OccAM's Laser: Occlusion-based Attribution Map for 3D Object Detectors on LiDAR Data

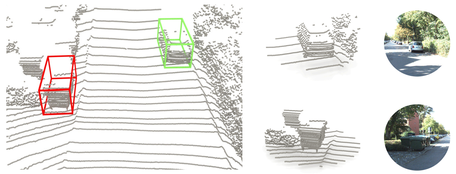
David Schinagl[1,2] Georg Krispel[1] Horst Possegger[1] Peter M. Roth[3,4] Horst Bischof[1,2]

[1]Graz University of Technology [2]Christian Doppler Laboratory for Embedded Machine Learning [3]Technical University of Munich [4]University of Veterinary Medicine, Vienna

try it on your own detector

Overview

The explainability of 3D object detectors on LiDAR data is largely unexplored - a major concern in safety-critical domains.
Can we explain these detections & what are class-specific features?



Introduce occlusion-based attribution maps for 3D object detection

Attribution Map Computation

Input $\mathcal{X} = \{\mathbf{x}_j\}_{j=1\ldots M}$ Point cloud — $F(\mathcal{X}) = \mathcal{D} = \{\mathbf{d}_k\}$ Detections

Masking $\Omega = \{\omega_j\}_{j=1\ldots M}$ Binary mask — $\widetilde{\mathcal{X}} = \{\mathbf{x}_j \in \mathcal{X} \mid \omega_j = 1\}$ Sampled point cloud

$F(\widetilde{\mathcal{X}}) = \widetilde{\mathcal{D}} = \{\widetilde{\mathbf{d}}_l\}$ Detections in sample

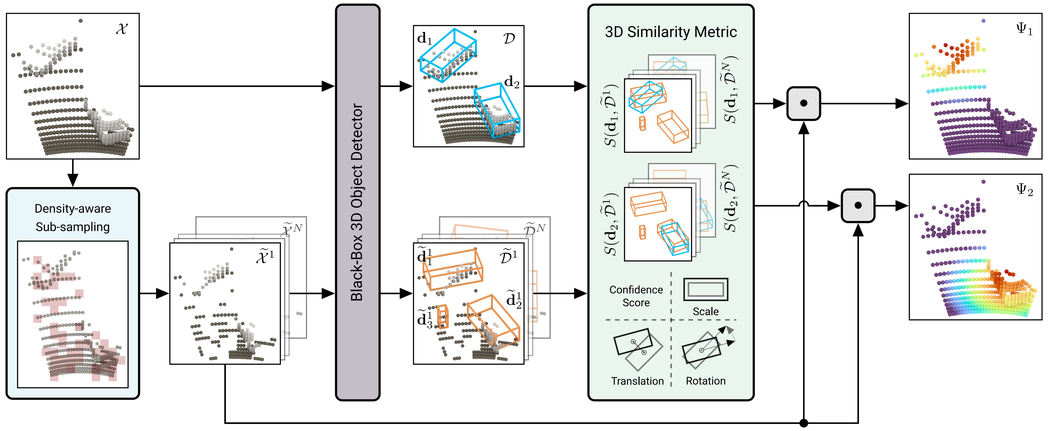
Ideally $\psi_{k,j} = \mathbb{E}_{\forall\Omega}[S(\mathbf{d}_k, \widetilde{\mathcal{D}}) | \omega_j = 1]$ Importance of point j on $\mathbf{d}_k$ — $S(\mathbf{d}_k, \widetilde{\mathcal{D}})$ evaluates detection despite mask

Approx. $\psi_{k,j} \approx \frac{1}{\mathbb{E}[\omega_j]\cdot N} \sum_{i=1}^{N} S(\mathbf{d}_k, \widetilde{\mathcal{D}}^i)\omega_j^i$ where $\{\Omega^i\}_{i=1\ldots N}$ is a set of random masks [3]

References

[1] Geiger et al. Are we ready for autonomous driving? The KITTI vision benchmark suite. *CVPR'12.*
[2] Lang et al. PointPillars: Fast Encoders for Object Detection from Point Clouds. *CVPR'19.*
[3] Petsiuk et al. Black-box Explanation of Object Detectors via Saliency Maps. *CVPR'21.*
[4] Shi et al. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *CVPR'20.*
[5] Yan et al. SECOND: Sparsely Embedded Convolutional Detection. *Sensors 18(10), 2018.*

OccAM

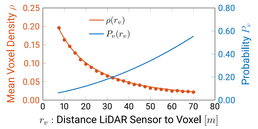


Voxel-based Sampling

- Reduce mask search space
- Similar to real occlusions
- Point-level via random grid shift
- Keep points in voxel with probability P_v

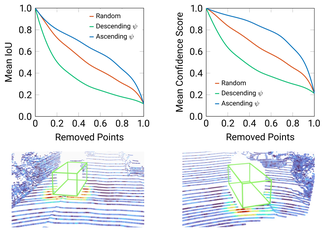
Density Awareness

- Challenge model properly
- Point density $\rho(r_v) \approx \frac{1}{a \cdot r_v^2 + b \cdot r_v + c}$
- Adapt P_v accordingly $P_v(r_v) = \frac{\lambda}{\rho(r_v)}$

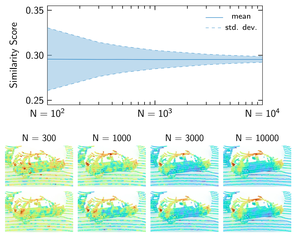


Evaluations

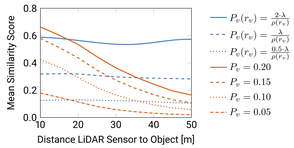
Point Dropping (within boxes)



Convergence

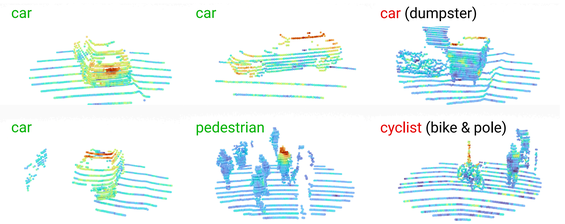


Density-aware Sampling



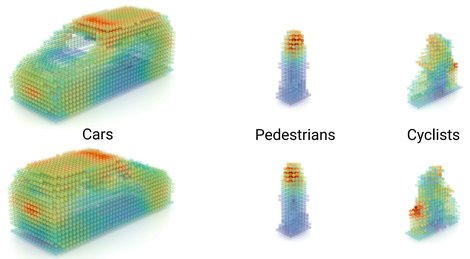
PointPillars [2] on KITTI [1]

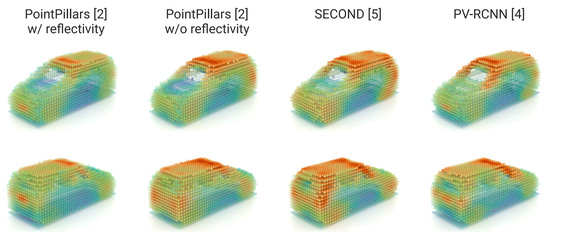
Attribution Map Examples



PointPillars [2] on KITTI [1]

Average Attribution Maps



PointPillars [2] on KITTI [1]

Architecture Comparison