

Sparse Beats Dense

Sparse Beats Dense: Rethinking Supervision in Radar-Camera Depth Completion

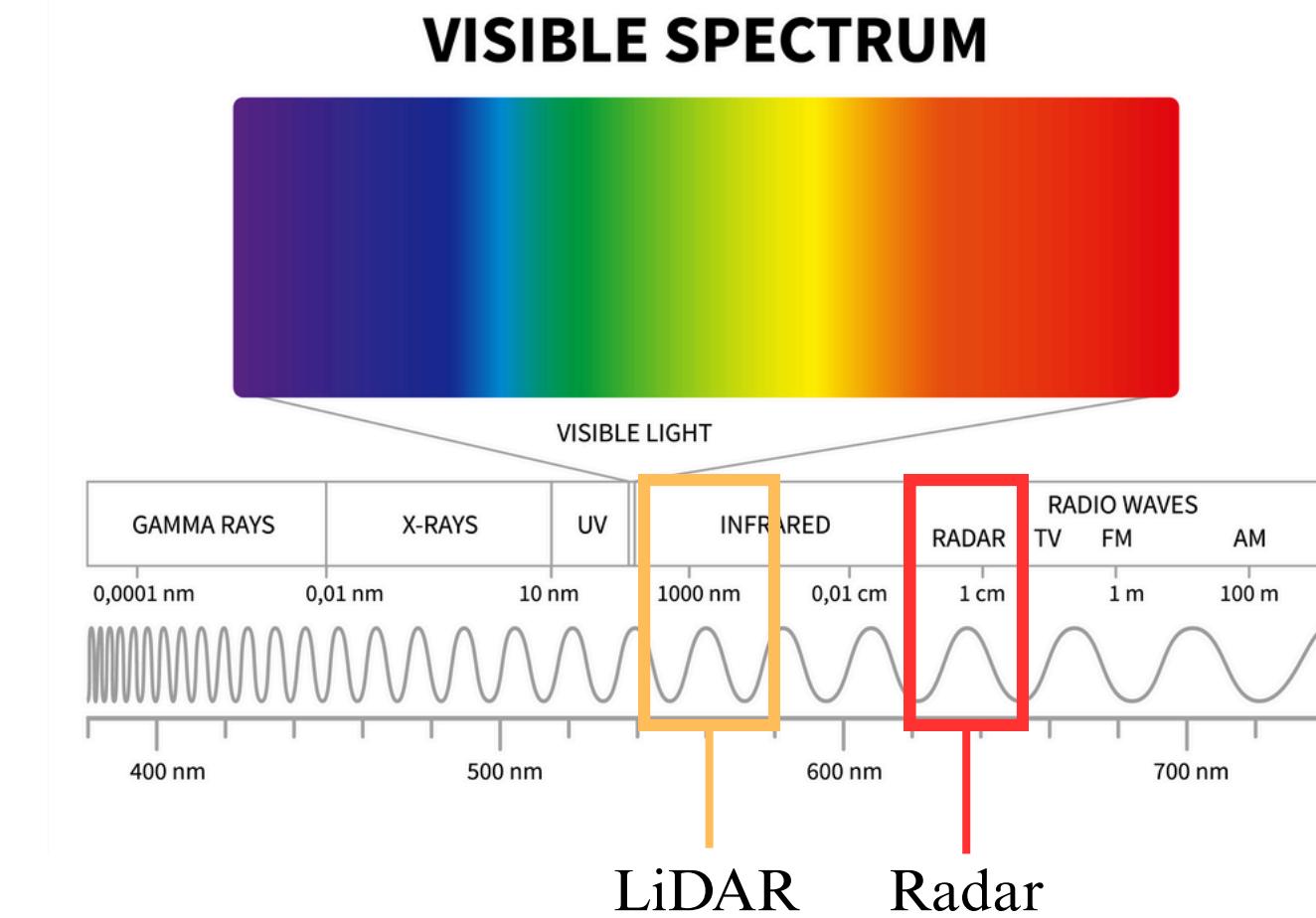
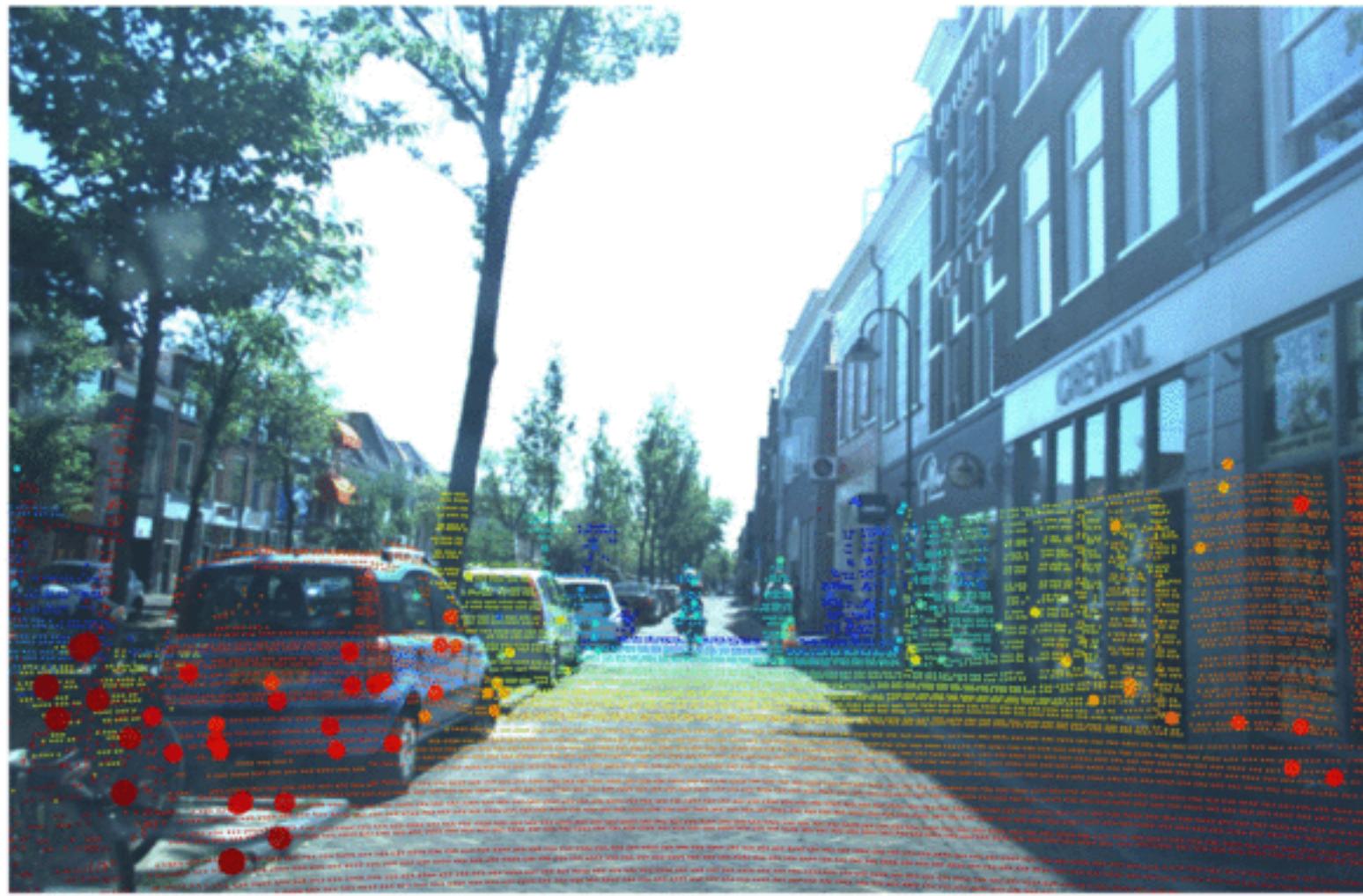
JinYong Kim

Introduction

Background

Introduction

- Radar는 Beam Width가 커서 LiDAR보다 1,000배는 더 Sparse, Noisy하다.
- Frequency Band : 30 MHz to 300 GHz
- 높은 가격합리성과 적은 전력 소비량이 장점이다.
- 3D Radar Coordinates : Radar sensor가 직접 관측한 좌표 체계, 3 channel로 구성된다. $p \in R^{nx3}$
 - Range(r) : 센서와 물체 사이의 직선 거리
 - Azimuth(θ) : 센서기준 수평 방향 각도 (좌/우)
 - Elevation(ϕ) : 센서기준 수직 방향 각도(위/아래)



Dataset

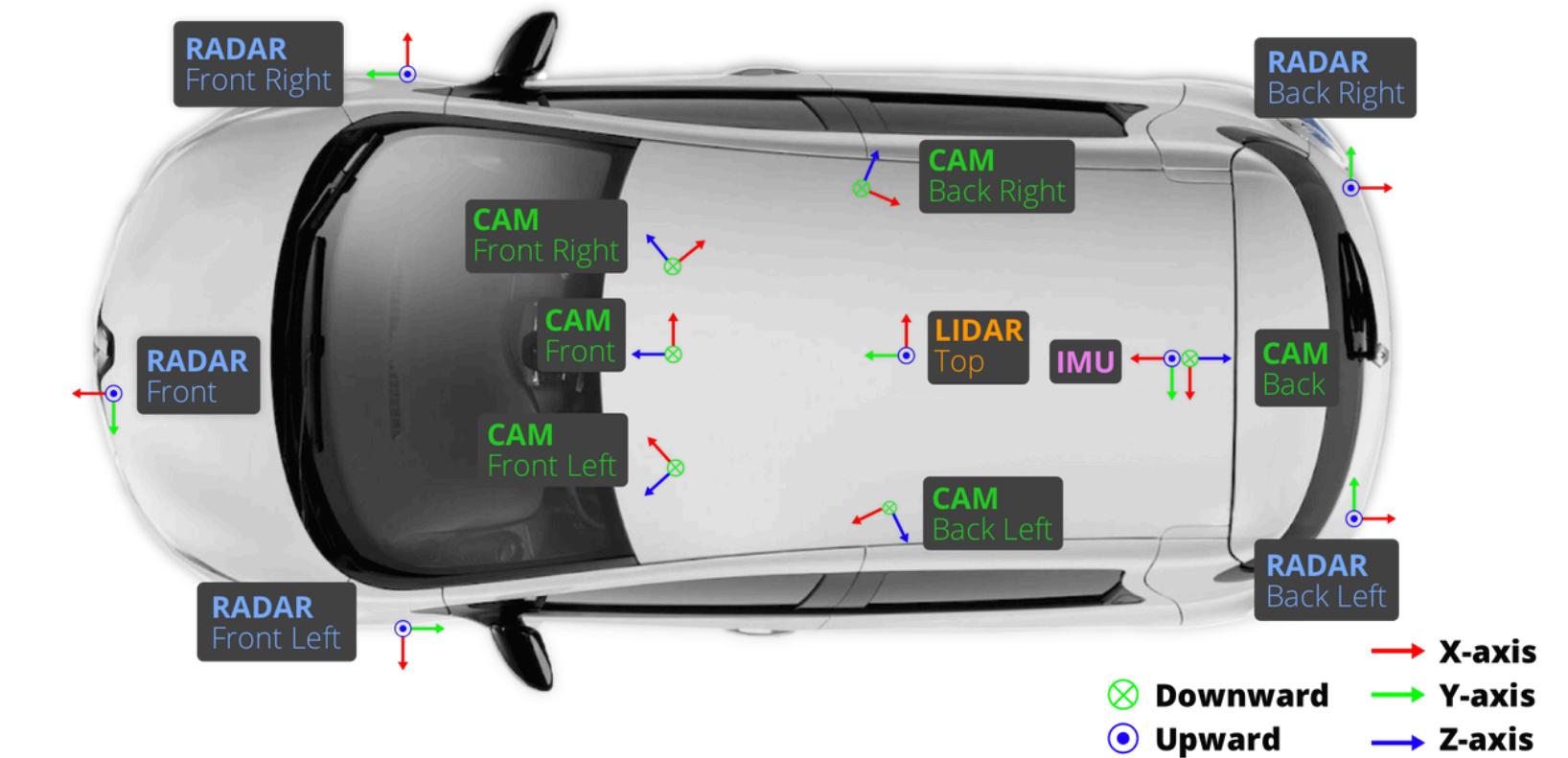
Introduction

nuScense :

- 1,000 Driving Scenes, 1.4M camera images, 1.4M RADAR sweeps
- 700 Train Scenes, 150 Validation Scenes, 150 Test Scenes



nuScense Test Dataset(Front)



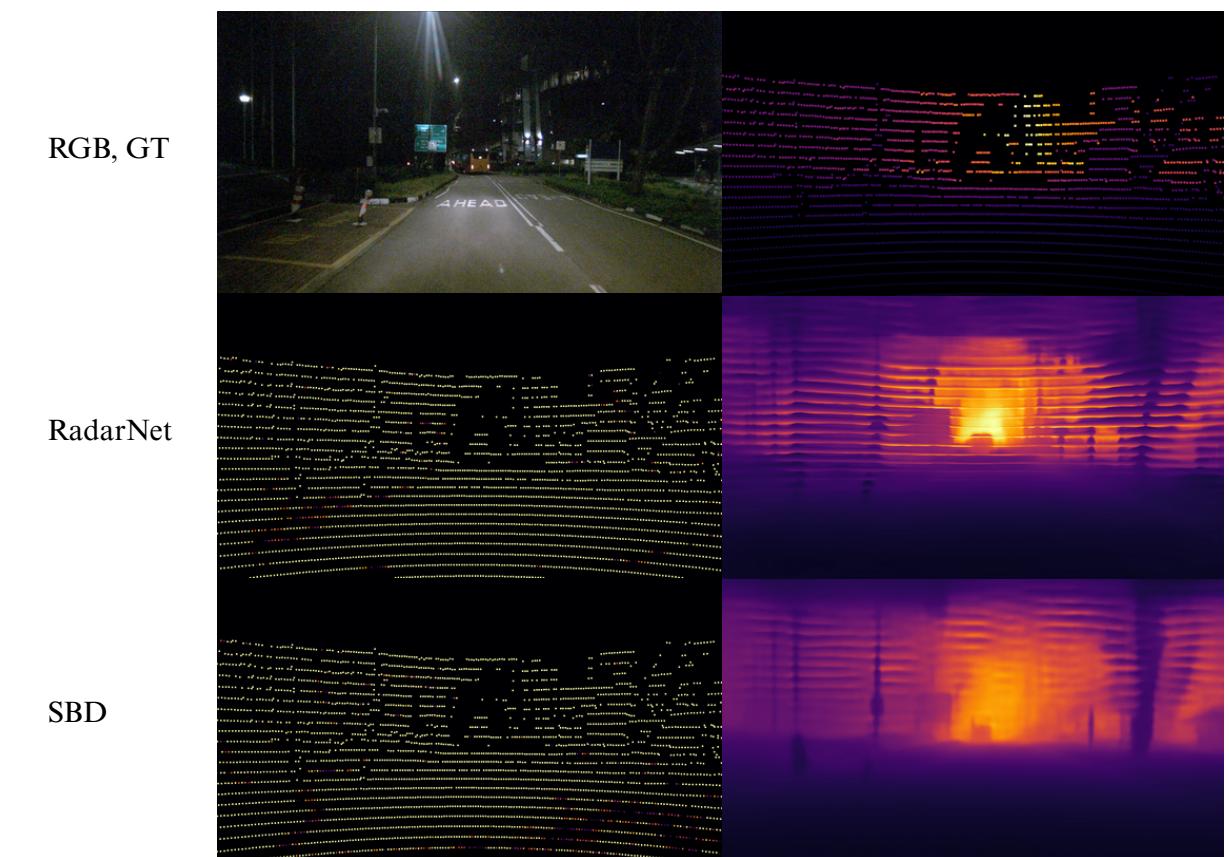
nuScense - Car Setup

Elevation Ambiguity

- Radar Sensor로 측정된 Radar Point들의 Elevation value가 ambiguity한 문제
- 차량용 Radar Sensor의 본질적인 한계

Stripe-like Pattern Artifacts

- Prediction 결과에 나타나는 줄무늬 형태의 현상

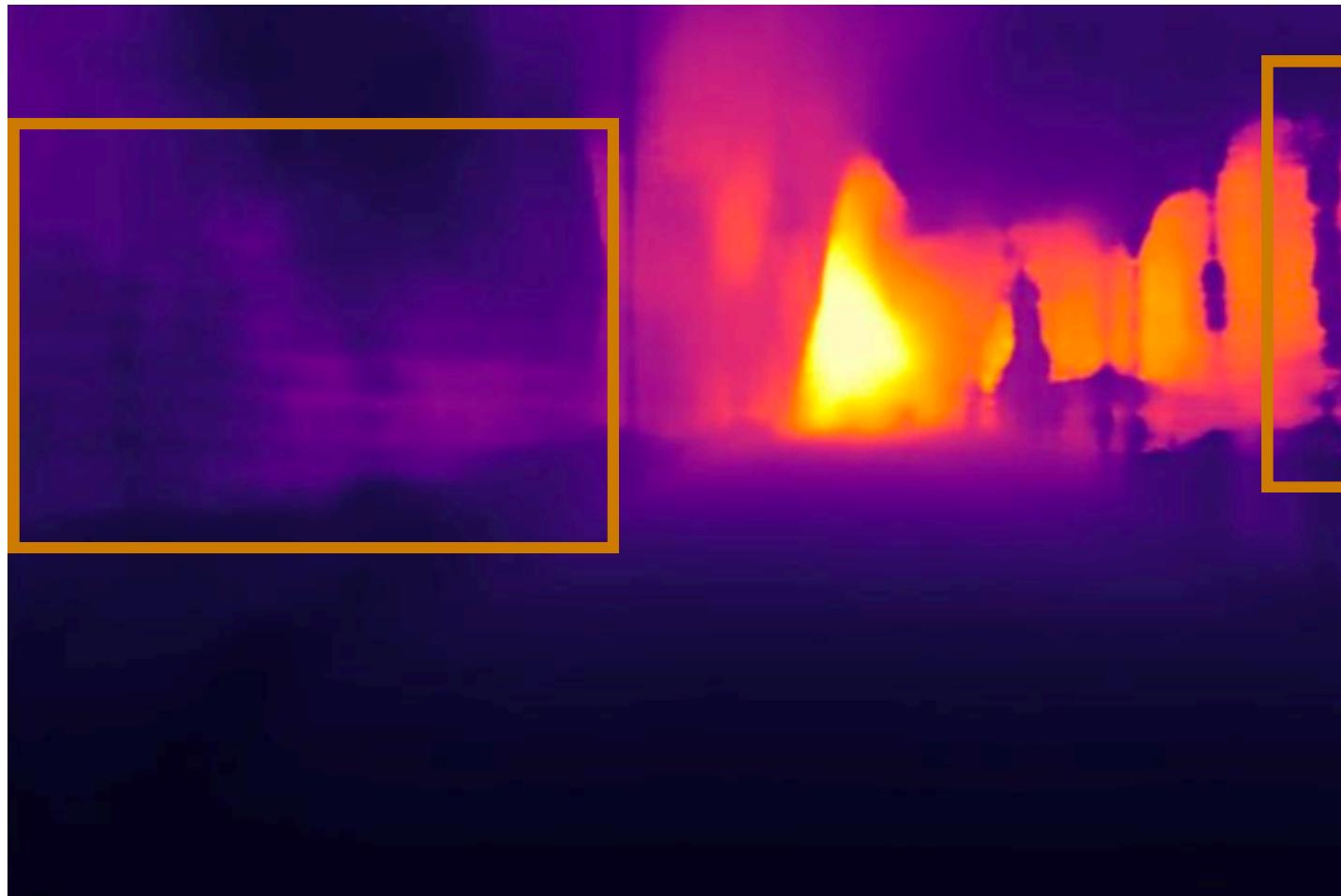


Problem Statement

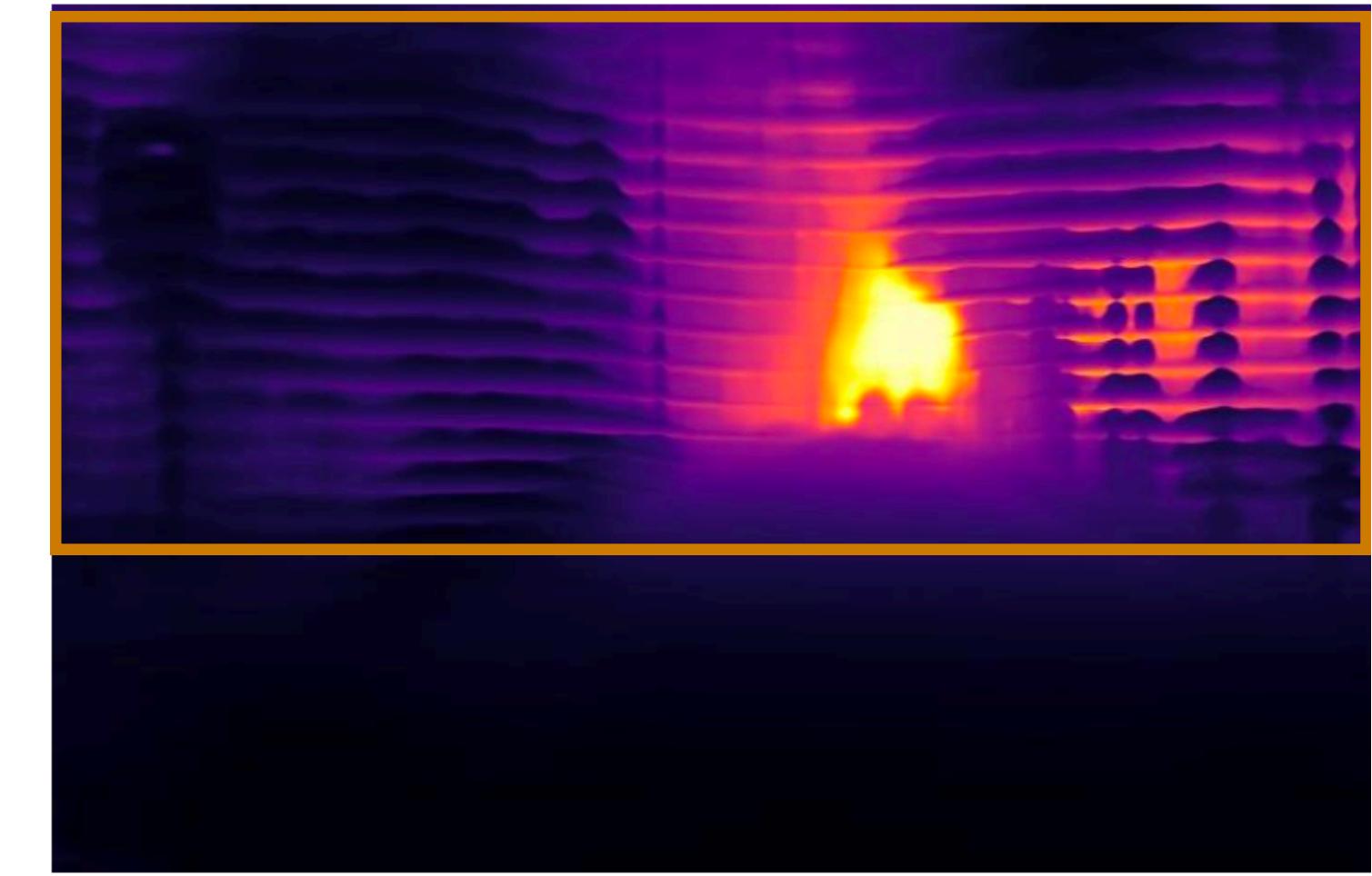
Introduction

Stripe-like Pattern Artifacts

- 이전 연구들의 결과를 보면 Stripe-like Pattern 문제들을 확인할 수 있다.
- RadarNet은 이 연구 이전의 SOTA 모델이다.



RC-PDA

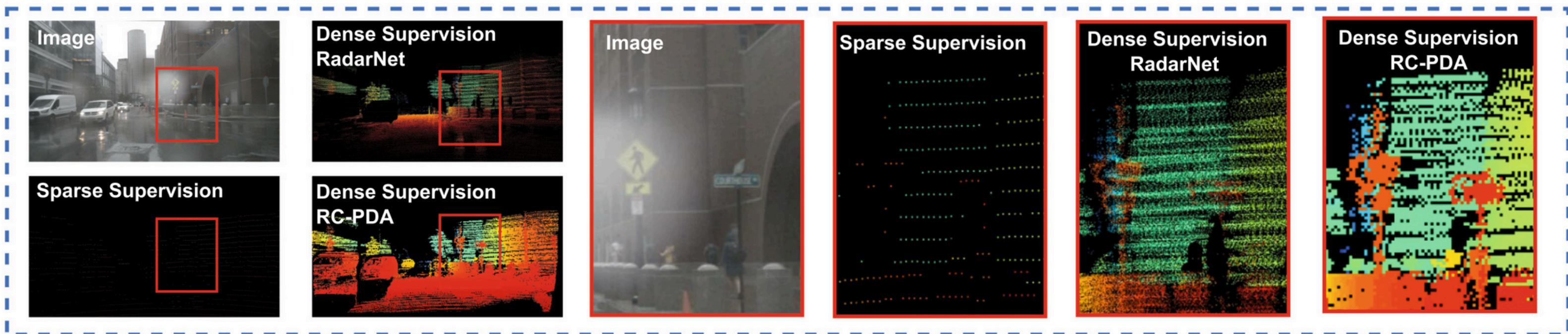


RadarNet

Previous Approach

Introduction

- 이전 연구들은 이러한 Stripe-like Pattern Artifacts를 완화하기 위해 Multi-Frame Stacking, Interpolation 방법들을 사용하였다.
 - RadarNet : 161 Frame(이전 80 + 현재 + 이후 80)의 LiDAR Depth를 누적하여 Dense GT 생성
 - RC-PDA : 24 Frame(이전 4 + 현재 + 이후 20)의 LiDAR Depth를 누적하여 Dense GT 생성
- 하지만, 이러한 Dense Supervision 방법들도 여전히 Stripe-like Pattern 문제들이 남아있다.
- 또한, Multi-Frame Stacking 방식은 건물과 표지판의 Depth Map이 혼합되는 문제 등이 발생하여, 오히려 noise를 만들게 한다.

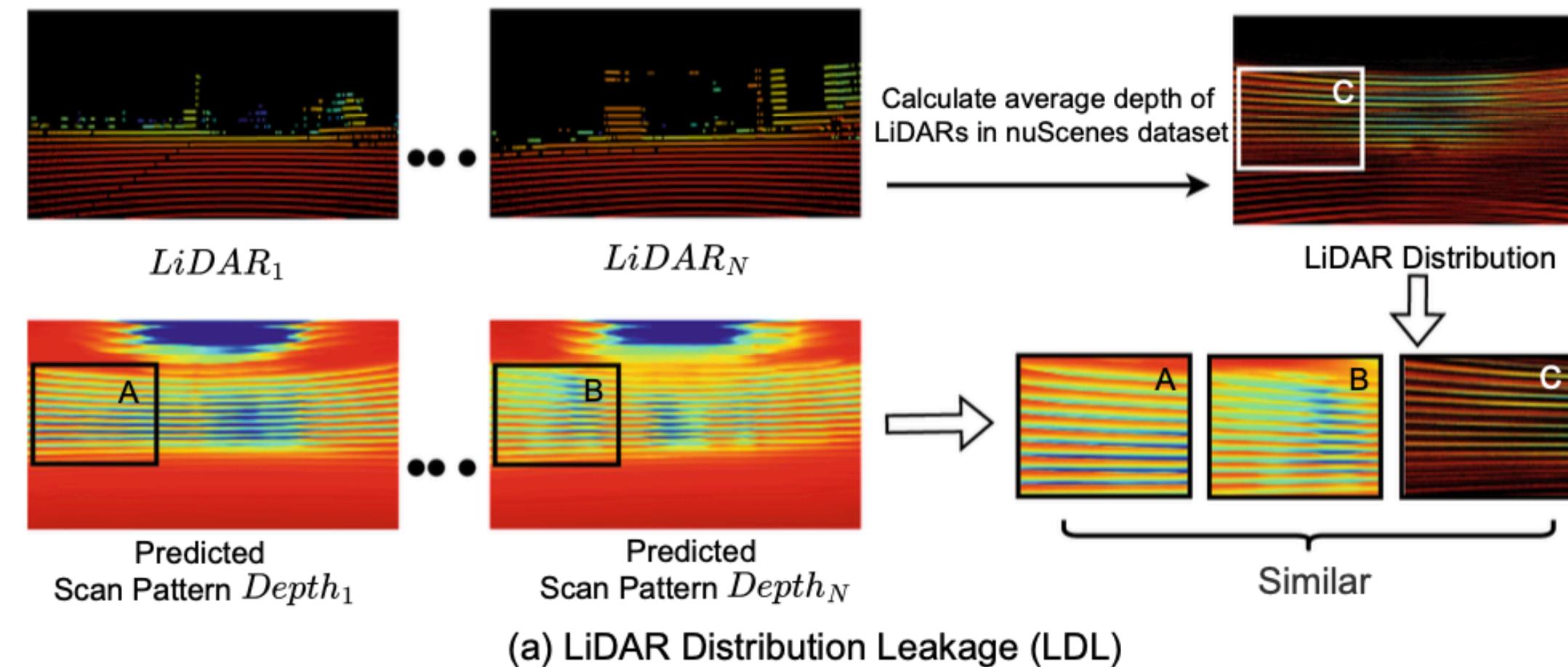


Methods

LiDAR Distribution Leakage

Methods

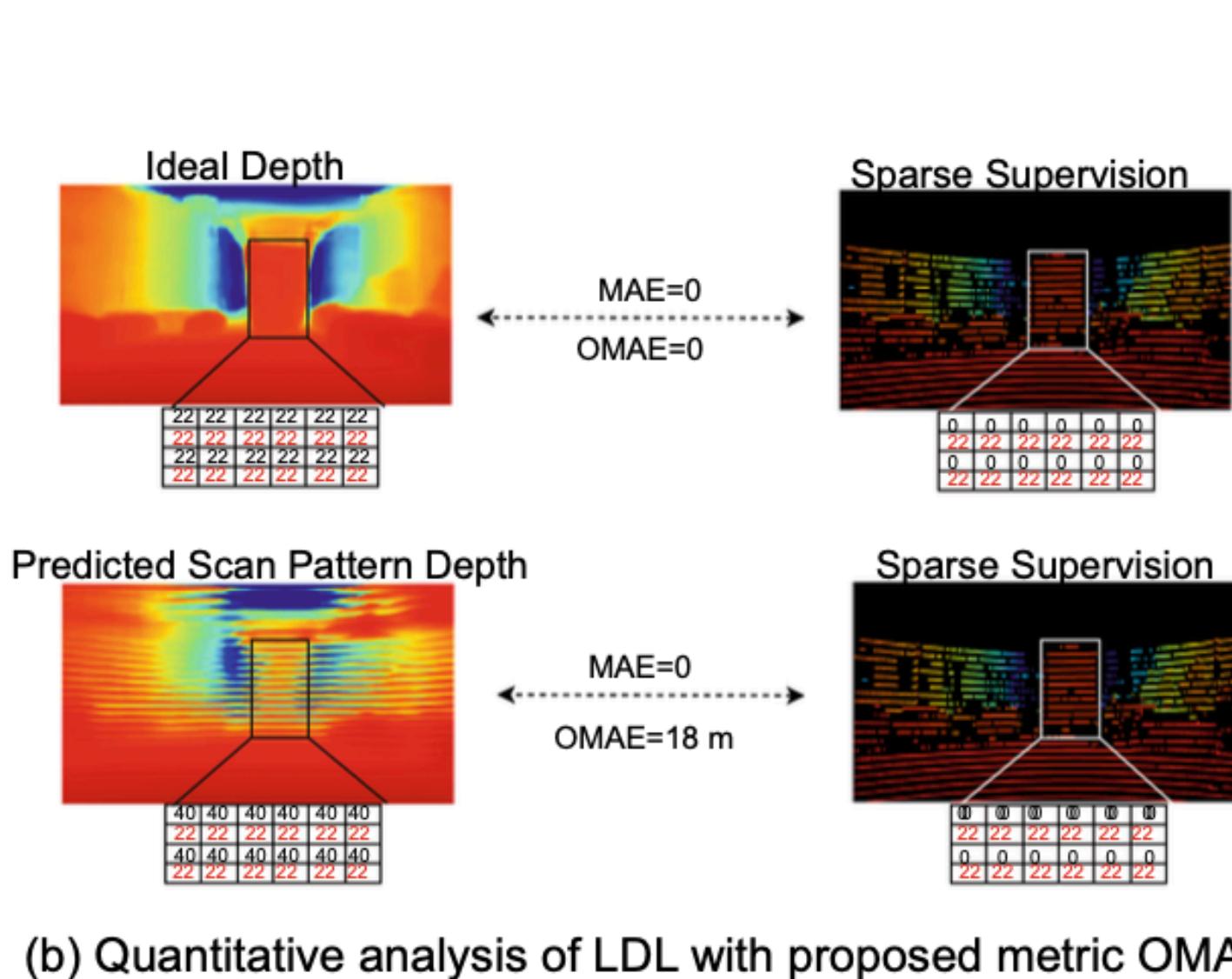
- Stripe-like Pattern Artifacts가 발생하는 이유는 2D로 투영된 LiDAR GT가 Stripe-like Positional Pattern을 갖기 때문이다.
 - **LiDAR Distribution Leakage(LDL)**
 - LiDAR Sensor의 Scanning Pattern과 LiDAR data를 2D image으로 Projection Transformation이 원인이다.
- 이러한 Sparse LiDAR data의 독특한 Distribution은 모델이 Sparse Point 외부의 Depth Representation 학습에 소홀하게 만들어 → 불완전한 Depth Completion 결과를 만든다.



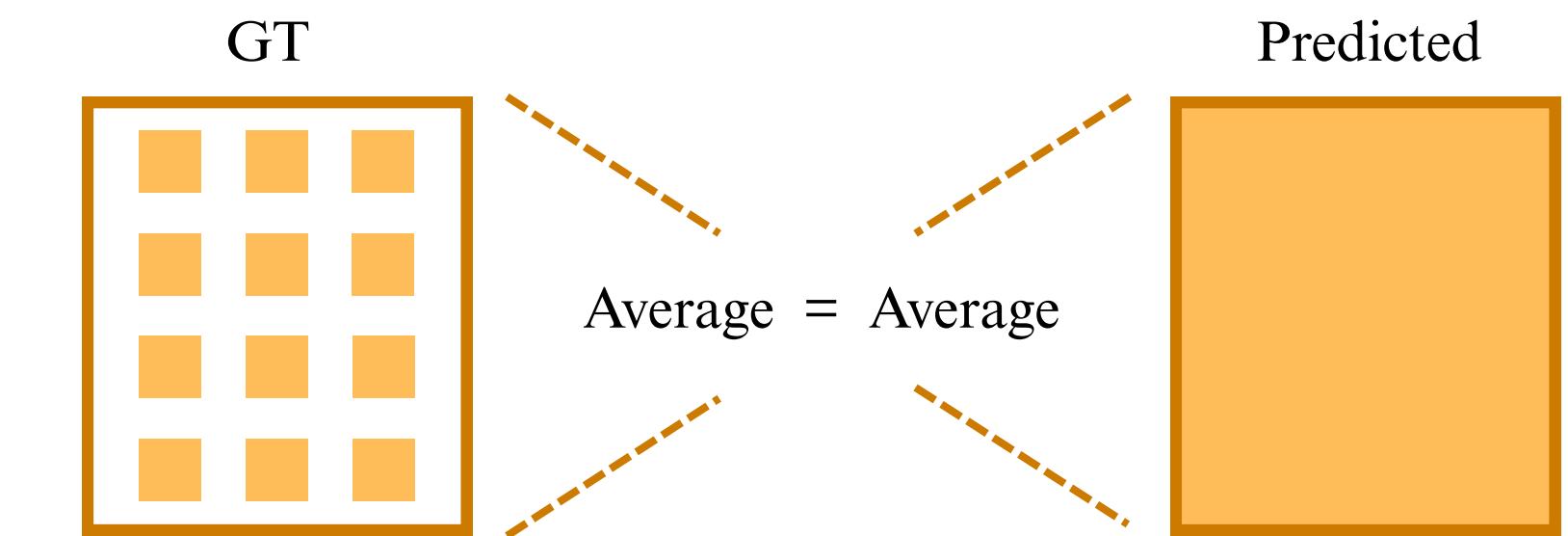
LiDAR Distribution Leakage

Methods

- LDL을 정량분석하기 위해 Object-level MAE(OMAE) metric을 제안한다.
 - GT에서 Object Region내의 Sparse Point들의 Average Depth Value와 예측된 Depth Map의 같은 Object Region내의 Average Depth Value를 사용하여 계산된다.
- Object Mask : ADE20K dataset으로 학습된 ViT-Adapter (Semantic Segmentation)



(b) Quantitative analysis of LDL with proposed metric OMAE



$$OMAE = \sum_{o \in \mathcal{O}} \frac{|\Omega_{gt}^o|}{|\Omega_{gt}|} \left| \frac{1}{|\Omega_{pred}^o|} \sum_{x \in \Omega_{pred}^o} \hat{d}(x) - \frac{1}{|\Omega_{gt}^o|} \sum_{x \in \Omega_{gt}^o} d_{gt}(x) \right|$$

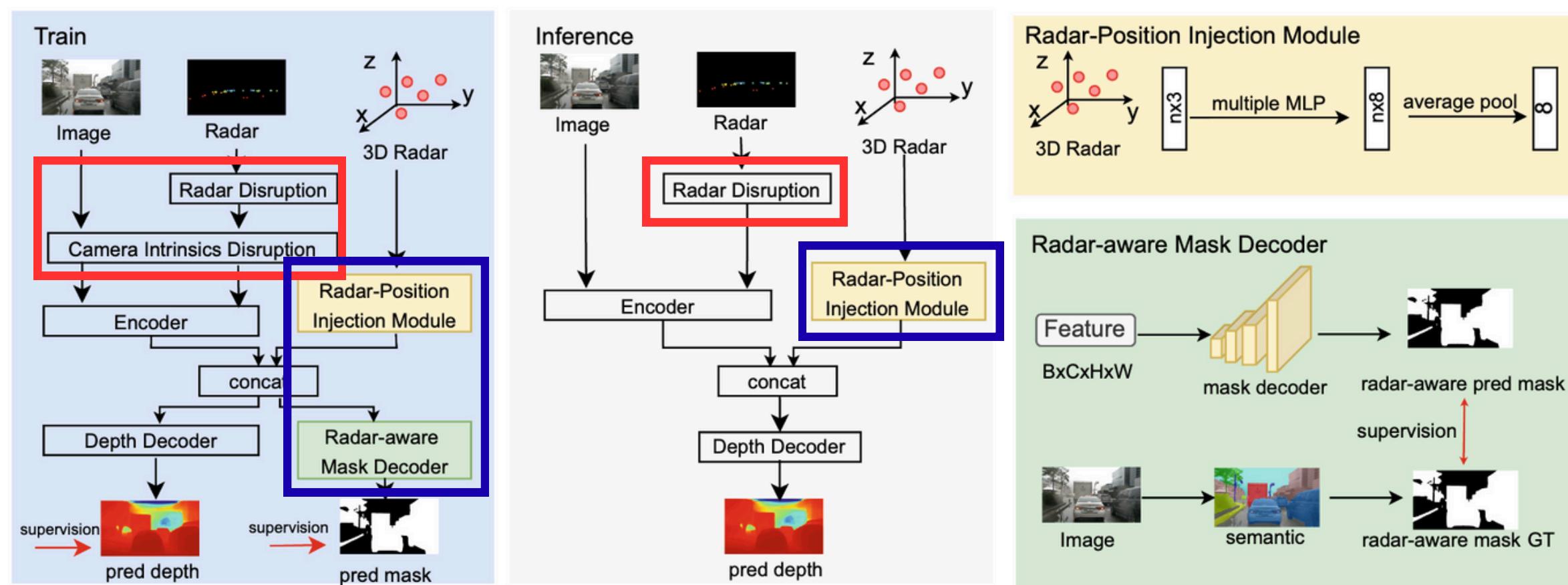
Where

- $|\Omega_{pred}^o|$: Object Region in Predicted Depth Map
- $|\Omega_{gt}^o|$: Object Sparse Point in GT
- $|\Omega_{gt}|$: All Points in GT
- $\hat{d}(x)$ $d_{gt}(x)$: Depth Value (Predicted / GT)

Architecture Overview

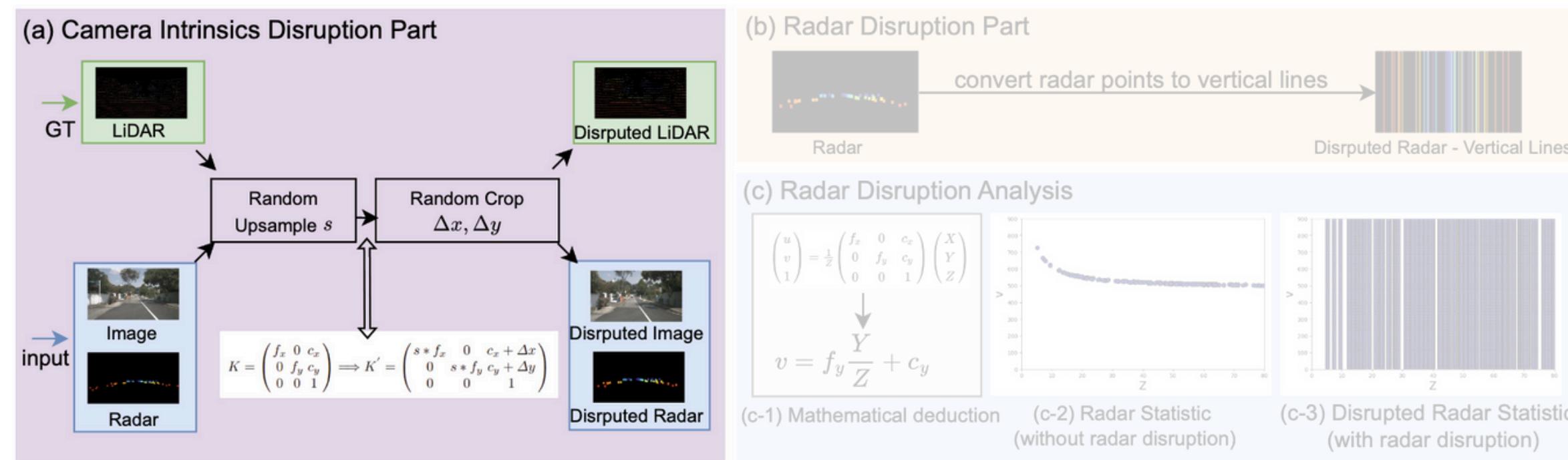
Methods

- Disruption Part
 - Camera Intrinsic Disruption
 - Radar Disruption
- Compensation Part
 - Radar-aware Mask Decoder
 - Radar-position Injection Module



Camera Intrinsics Disruption

- Stripe-like Pattern을 만드는 주요 요인 3가지(LiDAR Sensor 특성, Extrinsic Parameter, Intrinsic Parameter) 중에 구현에 용이한 Camera Intrinsic Parameter를 Disruption 대상으로 선택했다.
- 입력 Image, Radar와 GT LiDAR에 동일하게 Random Upsample과 Random Crop을 적용한다.
- 이 두 연산은 수학적으로 Camera Intrinsic Parameter K 를 K' 로 바꾸는 효과가 있다.
 - 즉, 각 샘플마다 카메라의 Focal Length, Principle Point가 달라지는 것과 같은 효과를 준다.

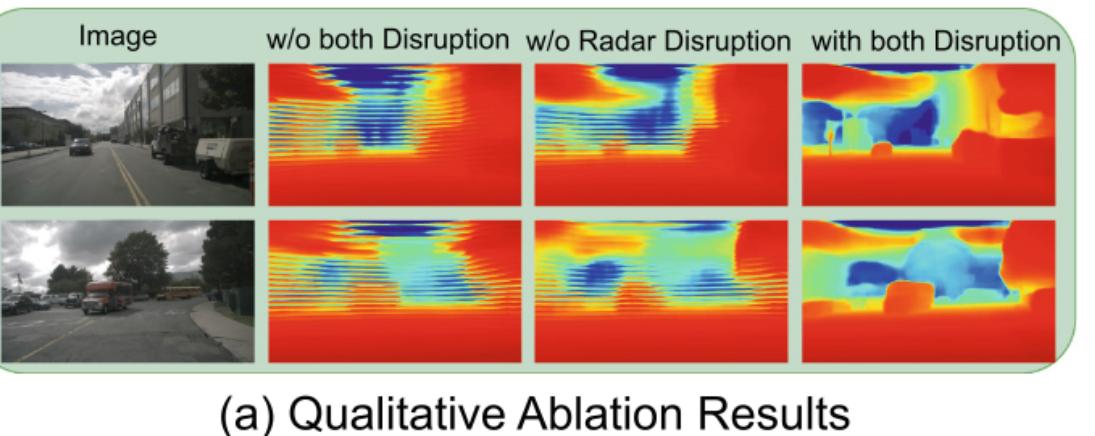


Disruption

Methods

Radar Disruption

- Camera Intrinsics Disruption 만으로는 Stripe-like artifacts를 완전히 없앨 수 없었다.
 - 모델이 여전히 2D Projected Radar Image를 통해 카메라 Intrinsic Parameter(f_y, c_y)를 쉽게 학습할 수 있었기 때문.
 - 특히 Y 값은 레이더 센서의 특성상 일정하다고 가정
 - v와 Z 값만으로 f_y, c_y 를 쉽게 추론 가능
- 입력 Radar Point를 vertical line으로 확장하여 Radar Disruption을 적용하였다.
 - → v와 Z 사이의 관계가 복잡해져 모델이 쉽게 Intrinsic Parameter를 학습할 수 없음
- Y, Z 값에 노이즈를 추가하는 방법도 실험하였지만, vertical line으로 확장하는 방법이 가장 효과적이었다.



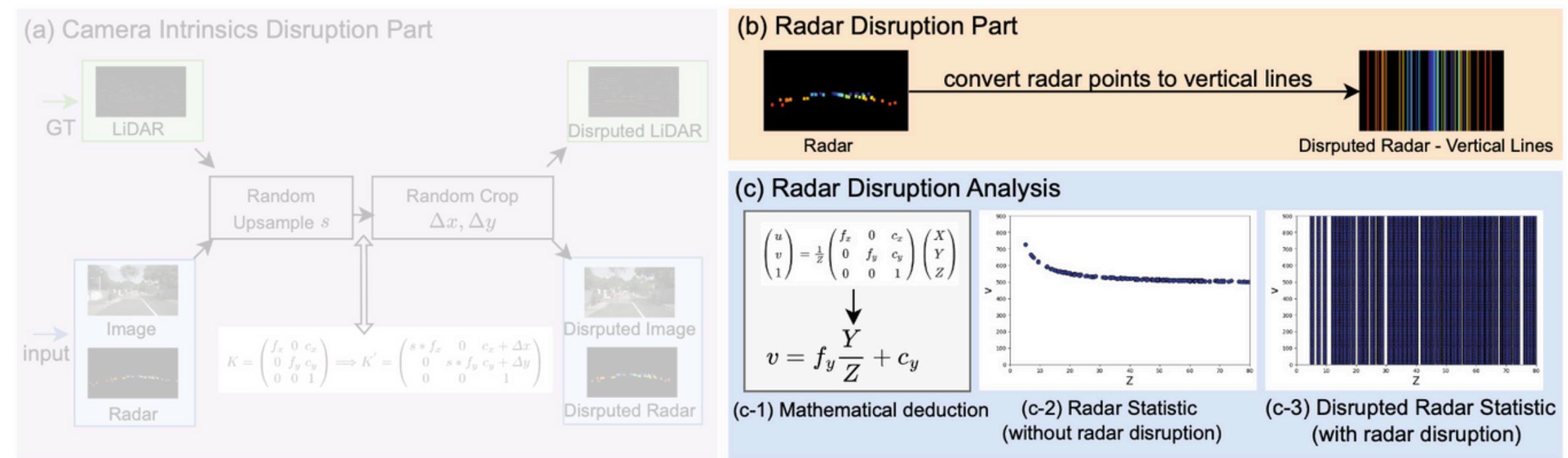
Camera Intrinsics Disruption	Radar Disruption	OMAE 0-80m (mm) ↓
✗	✗	2927.0
✓	✗	2745.8
✓	✓	2104.3

Table (b-1) Disruption ablation experimental results

Radar Disruption	OMAE 0-80m (mm) ↓
Z + Random Noise	2614.0
Y + Random Noise	2345.8
Y + Extend Height	2104.3

Table (b-2) Comparison of radar disruptions

(b) Quantitative Ablation Results



Compensation

Methods

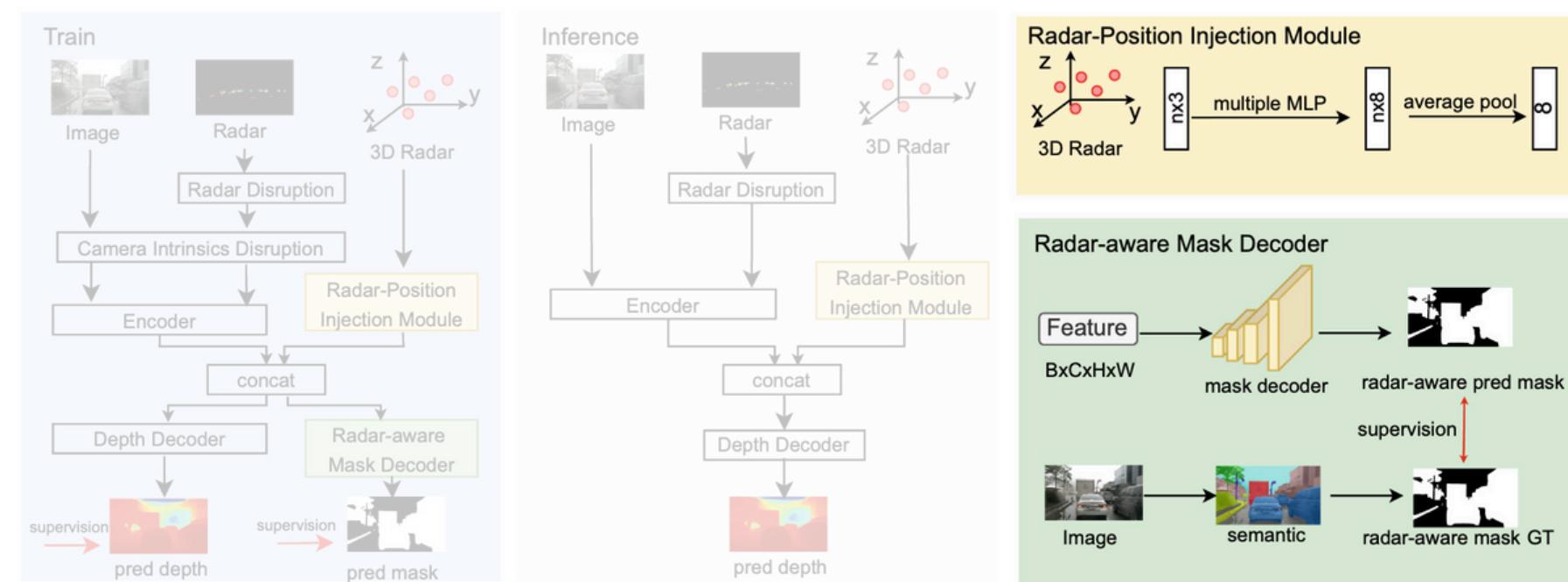
- Disruption 단계에서 LiDAR Distribution을 흔들었지만, 이 과정에서 손실되는 정보들을 보안할 필요가 있다.

Radar-Aware Mask Decoder

- Radar Point는 특정 물질들에서 더 잘 반사되는 특징이 있다. (Metals, Aluminium, certain Plastics et al.)
 - 주행환경에서 보면 Building, Wall, Tree, Car, Person et al. → 12 Categories of Object
 - 즉, Radar Point가 분포할 가능성이 높은 영역은 어느 정도 예측 가능
- Semantic Segmentation Mask → Binary Mask
- Radar-Aware Mask Decoder는 U-net Decoder 구조를 사용하여 Binary Mask와 비교하여 학습

Radar-Position Injection Module

- MLP를 사용하여 3D Radar Point의 정보들을 추출하여 사용한다.
- $F_{pj} = \Psi(\mathcal{P})$ where $F_{pj} \in \mathbb{R}^{nxC}$: Injected Positional Feature
 $\Psi(\mathcal{P})$: MLP

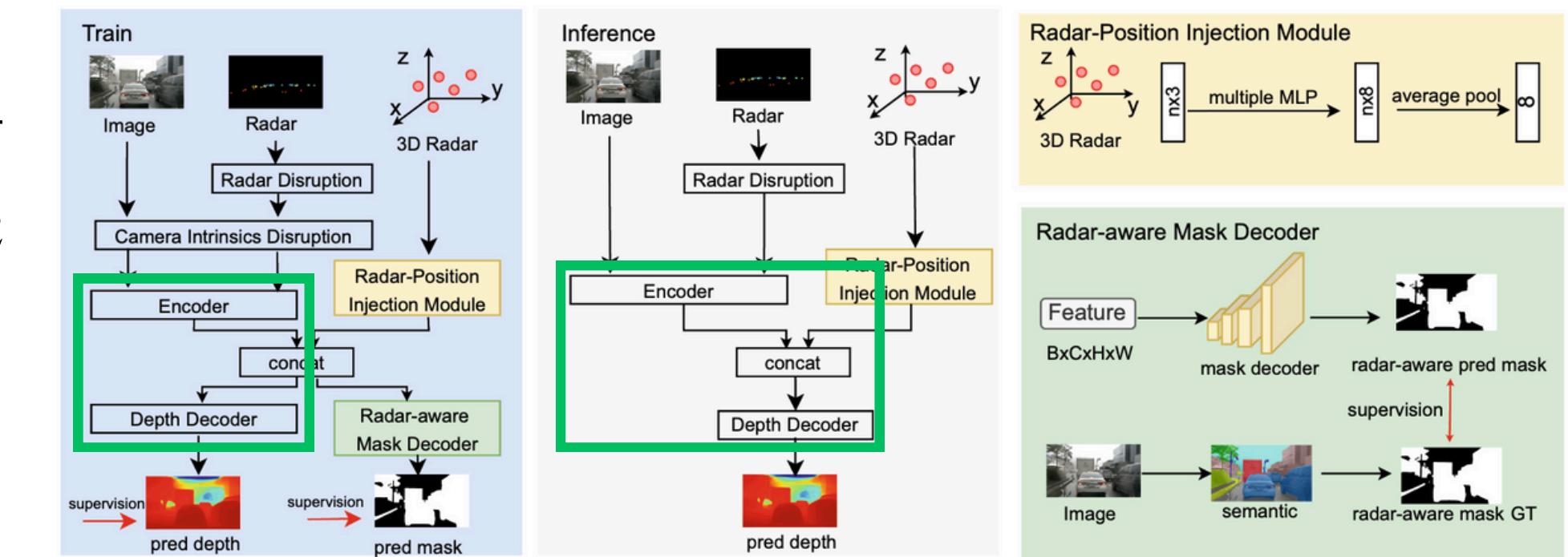
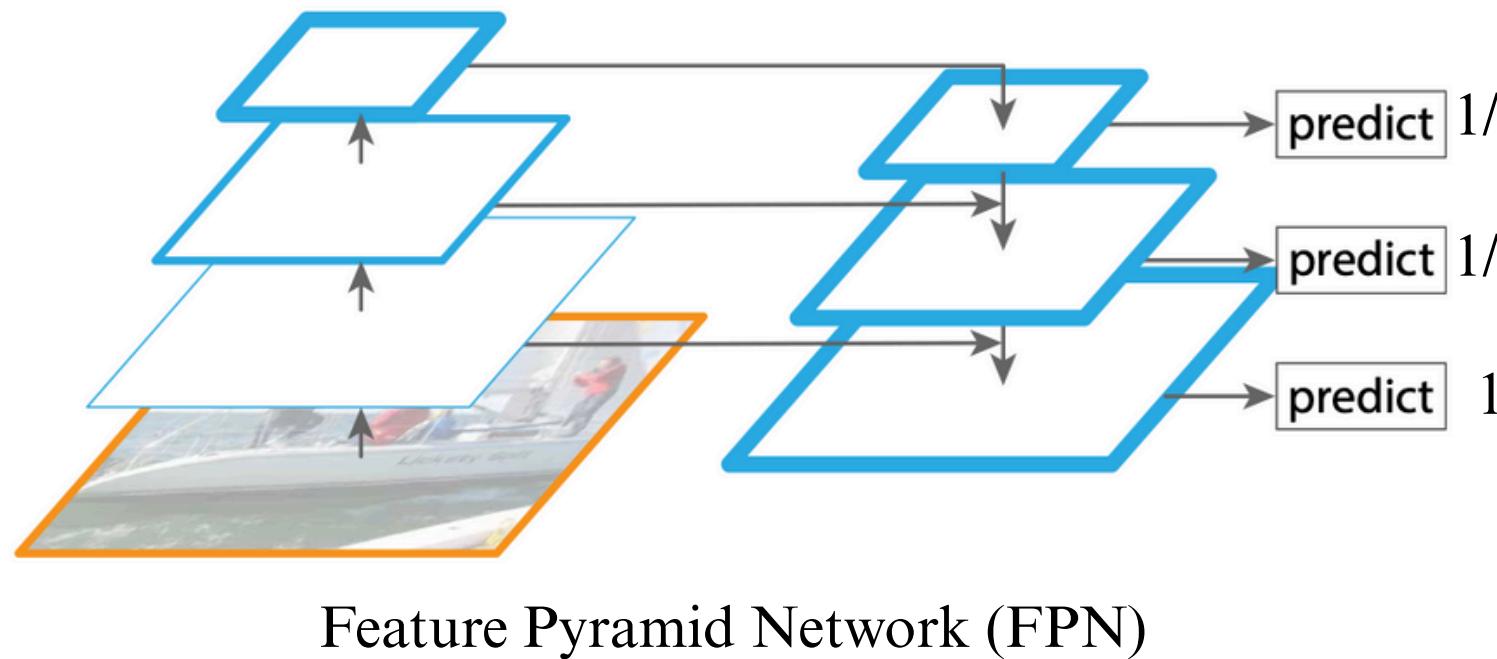


Depth Completion Network

Methods

Depth Encoder and Decoder

- Multi-Scale U-Net 구조를 Encoder, Depth Decoder로 사용하였다. (Backbone : ResNeXt)
 - Encoder에 input으로 Multi-Scale (1, 1/2, 1/4) Image, Radar를 concat하여 사용
 - Decoder는 output으로 Multi-Scale (1, 1/2, 1/4) Dense Depth을 출력
- Multi-Scale Feature Fusion으로 Feature Pyramid Network(FPN) 방식을 사용하였다.
- Encoder를 거친 Feature Map에 Radar-Position 값이 더해진다.



Loss Function

Methods

- Loss Function is **Smooth L1 Loss** for $L_{G_{1-3}}, L_{Mask}$

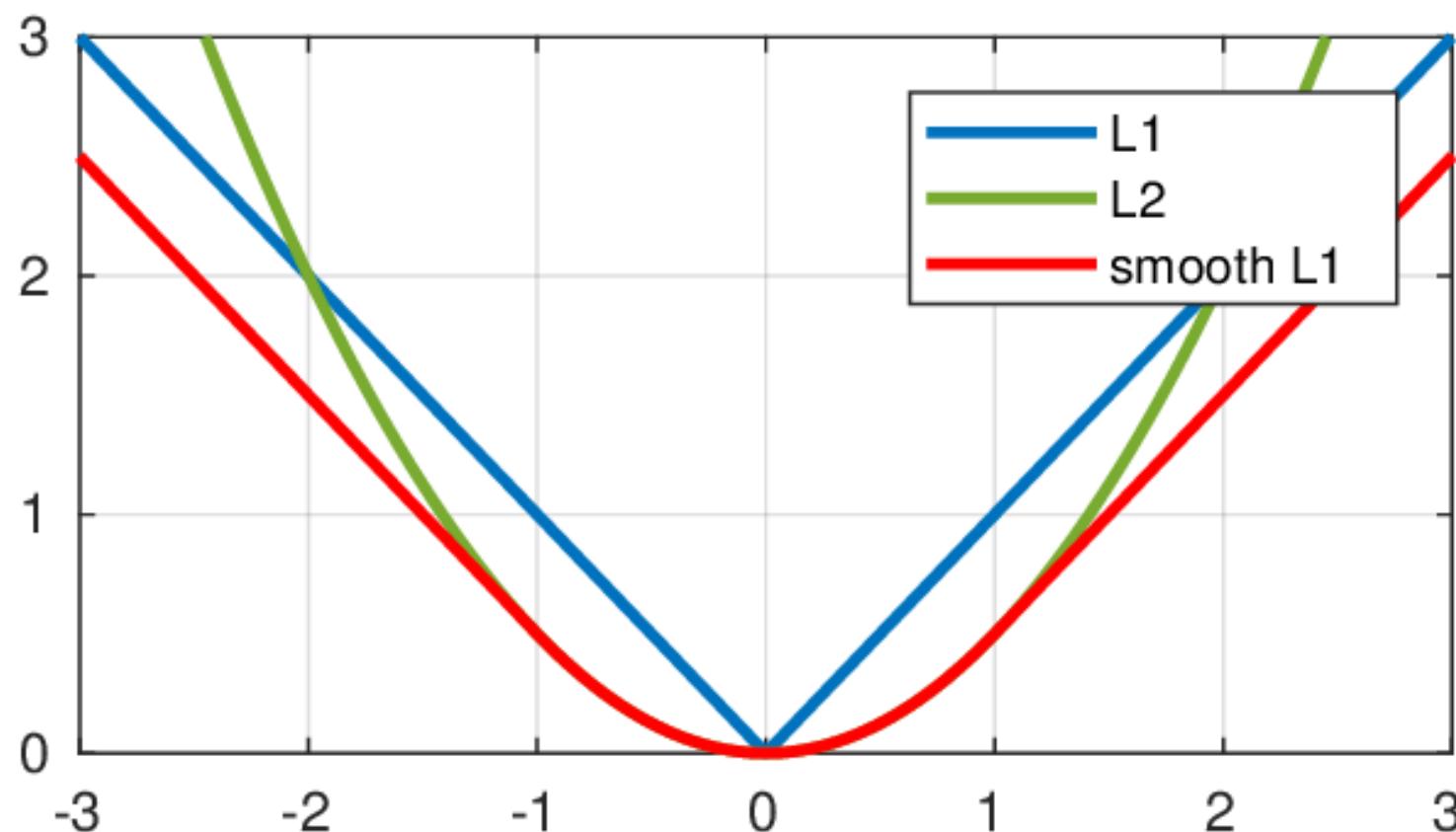
$$L = \lambda_1 L_{G_1} + \lambda_2 L_{G_2} + \lambda_3 L_{G_3} + \lambda_4 L_{Mask}$$

where $L_{G_{1-3}}$: Multi-Scale Depth Losses

L_{Mask} : Auxiliary Mask Loss

λ_{1-3} : Control Weight for Depth Losses (1, 0.5, 0.25)

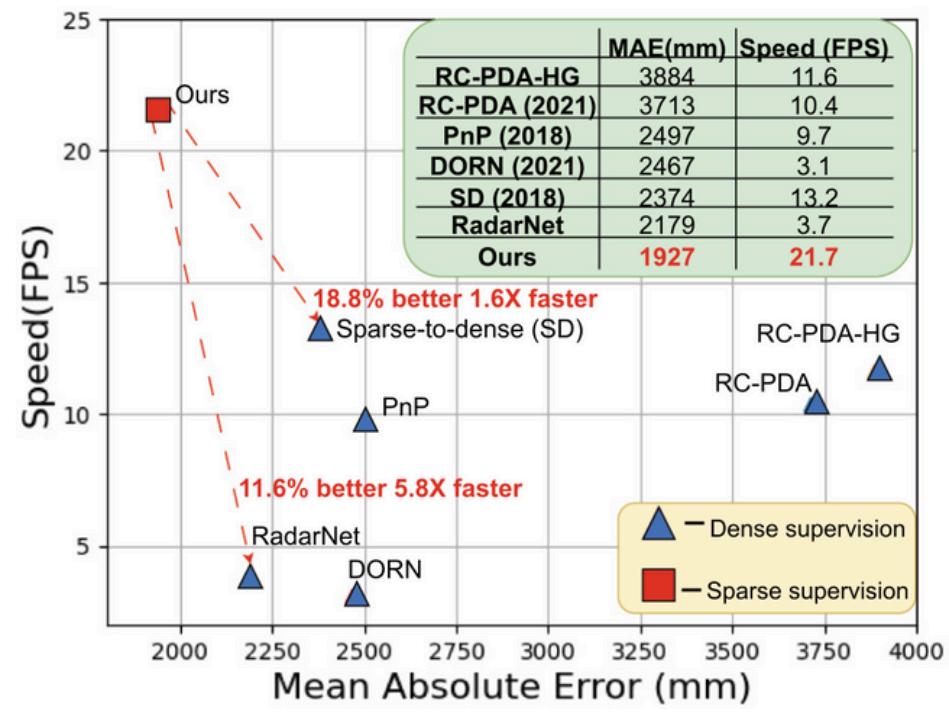
λ_4 : Control Weight for Auxiliary Mask Loss



$$\text{smooth}_{L_1}(\rho) = \begin{cases} 0.5\rho^2 & \text{if } |\rho| < 1 \\ |\rho| - 0.5 & \text{otherwise,} \end{cases}$$

where $\rho = \hat{d}(x) - d_{gt}(x)$: for $L_{G_{1-3}}$
 $\rho = \hat{seg}(x) - seg_{gt}(x)$: for L_{Mask}

Experiments



(b) Sparse beats Dense Supervision

Method	GT	Radar Frames	Images	OMAE (mm) ↓		MAE (mm) ↓		RMSE (mm) ↓			FPS
				{0–80 m}	{0–50, 70, 80 m}	{0–50, 70, 80 m}	{0–50, 70, 80 m}	{0–50, 70, 80 m}	{0–50, 70, 80 m}		
RC-PDA [21]	25	5	3	3635.4	2225.0	3326.1	3713.6	4156.5	6700.6	7692.8	10.4
RC-PDA with HG [21]	25	5	3	3584.2	2315.7	3485.6	3884.3	4321.6	7002.9	8008.6	11.6
DORN [20]	interpolation	15	1	2152.4	1926.6	2380.6	2467.7	4124.8	5252.7	5554.3	3.1
R4Dyn [7]	7	4	1	-	-	-	-	-	-	6434.0	-
Sparse-to-dense [22]	interpolation	3	1	-	-	-	2374.0	-	-	5628.0	13.2
PnP [33]	interpolation	3	1	-	-	-	2496.0	-	-	5578.0	9.7
RadarNet [30]	161	1	1	2165.9	1727.7	2073.2	2179.3	3746.8	4590.7	4898.7	3.7
Ours	1	1	1	1817.0	1524.5	1822.9	1927.0	3567.3	4303.6	4609.6	21.7

Table 1. Comparison with SOTA methods in the Radar-Camera Depth Completion Task

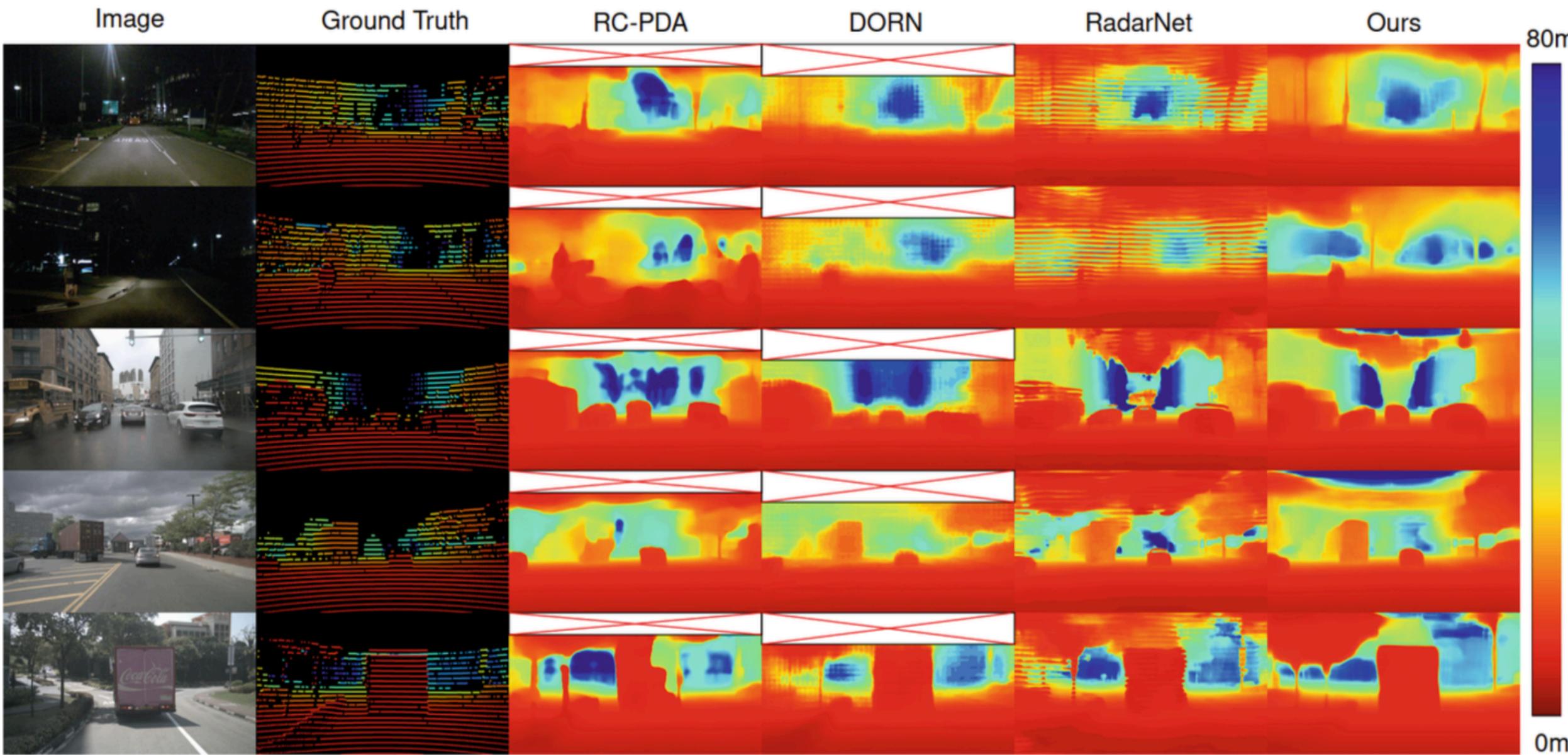


Figure 1. Quailtative results on nuScense test set

Experiments

Experiments

Table 2. Effectiveness of Compensation Part.

Radar-Position Injection Module	Radar-aware Mask Decoder	OMAE 0–80 m (mm)	MAE 0–80 m (mm) ↓	RMSE 0–80 m (mm) ↓
✗	✗	2104.3	2201.1	5103.3
✓	✗	2033.9	2116.4	4970.5
✓	✓	1817.0	1927.0	4609.6

Table 3. Retraining LiDAR Depth Completion Methods on Radar Depth Completion Tasks.

Original Input	Method	OMAE 0–80 m (mm)	MAE 0–80 m (mm) ↓	RMSE 0–80 m (mm) ↓
LiDAR-camera	CompletionFormer [37]	3088.5	3100.7	6285.8
LiDAR-camera	DySPN [18]	3452.1	3549.2	6801.9
Radar-camera	Ours	1817.0	1927.0	4609.6

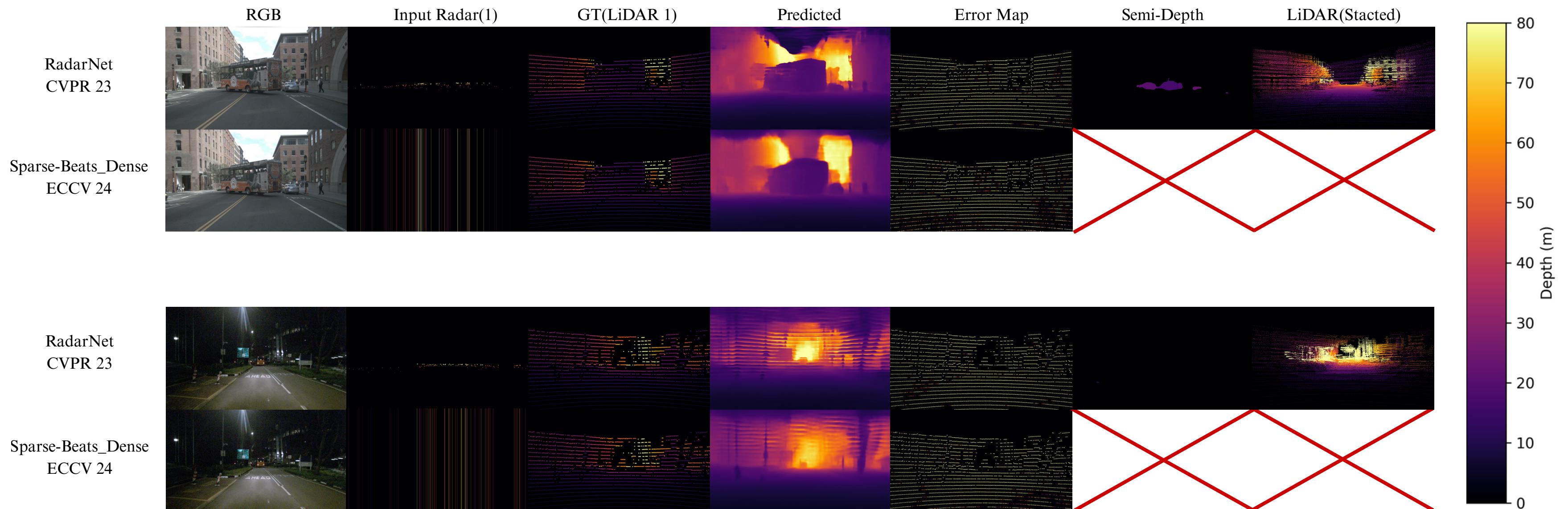
Table 4. Aligning Backbones of Dense Supervision Methods.

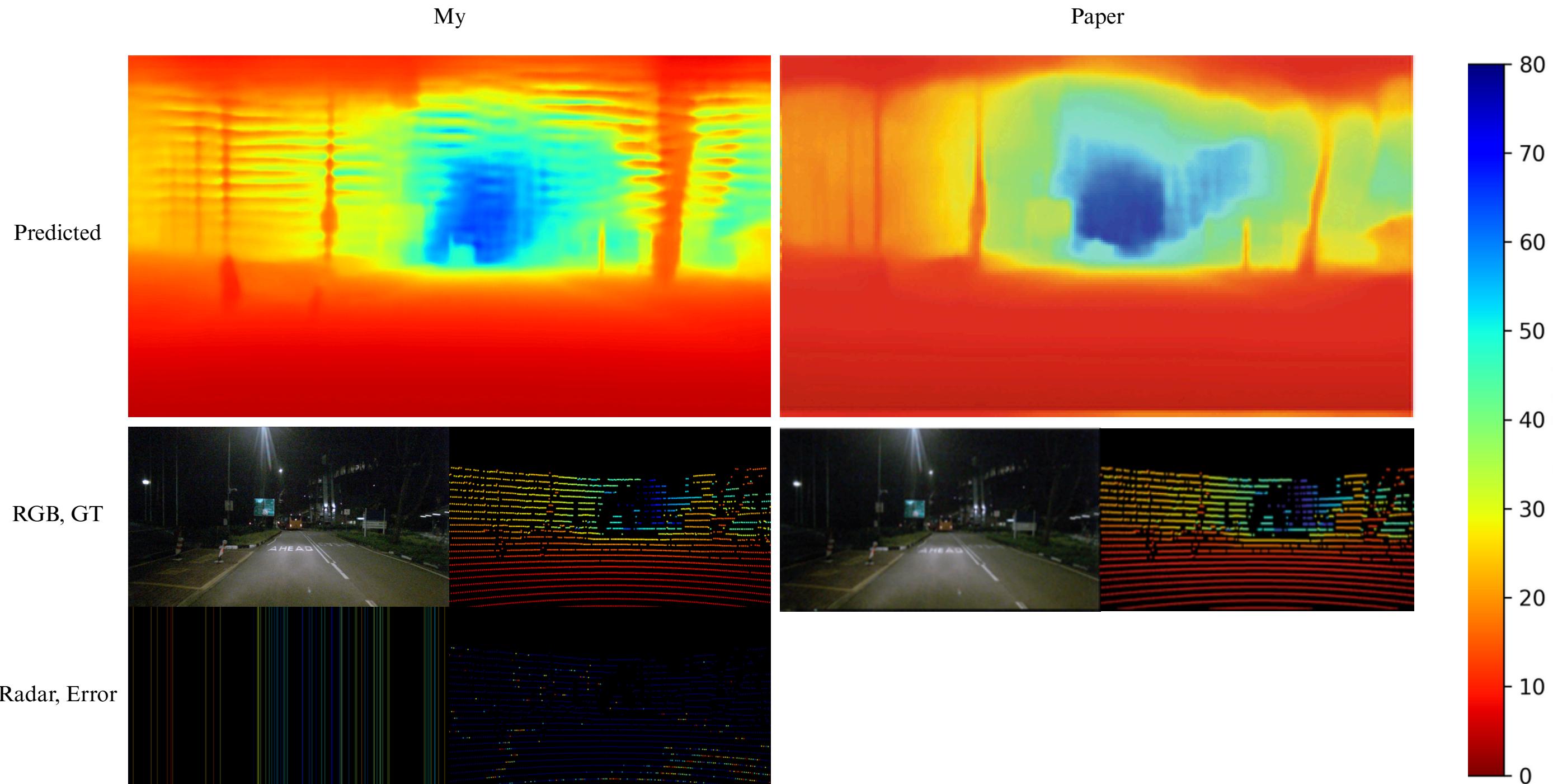
	GT	OMAE 0–80 m (mm)	MAE 0–80 m (mm) ↓	RMSE 0–80 m (mm) ↓
RadarNet [30]	161	3076.7	3104.5	6291.1
DORN [20]	interpolation	2775.0	2922.3	5895.6
RC-PDA [21]	25	2924.6	2820.1	6096.6
Disruption-Compensation	1	1817.0	1927.0	4609.6

Discussion

Inference - RadarNet, SBD

Discussion

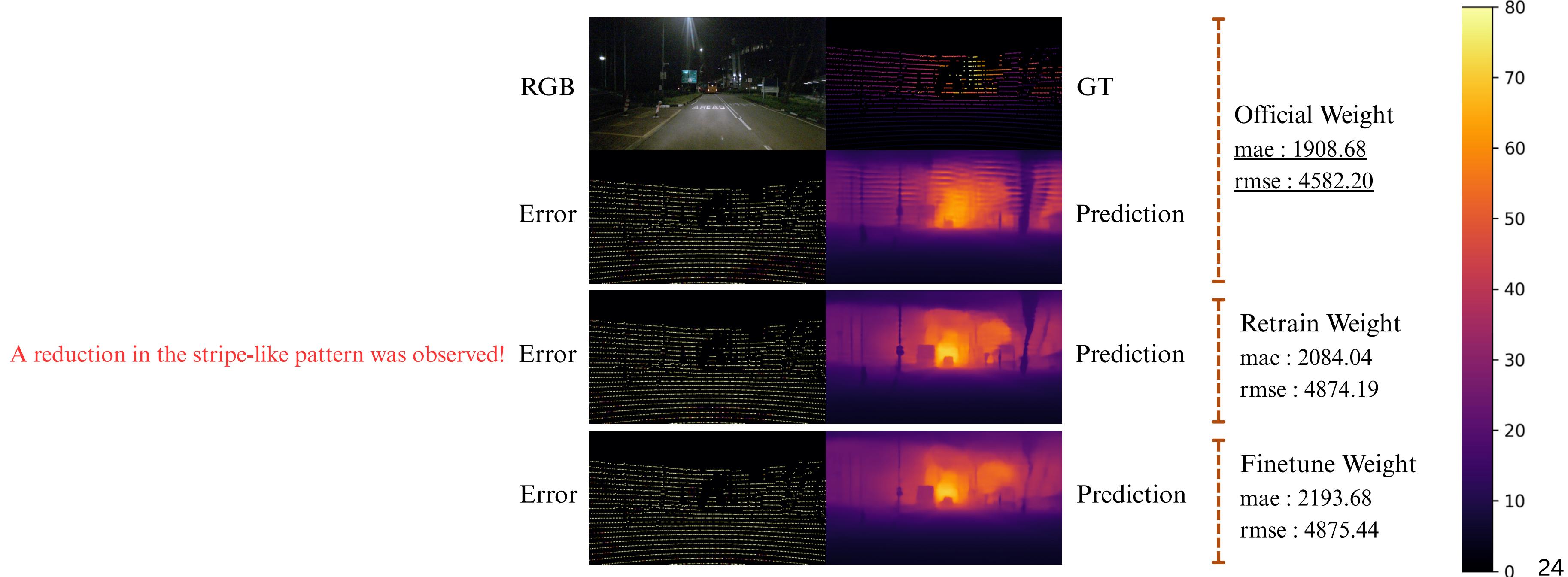




Reproducing - SBD

Discussion

- Finetune Sparse-Beats-Dense with Stacked Lidar GT(161) on reproduced weight
- Cherry-Picking
- Stripe-like Pattern is not completely solved



References

Paper

- [1] Li, Huadong, et al. "Sparse beats dense: Rethinking supervision in radar-camera depth completion." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.
- [2] Singh, Akash Deep, et al. "Depth estimation from camera image and mmwave radar point cloud." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
- [3] Long, Yunfei, et al. "Radar-camera pixel depth association for depth completion." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
- [4] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Smooth L1 Loss

- <https://lunarmonk.wordpress.com/2018/10/31/how-to-implement-loss-functions-in-tensorflow/>



Thank you

JinYong Kim

E-mail: valere2709@icloud.com
Phone: (+82) 10-7317-2709

