of 5,869). Column (a) shows the RGB input with radar returns coloured by depth. Column (b) shows the SemoDepth prediction with the MAE and RMSE of the frame in millimetres. Column (c) shows LiDAR ground truth from one sweep, and column (d) the absolute error at ground truth pixels, clipped at 20 m. Depth colours are clipped at 80 m.

Table 13: Inverse depth metrics on ZJU-4DRadarCam Li et al. [2024a], complementing Table 2. Baseline numbers are as reported by TacoDepth Wang et al. [2025]. Best per column in bold.

Method	0–50 m		0–70 m		0–80 m	
	iMAE (km^{-1}) ↓	iRMSE (km^{-1}) ↓	iMAE (km^{-1}) ↓	iRMSE (km^{-1}) ↓	iMAE (km^{-1}) ↓	iRMSE (km^{-1}) ↓
DORN Lo and Vandewalle [2021]	19.8	31.9	19.8	31.9	19.9	31.9
Singh et al. Singh et al. [2023]	18.1	35.3	18.0	35.2	17.9	35.1
TacoDepth Wang et al. [2025]	12.8	25.0	12.7	24.9	12.7	24.9
SemoDepth (Ours)	**10.4**	**22.8**	**10.4**	**22.8**	**10.4**	**22.8**

C.2 Qualitative results

Figure 6 shows SemoDepth on four test frames of ZJU-4DRadarCam Li et al. [2024a], together with the radar input and the LiDAR ground truth. No public ZJU weights exist for DORN Lo and Vandewalle [2021], Singh et al. Singh et al. [2023], or TacoDepth Wang et al. [2025], so the figure compares against ground truth.

D Extended limitations

The main text lists the limits that most affect our claims. This section covers them in more detail.

Supervision noise. The nuScenes target accumulates LiDAR over 160 sweeps, so moving objects leave trails and the far field contains artefacts. Our filter on the upper image removes the worst of them but not all. Table 10 shows that the cleaned configuration, which also adds gradient matching, mainly lowers RMSE and leaves MAE almost unchanged for both architectures. Some artefacts remain visible in qualitative depth maps. A principled treatment of dynamic objects in the accumulated target remains open.

Table 14: ZJU-4DRadarCam Li et al. [2024a] within 80 m, grouped by whether a method uses a monocular depth prior. Numbers for other methods come from RadarCam-Depth Li et al. [2024a] and TacoDepth Wang et al. [2025]. For RadarCam-Depth we list the best of its five configurations.

Monocular prior	Method	MAE (mm)	RMSE (mm)	iMAE (km^{-1})	iRMSE (km^{-1})
No	DORN Lo and Vandewalle [2021]	2447.6	4760.0	19.9	31.9
	Singh et al. Singh et al. [2023]	1979.5	4309.3	17.9	35.1
	TacoDepth Wang et al. [2025]	1201.1	2990.7	12.7	24.9
	SemoDepth (Ours)	1137.2	3053.0	10.4	22.8
Yes	RadarCam-Depth Li et al. [2024a]	1183.5	3229.0	10.4	22.8
	TacoDepth Wang et al. [2025]	1032.5	2850.3	9.4	20.9

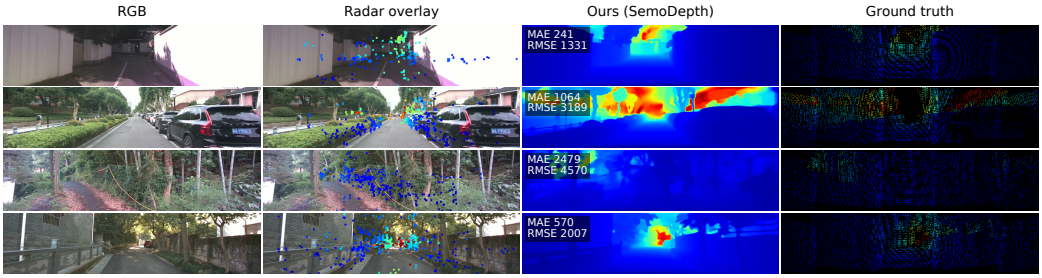


Figure 6: Qualitative results on ZJU-4DRadarCam Li et al. [2024a]. Columns show the RGB input, radar returns over the RGB, the SemoDepth prediction, and LiDAR ground truth from a single sweep. Each prediction panel gives the MAE and RMSE of the frame in millimetres within 80 m, and the colour map is clipped at 80 m. The frames were drawn at random from the test split (seed 0, indices 1105, 2093, 2607, and 3481 of 4,097), so the spread of errors reflects the test set rather than selection. SemoDepth recovers vehicle and lane geometry although the radar provides only about 380 returns per frame.

Fairness of baseline comparisons. Our protocol matches the published protocols of TacoDepth and RadarCam-Depth in split, input resolution, accumulation window (160 LiDAR sweeps at 20 Hz, about 8 s), sparse evaluation, and learning rate schedule. On ZJU-4DRadarCam every method uses the same released training target. Three differences remain. Our loss uses log space, Huber, and gradient terms instead of the plain L1 terms of TacoDepth. Our ZJU runs train for up to 100 epochs instead of 50. Our nuScenes target is filtered. The loss and the filter apply identically to every SemoDepth variant and to FusionMamba†, so they cannot explain the gap between the two architectures. Published baselines enter through their reported numbers. TacoDepth has no public release and cannot be retrained. FusionMamba† serves as the controlled baseline. Single seed ablations carry seed variance of the order of the smaller gaps between rows.

Radar corruption and missing radar. Spurious, sparse, and missing radar are measured in §4.6 and Table 11. SemoDepth degrades gracefully in each case. We did not test a systematic bias in radar depth, such as a calibration offset, or simultaneous failures of camera and radar, such as heavy spray.

Far tail on ZJU-4DRadarCam. SemoDepth under predicts far depth on both datasets (Table 12). On ZJU-4DRadarCam the bias reaches -7.3 m between 70 and 80 m, where 4D radar is sparsest, and this small tail explains the RMSE gap to TacoDepth at 70 and 80 m. Sparse far supervision and the compression of log space are likely causes.

Margin and radar density. The margin over TacoDepth is 5 to 8% MAE on ZJU-4DRadarCam and about 30% on nuScenes. ZJU uses 4D radar with elevation and about 380 returns per frame, roughly ten times denser than the 40 to 100 returns of nuScenes. Denser and less ambiguous radar leaves less work for any fusion design and less room for selection to add value. We have not tested a 3D radar dataset of comparable scale to nuScenes.

Night and weather. At night, fusion before the scan matches or beats SemoDepth on linear metrics, and the outcome depends on the seed (Table 6). Rain is covered, but fog is absent from nuScenes and untested.

Embedded deployment. All latencies are measured on a workstation GPU at FP32 and batch 1. We have not measured latency, memory, or the behaviour of the selective scan kernel on embedded automotive hardware, and quantisation is untested.

Safety of dense depth failures. Depth errors concentrate at far range, at night, and in the failure cases of Figure 5. A planner that trusts dense depth uniformly could act on these errors. Any deployment needs evaluation by scene type, uncertainty estimates, and monitoring of inputs that differ from the training data.

Scope and statistics. We evaluate front camera and automotive radar only. The argument for selection inside the scan does not depend on the sensors, but we have not tested LiDAR and camera, radar and thermal, or other pairs. Table 1 and the blend comparison use three seeds. Other ablations, the condition splits, and the corruption study use one seed, as is common in this field Li et al. [2024a], Wang et al. [2025], Sun et al. [2024], Li et al. [2024b].

E Broader impacts

Robust, low cost, and weather tolerant metric depth can enable safer driver assistance on vehicles that cannot afford automotive LiDAR. SemoDepth relies on radar already common in vehicles and degrades gracefully when radar is poor. It could lower the cost of features such as forward collision warning, automatic emergency braking, and adaptive cruise control. These features help most in cost sensitive markets and in weather where camera only systems degrade.

Better scene understanding from radar and camera is also dual use. The same depth predictions could feed surveillance or military systems. Large scale deployment shares the usual risks of automated perception. Failures can concentrate on rare scenes or atypical objects, such as pedestrians with unusual shapes or clothing. Such failures can have safety consequences if a planner assumes uniform accuracy. We do not study geographic or demographic robustness. nuScenes and ZJU-4DRadarCam come from narrow regions (Boston and Singapore, and one campus), which limits how far our numbers transfer.

The released checkpoints are research artefacts and not production components for driver assistance. They should be used only for research, and any deployment should add evaluation by scene type and monitoring for unfamiliar inputs. The method adds no capability for harm beyond existing radar camera depth methods Sun et al. [2024], Wang et al. [2025], Li et al. [2024b], so we do not restrict access.

F Compute resources

All experiments run on one workstation with one NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPU (96 GB, CUDA 13.0). Training uses bf16 mixed precision (§4.2) and needs about 24 GB at batch size 12. A full 50 epoch run on nuScenes ($27{,}430$ training frames) takes about 26 hours. A ZJU-4DRadarCam run ($26{,}055$ training frames, 100 epochs) takes about 24 hours. The filtering of the supervision target is a single pass over the training and validation data, about 40 minutes on 40 CPU threads. Inference takes 26.8 ms per frame at 900×1600.

The configurations of Tables 1, 2, 10, and 8 took about 13 GPU days. The additional ablations, the corruption study, and the analyses by condition and depth took about 3 GPU days. Including preliminary experiments that are not reported, the project used about 28 GPU days on the same workstation.

G Code and reproducibility

Code, preprocessing scripts, and trained checkpoints for SemoDepth are available at `https://github.com/HouV001/SemoDepth`. The repository contains the model (`mambadepth_fusion/`), training scripts for both datasets, an evaluation script for nuScenes, the LiDAR accumulation utility, the filtering of the supervision target, and unit tests. The tests include the check that numerically verifies Eq. (5) on a CUDA device. The README describes the environment and the data preparation. Training scripts fix the random seeds. Checkpoint selection on the validation split happens automatically during training.

H Licenses for existing assets

Table 15 lists the external datasets, code, and pretrained weights we use, with versions, licenses, and roles. All assets are used under their licenses. nuScenes Caesar et al. [2020] is used under its non commercial research license, and no asset is redistributed.

Table 15: External assets used in SemoDepth. Versions are the snapshots used for our experiments. Licenses are summarised, and the full texts ship with each asset.

Asset	Version	License	Role in this work
nuScenes Caesar et al. [2020]	v1.0-trainval	CC BY-NC-SA 4.0	Primary benchmark
ZJU-4DRadarCam Li et al. [2024a]	official release	per upstream repository	Secondary benchmark
nuscenes-devkit	1.1.x	Apache 2.0	LiDAR accumulation and projection
ResNet-34 (torchvision)	ImageNet1K-V1	BSD-3-Clause	Image encoder initialisation
`mamba-ssm`	1.2.x	Apache 2.0	Selective scan CUDA kernel
`causal-conv1d`	1.2.x	BSD-3-Clause	Mamba dependency
PyTorch	2.x	BSD-3-Clause	Training framework
PCA-GM Wang et al. [2019]	reference implementation	per upstream repository	Building block of the radar GSE
TacoDepth Wang et al. [2025]	paper formulae	no code reused	Baseline and loss reference
RadarCam-Depth Li et al. [2024a]	ZJU release	per upstream repository	Baseline and host of the ZJU dataset

Each method we compare against is cited where it is used. Our code consists of new training, evaluation, and visualisation scripts that use the listed assets through their public interfaces.

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 contributions in the abstract and introduction are RMS (§3.2), MVSP (§3.3), and the results on nuScenes. They are supported by §4.3, the ablations in §4.5, and the robustness study in §4.6.

   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: §5 discusses the night boundary, noise in the accumulated LiDAR target, the fairness of baseline comparisons, and the far tail on ZJU-4DRadarCam. Appendix D extends this to radar corruption, embedded deployment, safety, sensor scope, and seed variance.

   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: The paper presents an empirical method without formal theorems. The property at initialisation in Eq. (5) is an identity that a unit test in the released code verifies numerically.

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: §3 describes the architecture, §4.1 the datasets and protocols, and §4.2 the training recipe, including optimiser, schedule, epochs, stopping rule, batch size, depth ranges, and loss weights. Code and checkpoints are public (Appendix G).

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: Both datasets are public. Code, preprocessing scripts for LiDAR accumulation and target filtering, training scripts, and checkpoints are available at `https://github.com/HouV001/SemoDepth` (Appendix G).

   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: §4.1 and §4.2 give the data splits, image resolution, optimiser, learning rate schedule, batch size, precision, loss weights, stopping rule, evaluation ranges, and metrics.

   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: [Yes].

   Justification: Table 1 reports the mean and standard deviation over three training seeds, and the blend ablation also uses three seeds (§4.5). The variability captured is the training seed. Other ablations, condition splits, and corruption results use one seed, as is common in this field, which Appendix D states.

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: §4.4 reports latency, parameters, and FLOPs. Appendix F gives the GPU, the time per run, and the total compute of the project, including experiments that are not reported.

   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 work uses public driving datasets under their licenses and collects no data from human subjects. It complies with 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: Appendix E discusses safer and cheaper driver assistance, dual use of radar perception, failures on rare scenes, and the narrow geography of the datasets.

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 released model predicts depth from public driving benchmarks. It generates no text or images with a high risk of misuse and uses no scraped data.

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: Datasets and code dependencies are cited and used under their licenses. Appendix H lists each asset with its version, license, and role.

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 public repository documents the environment, data preparation, training, and evaluation in its README (Appendix G).

    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 work involves no crowdsourcing or research with human subjects. All data come from public driving datasets.

    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: This paper involves no research with human subjects.

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 part of the method. Any use was limited to writing and editing assistance, which the NeurIPS policy does not require us to declare.

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.