Beyond One Shot, Beyond One Perspective: Cross-View and Long-Horizon Distillation for Better LiDAR Representations

Supplementary Material

Table of Contents

A. Additional Implementation Details

In this section, we provide a comprehensive overview of the datasets, hyperparameter settings, and the training and evaluation protocols employed in our experiments.

A.1. Datasets

In this work, we conduct extensive experiments across varying driving datasets, covering both urban and campus driving environments, various LiDAR sensor configurations, and a wide range of static and dynamic object distributions. Our datasets span real-world and synthetic scenarios, enabling a comprehensive evaluation of model performance across different environmental conditions. Below, we provide a detailed overview of each dataset used in this work:

- **nuScenes [5]:** A large-scale, multi-modality dataset designed for autonomous driving, featuring a 32-beam LiDAR and six cameras capturing diverse urban driving scenes. The dataset consists of $1,000$ sequences collected in Boston and Singapore, with a standard split of 700 training, 150 validation, and 150 testing scenes. For pretraining, we adopt the SLidR [18] protocol, partitioning the training set into a mini-train/val split (600 training, 100 validation). To systematically analyze model performance under different data availability levels, we construct sub-training sets by uniformly sampling 1%, 5%, 10%, 25%, and 100% of the training data for fine-tuning.
- **SemanticKITTI [1]:** A large-scale dataset tailored for semantic scene understanding in autonomous driving, featuring high-resolution 3D point clouds captured by a 64-beam LiDAR sensor. It provides dense point-wise annotations for all 22 sequences from the KITTI Odometry Benchmark [6], covering a diverse range of driving environments, including urban streets, highways, and residential areas. This dataset serves as a benchmark for evaluating semantic segmentation models in real-world autonomous driving scenarios. To systematically assess model performance in data-efficient settings, we construct a 1% sub-training set by uniformly sampling training scans, enabling experiments in low-data regimes.
- **Waymo Open [21]:** A large-scale dataset collected from real-world autonomous driving scenarios, comprising $1,150$ driving sequences – 798 for training, 202 for validation, and 150 for testing. The dataset is captured using a multi-LiDAR setup, consisting of one mid-range and four short-range LiDAR sensors, enabling the collection of dense and diverse point clouds across different traffic conditions. To assess model performance in data-efficient settings, we construct a 1% sub-training set through uniform sampling of the training scans.
- **ScribbleKITTI [22]:** A weakly labeled variant of the SemanticKITTI dataset [1], where annotations are provided as sparse line scribbles instead of dense point-wise labels. This dataset is specifically designed to advance research in weakly supervised learning for semantic segmentation in autonomous driving, reducing the reliance on expensive manual annotations. By leveraging its sparse annotations, models can be trained to learn robust semantic representations with limited supervision. In this work, we construct 1% and 10% sub-training sets through uniform sampling to systematically investigate the effectiveness of weak supervision in data-efficient learning scenarios.
- **RELLIS-3D [7]:** A multimodal dataset collected in off-road environments at the Rellis Campus of Texas A&M University, specifically designed to support research in unstructured terrain scenarios. This dataset contains diverse and complex scenes with vegetation, rough surfaces, and varying elevations, posing challenges distinct from structured urban driving. In this work, we construct

1% and 10% sub-training sets to systematically evaluate model generalization in non-urban environments.

- **SemanticPOSS [14]:** A small-scale dataset collected in off-road environments at Peking University, designed to evaluate the performance of autonomous perception systems in complex, unstructured scenes. It captures diverse scenarios involving both static and dynamic objects, featuring dense point clouds that pose challenges distinct from structured urban settings. The dataset comprises 6 sequences, providing a compact yet challenging benchmark for semantic segmentation. In this work, we utilize sequences 00 and 01 to form a half sub-training set, while sequences 00 to 05 (excluding 02 for validation) constitute the full training set, ensuring a systematic evaluation of model performance in off-road conditions.
- **SemanticSTF [24]:** A dataset specifically designed to evaluate the robustness of LiDAR-based perception models under adverse weather conditions. It includes four challenging scenarios: *"snowy"*, *"dense fog"*, *"light fog"*, and *"rain"*, simulating real-world environmental variations that can significantly impact LiDAR sensor performance. Captured across diverse outdoor settings, this dataset serves as a critical benchmark for assessing the degradation of semantic segmentation models when exposed to extreme weather conditions. To systematically analyze model performance in data-efficient learning settings, we construct both half and full sub-training sets by uniformly sampling training scans.
- **SynLiDAR [23]:** A large-scale synthetic dataset generated using Unreal Engine 4, designed to simulate diverse LiDAR perception scenarios in unstructured virtual environments. It consists of 13 sequences, totaling $198,396$ scans, providing high-quality point clouds that closely mimic real-world sensor data while eliminating the need for costly manual annotations. The dataset covers various terrains and object distributions, making it a valuable resource for studying domain adaptation and generalization in LiDAR-based perception models. In this work, we construct 1% and 10% sub-training sets through uniform sampling to systematically evaluate model performance.
- **DAPS-3D [8]:** A dataset comprising two subsets: DAPS-1, a semi-synthetic collection featuring large-scale 3D scenes, and DAPS-2, which contains real-world LiDAR scans recorded by a cleaning robot operating in VDNH Park, Moscow. This dataset is specifically designed to facilitate research on domain adaptation and semi-synthetic training paradigms by bridging the gap between synthetic and real-world data. In this work, we extract training scans from the sequence "38-18_7_72_90" within the DAPS-1 subset and construct both half and full sub-training sets to evaluate model performance.
- **Synth4D [17]:** A synthetic dataset generated using the CARLA simulator, designed to replicate real-world driving conditions with Velodyne LiDAR sensors. It comprises two subsets: Synth4D-KITTI and Synth4D-nuScenes, each crafted to closely resemble their respective real-world counterparts. In this work, we utilize the Synth4D-nuScenes subset and construct 1% and 10% sub-training sets by uniformly sampling training scans to assess model performance in low-data regimes.
- **nuScenes-C [9]:** An extension of the nuScenes dataset [3], designed to evaluate the robustness of LiDAR-based perception models under challenging environmental conditions. The dataset introduces eight types of synthetic corruptions – *"fog"*, *"wet ground"*, *"snow"*, *"motion blur"*, *"beam missing"*, *"crosstalk"*, *"implement echo"*, and *"cross-sensor"* – each with three severity levels: easy, moderate, and hard. These corruptions simulate the real-world effects of sensor degradation and environmental disturbances, such as bad weather or sensor malfunctions, that can degrade perception performance. This dataset serves as a benchmark to assess model resilience against these distortions, providing a comprehensive tool to evaluate the reliability and robustness of perception systems, particularly in autonomous driving applications.

A.2. Training Configurations

Image Preprocessing. For image-based inputs, we apply standard data augmentation techniques. Specifically, we apply horizontal flipping with a 50% probability and resize all input images to a fixed resolution of 448×224 pixels to ensure consistency across different datasets. Following ScaLR [16], in this work, we do not utilize any Vision Foundation Models (VFMs) to generate superpixels in this work.

Point Cloud Preprocessing. For LiDAR point clouds, we employ a set of geometric transformations to enhance data diversity during training. We apply random rotations around the z-axis within the range of $[-180°, 180°]$, which accounts for minor variations in sensor orientation. Additionally, we randomly flip the point cloud along the x-axis and y-axis with a probability of 50% each. To introduce scale invariance, we apply a random scaling factor sampled uniformly from the range $[0.95, 1.05]$. Finally, we voxelize the point cloud using a cylindrical partitioning strategy with a voxel resolution of 0.1 meters.

Optimization. We employ the AdamW optimizer [12] for training, with an initial learning rate of 0.01. To dynamically adjust the learning rate, we adopt the OneCycle learning rate scheduler [20]. The per-GPU batch size is set to 2, resulting in a total batch size of 16 across 8 GPUs. For downstream fine-tuning, we apply a differentiated learning rate strategy. The backbone network is trained with a lower learning rate, while the task-specific head is updated with a learning rate that is 10× higher to facilitate rapid adaptation to new tasks. Fine-tuning experiments are conducted across multiple data regimes, systematically evaluating model performance under varying levels of data availability.

A.3. Evaluation Configurations

This section outlines the evaluation metrics used to assess model performance in 3D semantic segmentation, robustness evaluation, and 3D object detection. Each metric is designed to quantify different aspects of model effectiveness.

3D Semantic Segmentation. We follow standard evaluation practices by reporting the Intersection-over-Union (IoU) for each category and the mean IoU (mIoU) across all categories. The IoU for a given class i is defined as:

$$\mathrm{IoU}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i} ~, \tag{1}$$

where TP_i (True Positives) denotes correctly classified points, FP_i (False Positives) represents incorrectly predicted points, and FN_i (False Negatives) accounts for misclassified ground-truth points. Higher IoU values indicate better segmentation accuracy. The mean IoU (mIoU) is computed as the average IoU across all classes:

$$\mathrm{mIoU} = \frac{1}{|\mathbb{C}|} \sum_{i \in \mathbb{C}} \mathrm{IoU}_i ~, \tag{2}$$

where $\mathbb{C}$ represents the set of all semantic categories. mIoU provides a holistic measure of overall segmentation quality.

Robustness Evaluation. To evaluate model robustness against real-world perturbations, we adopt the Corruption Error (CE) and Resilience Rate (RR) metrics, following the Robo3D [9] evaluation protocol. These metrics assess performance under various corruption types, including sensor noise, adverse weather conditions, and motion blur.

- **Corruption Error (CE).** CE quantifies the model's performance degradation under corruption type i, defined as:

  $$\mathrm{CE}_i = \frac{\sum_{j=1}^{3}(1 - \mathrm{IoU}_{i,j})}{\sum_{j=1}^{3}(1 - \mathrm{IoU}_{i,j}^{\mathrm{base}})} ~, \tag{3}$$

  where $\mathrm{IoU}_{i,j}$ represents the mIoU under corruption type i at severity level j, and $\mathrm{IoU}_{i,j}^{\mathrm{base}}$ is the mIoU of a baseline model under the same conditions. Lower CE values indicate greater robustness, meaning the model maintains performance even under significant corruption.
- **Resilience Rate (RR).** RR measures the model's ability to recover from corruptions, computed as:

  $$\mathrm{RR}_i = \frac{\sum_{j=1}^{3}\mathrm{IoU}_{i,j}}{3 \times \mathrm{IoU}^{\mathrm{clean}}} ~, \tag{4}$$

  where $\mathrm{IoU}^{\mathrm{clean}}$ represents the mIoU on the "clean" validation set, which serves as a reference for the model's performance under ideal conditions. A higher RR signifies greater resilience to external disturbances.

Additionally, we report the mean Corruption Error (mCE) and mean Resilience Rate (mRR) across all corruption types to summarize overall robustness.

3D Object Detection. We evaluate 3D object detection performance using the mean Average Precision (mAP) and the nuScenes Detection Score (NDS), following nuScenes [3].

- **Mean Average Precision (mAP).** mAP is computed by averaging the Average Precision (AP) across object classes and distance-based matching thresholds:

  $$\mathrm{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \mathrm{AP}_{c,d} ~, \tag{5}$$

  where $\mathbb{C}$ denotes the set of object categories, and $\mathbb{D} = \{0.5, 1, 2, 4\}$ represents different distance thresholds used for AP computation. mAP evaluates detection accuracy by considering both localization precision and recall across various object classes and distances.
- **nuScenes Detection Score (NDS).** NDS provides a comprehensive evaluation by combining mAP with additional detection quality metrics, computed as:

  $$\mathrm{NDS} = \frac{1}{10}[5\mathrm{mAP} + \sum_{\mathrm{mTP} \in \mathrm{TP}} (1 - \min(1, \mathrm{mTP}))] ~, \tag{6}$$

  where mTP is the mean true positive metric calculated across all object classes, representing the average number of true positives per class:

  $$\mathrm{mTP} = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \mathrm{TP}_c ~. \tag{7}$$

  NDS captures both detection accuracy and localization precision, making it a more holistic metric for evaluating 3D object detection performance.

B. Additional Quantitative Results

In this section, we provide a detailed evaluation of class-wise LiDAR semantic segmentation performance, showing the advantages of our approach over existing methods.

B.1. Class-Wise Linear Probing Results

Tab. B presents the class-wise IoU scores for linear probing experiments. To ensure a fair comparison, we reimplemented ScaLR [16] based on the technical details provided in the original paper and publicly available code. The results show that LiMA consistently outperforms prior state-of-the-art methods across most categories, with particularly notable gains in the segmentation of dynamic objects. This improvement highlights LiMA's capability to capture rich spatiotemporal representations, which are crucial for understanding moving entities in autonomous driving scenarios.

B.2. Class-Wise Fine-Tuning Results

We report the class-wise IoU scores for 1% fine-tuning experiments in Tab. C. Compared to the baseline [16], LiMA achieves significant performance gains across most categories. These improvements stem from LiMA's ability to leverage temporal feature distillation, effectively capturing long-term dependencies for enhanced feature learning. However, we observe performance degradation in certain underrepresented categories, likely due to class imbalance. This suggests potential avenues for future work, such as incorporating class-aware optimization strategies or reweighting schemes to mitigate the impact of data sparsity.

B.3. Knowledge Distillation Strategy

In the main page, knowledge distillation is employed to transfer long-term temporally fused image features $\mathcal{F}_d$ into LiDAR representations $\mathcal{F}_p$. In this section, we provide a detailed overview of commonly used distillation strategies and analyze their effectiveness in this context.

- **Cosine Similarity** enforces alignment between feature representations by measuring their angular difference while being invariant to feature magnitudes. The corresponding loss function is formulated as:

$$\mathcal{L}_{\text{cos}}(\mathcal{F}_d, \mathcal{F}_p) = \frac{1}{M}\sum_{i=1}^{M}(1 - \langle \mathbf{f}_d^i, \mathbf{f}_p^i \rangle) \ , \qquad (8)$$

  where M denotes the number of corresponding point-pixel pairs, and $\langle \cdot, \cdot \rangle$ is the dot product. By minimizing $\mathcal{L}_{\text{cos}}$, the LiDAR representation $\mathcal{F}_p$ is encouraged to learn features with a similar directional alignment to $\mathcal{F}_d$.
- ℓ_2 **distance** objective minimizes the Euclidean distance between the teacher and student representations, enforcing consistency while preserving feature magnitudes:

$$\mathcal{L}_{\text{dist}}(\mathcal{F}_d, \mathcal{F}_p) = \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{f}_d^i - \mathbf{f}_p^i\|_2 \ . \qquad (9)$$

  This formulation ensures that $\mathcal{F}_p$ closely approximates the fine-grained feature structure of $\mathcal{F}_d$.
- **Contrastive Learning** enhances feature discrimination by bringing positive pairs closer while pushing negative pairs apart. The contrastive loss is formulated as:

$$\mathcal{L}_{\text{cont}}(\mathcal{F}_d, \mathcal{F}_p) = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{\langle \mathbf{f}_d^i, \mathbf{f}_p^i \rangle / \tau}}{\sum_{j=1}^{M} e^{\langle \mathbf{f}_d^i, \mathbf{f}_p^j \rangle / \tau}} \ , \qquad (10)$$

  where $\tau > 0$ is a temperature scale factor. This loss function encourages $\mathcal{F}_p$ to learn meaningful and discriminative features from $\mathcal{F}_d$. In our implementation, we follow PPKT [11] and randomly sample 4096 pairs to construct a contrastive objective.

Table A. Comparison of **different distillation strategies**.

#	Distillation	nuScenes		KITTI	Waymo
		LP	1%	1%	1%
(a)	Cosine Similarity	51.23	48.23	47.34	47.84
(b)	ℓ_2 Distance	**56.65**	**51.29**	**50.44**	**51.35**
(c)	Contrastive Learning	54.23	50.34	48.86	50.23
(d)	KL Divergence	55.34	49.53	49.43	50.46

- **Kullback-Leibler (KL) Divergence** measures the discrepancy between two probability distributions, aligning the student's predictive distribution with the teacher's. The KL divergence loss is defined as:

$$\mathcal{L}_{\text{KL}}(\mathcal{F}_d, \mathcal{F}_p) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{f}_d^i \log\frac{\mathbf{f}_d^i}{\mathbf{f}_p^i} \ . \qquad (11)$$

  By minimizing $\mathcal{L}_{\text{KL}}$, $\mathcal{F}_p$ is encouraged to approximate the predictive distribution of $\mathcal{F}_d$, leading to improved generalization performance.

Tab. A presents a comparative analysis of various distillation strategies. Our results indicate that the ℓ_2 distance metric achieves the highest overall performance. The cosine similarity-based approach, which focuses solely on angular alignment, proves inadequate for feature alignment as it disregards magnitude differences – an essential aspect of representation learning. Contrastive learning aims to enhance feature discrimination by pulling positive pairs closer while pushing negatives apart, but this objective may not be optimal for direct feature matching in distillation. KL divergence effectively aligns predictive distributions but is susceptible to data distribution shifts, particularly in low-data regimes. In contrast, ℓ_2 distance minimizes Euclidean discrepancy directly, ensuring a more stable and effective optimization objective for knowledge transfer.

C. Additional Qualitative Results

In this section, we present additional qualitative examples to provide a visual comparison of the different approaches discussed in the main body of the paper.

C.1. LiDAR Semantic Segmentation Results

Fig. A, Fig. B, and Fig. C showcase qualitative LiDAR semantic segmentation results, comparing LiMA with the baseline method [16] on the *nuScenes* [5], *SemanticKITTI* [1], and *Waymo Open* [21] datasets, respectively. The models were pretrained on *nuScenes* [3] and fine-tuned using 1% of the available annotations from each dataset. As shown, LiMA consistently outperforms the baseline in most categories, especially for dynamic objects such as vehicles.

In particular, LiMA can leverage long-term temporal features and align spatial-temporal information, enabling more accurate segmentation in scenarios where objects exhibit rapid motion or change. The error maps further highlight its superior performance, with fewer misclassifications and better localization of the dynamic objects compared to the baseline, showing the robustness of LiMA to both the dataset's limited annotations and its inherent complexity.

C.2. 3D Object Detection Results

Fig. D presents qualitative LiDAR detection results, comparing LiMA with the baseline method [16] on the *nuScenes* [3] dataset, where models were fine-tuned using 5% of the available annotations. For better visualization, the confidence threshold is set to 0.5. As shown, LiMA consistently outperforms the baseline by producing accurate and confident bounding boxes, particularly for small objects.

The long-term memory mechanism effectively propagates and aggregates temporal features, improving object localization and mitigating false positives. In particular, it demonstrates superior performance in detecting dynamic objects such as pedestrians and vehicles, which often suffer from motion-induced distortions. The error analysis highlights LiMA's ability to maintain spatial-temporal consistency, resulting in fewer missed detections and more stable bounding box predictions.

C.3. Cosine Similarity Results

In Fig. E, we present additional cosine similarity maps computed during the pretraining phase. These maps provide an intuitive understanding of how well LiMA aligns the image and LiDAR point features within the same semantic space. The cosine similarity score, which measures the angular difference between features, is used to evaluate the consistency and semantic relevance between image and LiDAR data.

As depicted, the query point (indicated by the red dot) exhibits high cosine similarity with both the corresponding image and LiDAR point features projected onto the image plane. This demonstrates our ability to effectively bridge the gap between two sensor modalities – image and LiDAR – by learning a shared feature space that preserves semantic consistency across them. The resulting high similarity scores (represented in red) indicate that LiMA succeeds in aligning the visual and LiDAR representations, enhancing its capacity to transfer knowledge across modalities and improve feature fusion for downstream tasks.

D. Broad Impact & Limitations

In this section, we discuss the broader impact of our proposed LiMA framework, highlighting its contributions to autonomous perception and beyond. Additionally, we outline potential limitations and areas for future improvement.

D.1. Broader Impact

The LiMA framework introduces a novel approach to learning robust LiDAR-based representations through long-term temporal modeling and cross-modal feature alignment. This has several significant implications for both academic research and real-world applications:

Advancing Data-Efficient Perception. By effectively leveraging pretraining with limited labeled data, LiMA reduces reliance on large-scale human annotations, addressing one of the primary bottlenecks in deep learning for 3D perception. This advancement is particularly valuable for safety-critical applications where data collection is expensive or infeasible, such as autonomous driving and robotics.

Improving Robustness in Dynamic Environments. Through explicit modeling of temporal and spatial dependencies, LiMA enhances the ability to segment dynamic objects with high accuracy. This contributes to improved situational awareness and decision-making for autonomous systems in complex and rapidly changing environments.

Facilitating Cross-Modal Learning. The ability to align image and LiDAR features in a shared representation space enhances multi-sensor fusion strategies. This can benefit perception tasks beyond segmentation, including object detection, tracking, and scene understanding, enabling more effective deployment in real-world settings.

Potential for Transfer Learning and Generalization. The insights from LiMA's pretraining strategies can be applied to a broader range of 3D vision tasks, fostering new research directions in self-supervised learning, domain adaptation, and transfer learning for sparse and multimodal data.

D.2. Potential Limitations

Despite its advantages, LiMA has certain limitations that should be considered for future research:

Sensitivity to Sensor Calibration. The framework assumes well-calibrated LiDAR and camera sensors for effective cross-modal feature alignment. Misalignment in real-world deployments may lead to suboptimal feature fusion. Future work could explore self-calibration mechanisms or uncertainty-aware fusion techniques.

Dependence on Temporal Consistency. LiMA relies on long-term temporal information, which may not be optimal in scenarios where past observations are unreliable due to sensor noise, occlusions, or drastic environmental changes. Investigating adaptive temporal modeling techniques could further enhance robustness.

Table B. The **per-class IoU scores** of state-of-the-art pretraining methods pretrained and linear-probed on the *nuScenes* [3, 5] dataset. All scores are given in percentage (%). The Best and 2nd Best scores under each group are highlighted in Green and Red.

Method	**mIoU**	barrier ⬛	bicycle 🟥	bus 🟥	car 🟧	construction vehicle 🟧	motorcycle 🟥	pedestrian 🟦	traffic cone ⬛	trailer 🟧	truck 🟥	driveable surface 🟦	other flat 🟥	sidewalk 🟪	terrain 🟩	manmade 🟫	vegetation 🟩
Random	8.1	0.5	0.0	0.0	3.9	0.0	0.0	0.0	6.4	0.0	3.9	59.6	0.0	0.1	16.2	30.6	12.0
Distill: None																	
PointContrast [25]	**21.9**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DepthContrast [28]	**22.1**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ALSO [2]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BEVContrast [19]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ResNet-50																	
PPKT [11]	35.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SLidR [18]	39.2	**44.2**	**0.0**	**30.8**	**60.2**	**15.1**	**22.4**	**47.2**	**27.7**	**16.3**	**34.3**	**80.6**	**21.8**	**35.2**	**48.1**	**71.0**	**71.9**
ST-SLidR [13]	40.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TriCC [15]	38.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Seal [10]	**45.0**	**54.7**	**5.9**	**30.6**	**61.7**	**18.9**	**28.8**	**48.1**	**31.0**	**22.1**	**39.5**	**83.8**	**35.4**	**46.7**	**56.9**	**74.7**	**74.7**
CSC [4]	**46.0**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HVDistill [27]	39.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ViT-S																	
PPKT [11]	38.6	43.8	0.0	31.2	53.1	15.2	0.0	42.2	16.5	18.3	33.7	79.1	37.2	45.2	52.7	75.6	74.3
SLidR [18]	44.7	45.0	8.2	34.8	58.6	**23.4**	**40.2**	43.8	19.0	22.9	40.9	82.7	38.3	47.6	53.9	77.8	77.9
Seal [10]	45.2	48.9	**8.4**	30.7	68.1	17.5	37.7	**57.7**	17.9	20.9	40.4	83.8	36.6	44.2	54.5	76.2	**79.3**
SuperFlow [26]	46.4	49.8	**6.8**	45.9	63.4	18.5	31.0	**60.3**	**28.1**	**25.4**	**47.4**	86.2	38.4	47.4	56.7	74.9	77.8
ScaLR [16]	**49.7**	**58.5**	3.2	**62.4**	**68.8**	20.2	32.3	49.0	**31.8**	21.7	45.9	**90.0**	**39.5**	**53.1**	**62.1**	**78.0**	78.1
LiMA	**54.8**	**61.9**	3.5	**71.6**	**73.3**	**29.3**	**46.8**	53.9	**31.8**	**27.7**	**55.5**	**91.9**	**43.5**	**59.7**	**65.8**	**80.2**	**80.0**
Distill: ViT-B																	
PPKT [11]	40.0	29.6	0.0	30.7	55.8	6.3	22.4	56.7	18.1	**24.3**	42.7	82.3	33.2	45.1	53.4	71.3	75.7
SLidR [18]	45.4	46.7	7.8	46.5	58.7	23.9	34.0	47.8	17.1	**23.7**	41.7	83.4	39.4	47.0	54.6	76.6	77.8
Seal [10]	46.6	49.3	**8.2**	35.1	70.8	22.1	41.7	**57.4**	15.2	21.6	42.6	84.5	38.1	46.8	55.4	77.2	**79.5**
SuperFlow [26]	47.7	45.8	**12.4**	52.6	67.9	17.2	**40.8**	**59.5**	25.4	21.0	47.6	85.8	37.2	48.4	56.6	76.2	78.2
ScaLR [16]	**51.9**	**61.6**	3.1	**70.2**	**70.9**	**25.2**	29.5	48.3	**32.8**	22.3	**49.9**	**90.9**	**45.3**	**57.9**	**64.9**	**79.2**	78.3
LiMA	**56.7**	**63.4**	4.1	**73.7**	**76.4**	**32.8**	**43.4**	54.6	**38.8**	**24.3**	**57.2**	**92.9**	**51.3**	**63.8**	**68.4**	**81.0**	**80.5**
Distill: ViT-L																	
PPKT [11]	41.6	30.5	0.0	32.0	57.3	8.7	24.0	**58.1**	19.5	**24.9**	44.1	83.1	34.5	45.9	55.4	72.5	76.4
SLidR [18]	45.7	46.9	**6.9**	44.9	60.8	22.7	40.6	44.7	17.4	23.0	40.4	83.6	39.9	47.8	55.2	78.1	78.3
Seal [10]	46.8	53.1	**6.9**	35.0	65.0	22.0	**46.1**	**59.2**	16.2	23.0	41.8	84.7	35.8	46.6	55.5	78.4	**79.8**
SuperFlow [26]	48.0	52.3	**12.7**	46.5	64.7	21.4	44.9	56.2	26.7	19.9	43.2	84.2	38.1	47.4	56.9	76.0	79.2
ScaLR [16]	**51.8**	**61.2**	3.4	**65.4**	**72.4**	**25.6**	34.3	51.7	**28.8**	23.6	**50.4**	**90.6**	**44.1**	**55.6**	**64.5**	**79.3**	79.5
LiMA	**56.7**	**63.6**	3.5	**72.4**	**75.0**	**33.7**	**48.5**	55.7	**37.4**	**25.3**	**59.0**	**92.6**	**48.3**	**62.0**	**68.1**	**81.1**	**80.6**

Table C. The **per-class IoU scores** of state-of-the-art pretraining methods pretrained and fine-tuned on *nuScenes* [3, 5] dataset with 1% annotations. All scores are given in percentage (%). The Best and 2nd Best scores under each group are highlighted in Green and Red.

Method	**mIoU**	barrier ■	bicycle ■	bus ■	car ■	construction vehicle ■	motorcycle ■	pedestrian ■	traffic cone ■	trailer ■	truck ■	driveable surface ■	other flat ■	sidewalk ■	terrain ■	manmade ■	vegetation ■
Random	**30.3**	**0.0**	**0.0**	**8.1**	**65.0**	**0.1**	**6.6**	**21.0**	**9.0**	**9.3**	**25.8**	**89.5**	**14.8**	**41.7**	**48.7**	**72.4**	**73.3**
Distill: None																	
PointContrast [25]	32.5	**0.0**	**1.0**	5.6	**67.4**	0.0	**3.3**	**31.6**	5.6	**12.1**	**30.8**	**91.7**	**21.9**	**48.4**	**50.8**	**75.0**	**74.6**
DepthContrast [28]	31.7	**0.0**	0.6	**6.5**	64.7	**0.2**	**5.1**	29.0	**9.5**	**12.1**	29.9	**90.3**	17.8	**44.4**	49.5	73.5	74.0
ALSO [2]	37.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BEVContrast [19]	**37.9**	**0.0**	**1.3**	**32.6**	**74.3**	**1.1**	0.9	**41.3**	**8.1**	**24.1**	**40.9**	89.8	**36.2**	44.0	**52.1**	**79.9**	**79.7**
Distill: ResNet-50																	
PPKT [11]	37.8	**0.0**	**2.2**	20.7	**75.4**	1.2	13.2	45.6	8.5	17.5	38.4	**92.5**	19.2	52.3	56.8	80.1	80.9
SLidR [18]	38.8	**0.0**	1.8	15.4	73.1	**1.9**	19.9	47.2	17.1	14.5	34.5	92.0	27.1	53.6	**61.0**	79.8	82.3
ST-SLidR [13]	40.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TriCC [15]	41.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Seal [10]	**45.8**	**0.0**	**9.4**	**32.6**	**77.5**	**10.4**	**28.0**	**53.0**	**25.0**	**30.9**	**49.7**	**94.0**	**33.7**	**60.1**	59.6	**83.9**	**83.4**
CSC [4]	**47.0**	**0.0**	0.0	**58.7**	74.0	0.1	**40.9**	**58.9**	**31.8**	**23.7**	**45.1**	**92.5**	**33.0**	**56.4**	**62.4**	**81.6**	**84.2**
HVDistill [27]	42.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distill: ViT-S																	
PPKT [11]	40.6	0.0	0.0	25.2	73.5	9.1	6.9	51.4	8.6	11.3	31.1	93.2	**41.7**	58.3	**64.0**	82.0	**82.6**
SLidR [18]	41.2	0.0	0.0	26.6	72.0	**12.4**	15.8	51.4	**22.9**	11.7	35.3	92.9	36.3	58.7	63.6	81.2	82.3
Seal [10]	44.3	20.0	0.0	19.4	**74.7**	10.6	**45.7**	**60.3**	**29.2**	**17.4**	38.1	93.2	26.0	**58.8**	**64.5**	81.9	81.9
SuperFlow [26]	**47.8**	**38.2**	**1.8**	25.8	**79.0**	**15.3**	43.6	**60.3**	0.0	**28.4**	**55.4**	**93.7**	28.8	**59.1**	59.9	**83.5**	**83.1**
ScaLR [16]	45.9	35.8	**6.0**	**57.0**	72.7	0.6	42.7	47.5	3.7	8.4	55.3	92.7	28.3	56.3	62.7	**83.3**	81.3
LiMA	**48.8**	**42.2**	0.8	**66.0**	74.4	0.0	**47.3**	**54.2**	0.0	14.8	**59.2**	**93.8**	**42.6**	58.6	62.0	83.0	81.1
Distill: ViT-B																	
PPKT [11]	40.9	0.0	0.0	24.5	73.5	**12.2**	7.0	51.0	**13.5**	15.4	36.3	93.1	40.4	59.2	63.5	81.7	82.2
SLidR [18]	41.6	0.0	0.0	26.7	73.4	10.3	16.9	51.3	**23.3**	12.7	38.1	93.0	37.7	58.8	63.4	81.6	**82.7**
Seal [10]	46.0	43.0	0.0	26.7	**81.3**	9.9	41.3	**56.2**	0.0	**21.7**	51.6	93.6	**42.3**	**62.8**	64.7	82.6	**82.7**
SuperFlow [26]	48.1	39.1	0.9	30.0	**80.7**	10.3	**47.1**	**59.5**	5.1	**27.6**	55.4	93.7	29.1	61.1	63.5	82.7	**83.6**
ScaLR [16]	**48.9**	**52.8**	**4.1**	**66.6**	71.7	0.2	**44.0**	46.5	11.1	5.8	**56.1**	**93.8**	35.8	**61.7**	**66.8**	**83.7**	81.8
LiMA	**51.3**	**53.2**	**3.6**	**69.0**	78.1	**11.0**	**47.1**	52.4	7.5	4.9	**62.2**	**94.0**	**40.5**	60.3	**66.0**	**85.1**	82.6
Distill: ViT-L																	
PPKT [11]	42.1	0.0	0.0	24.4	78.8	15.1	9.2	54.2	14.3	12.9	39.1	92.9	37.8	59.8	64.9	82.3	83.6
SLidR [18]	42.8	0.0	0.0	23.9	78.8	15.2	20.9	55.0	**28.0**	17.4	41.4	92.2	41.2	58.0	64.0	81.8	82.7
Seal [10]	46.3	41.8	0.0	23.8	**81.4**	**17.7**	46.3	**58.6**	0.0	**23.4**	54.7	93.8	**41.4**	**62.5**	65.0	83.8	**83.8**
SuperFlow [26]	**50.0**	44.5	0.9	22.4	**80.8**	**17.1**	**50.2**	**60.9**	**21.0**	**25.1**	55.1	**93.9**	35.8	61.5	62.6	83.7	83.7
ScaLR [16]	49.1	**46.5**	**4.9**	**70.5**	77.0	2.5	45.9	47.7	9.1	4.9	**55.6**	93.8	35.4	59.4	**66.2**	**84.1**	82.5
LiMA	**53.2**	**54.0**	**5.5**	**71.3**	76.7	11.2	**59.3**	54.2	10.2	9.4	**61.0**	**94.7**	**43.4**	**63.4**	**68.9**	**84.2**	**84.1**

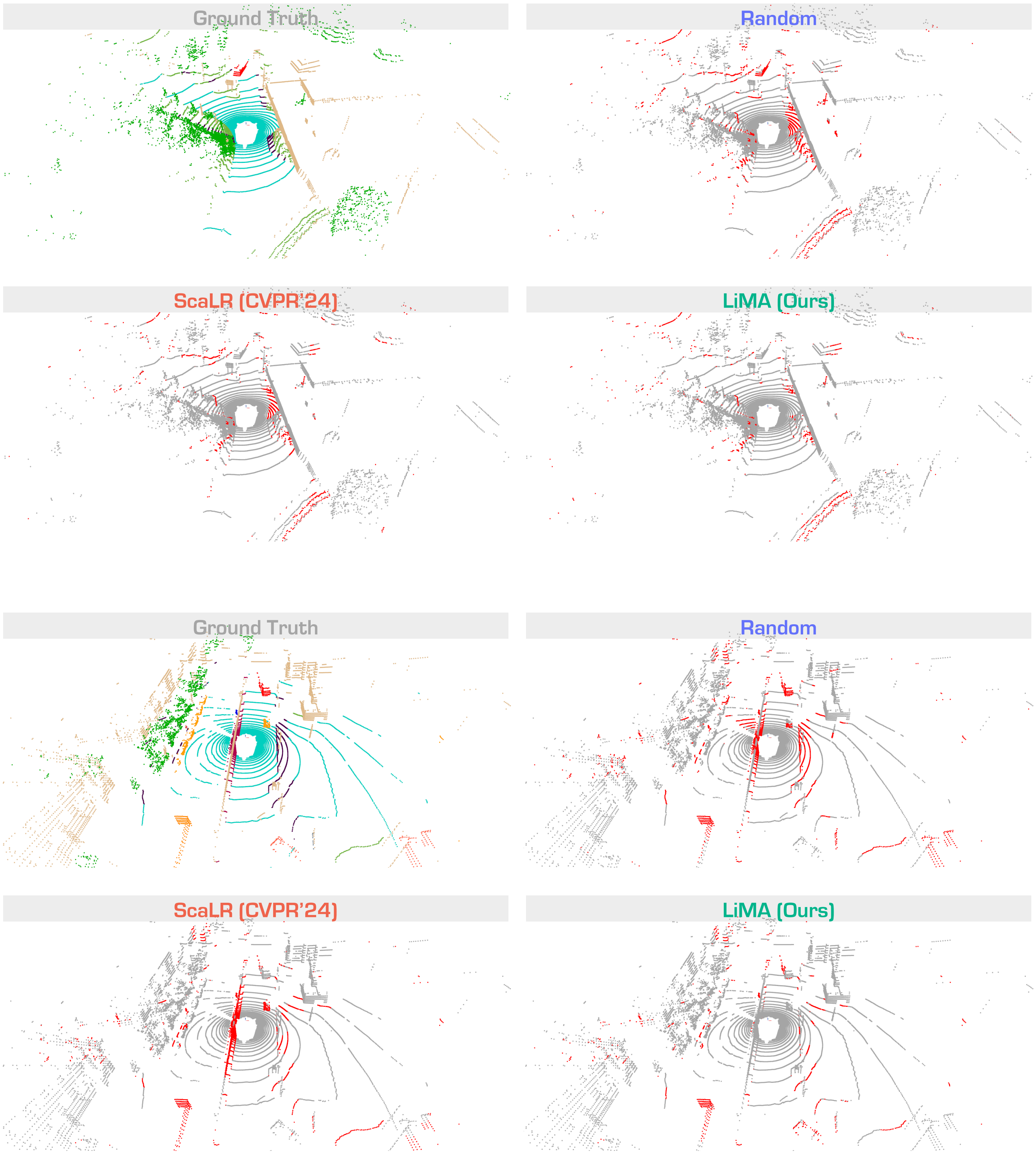

Figure A. **Qualitative assessments** of state-of-the-art methods, pretrained on *nuScenes* [3] and fine-tuned on *nuScenes* [5] with 1% annotations. The error maps depict **correct** and **incorrect** predictions in **gray** and **red**, respectively. Best viewed in colors.

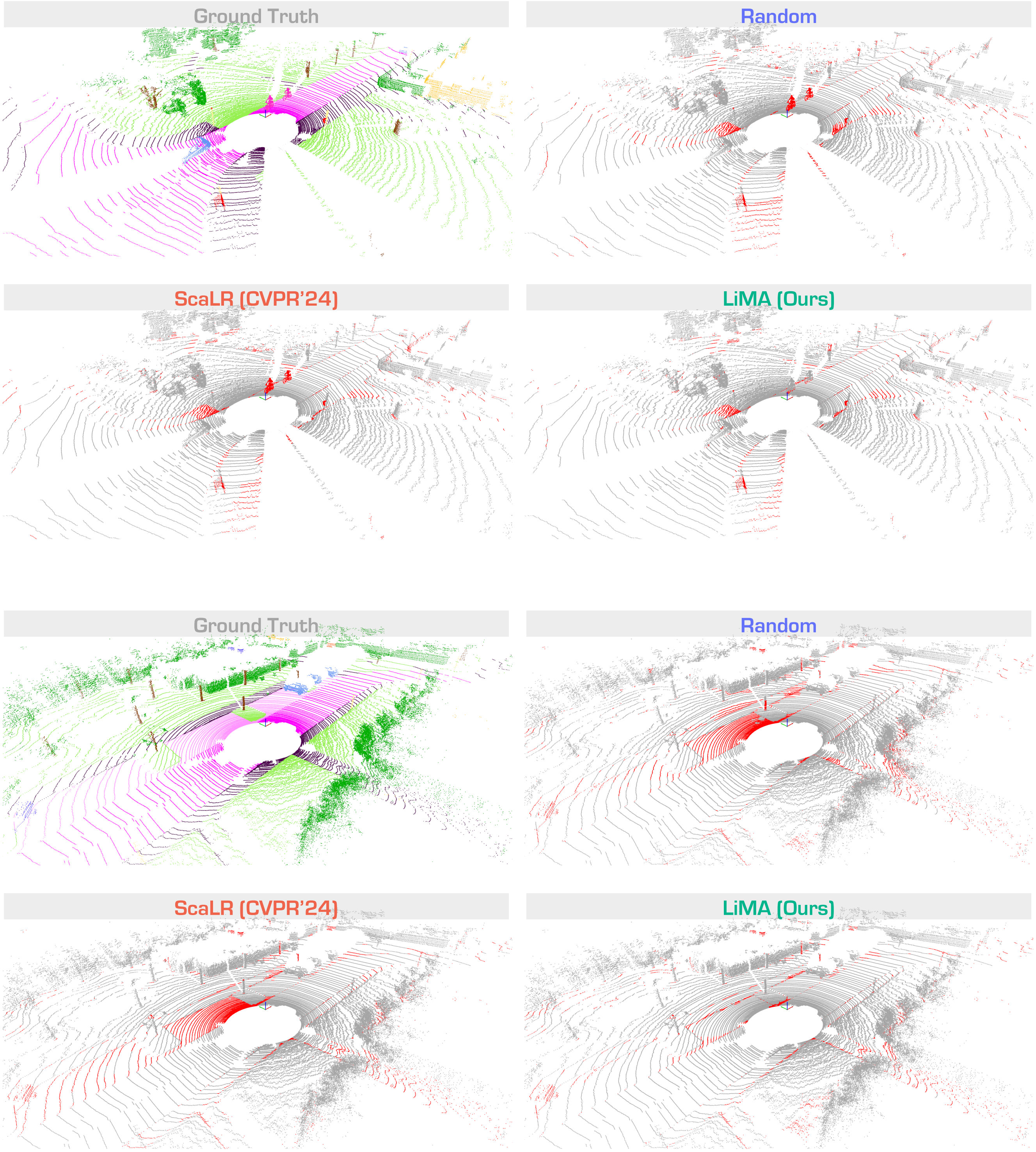

Figure B. **Qualitative assessments** of state-of-the-art methods, pretrained on *nuScenes* [3] and fine-tuned on *SemanticKITTI* [1] with 1% annotations. The error maps depict **correct** and **incorrect** predictions in **gray** and **red**, respectively. Best viewed in colors.

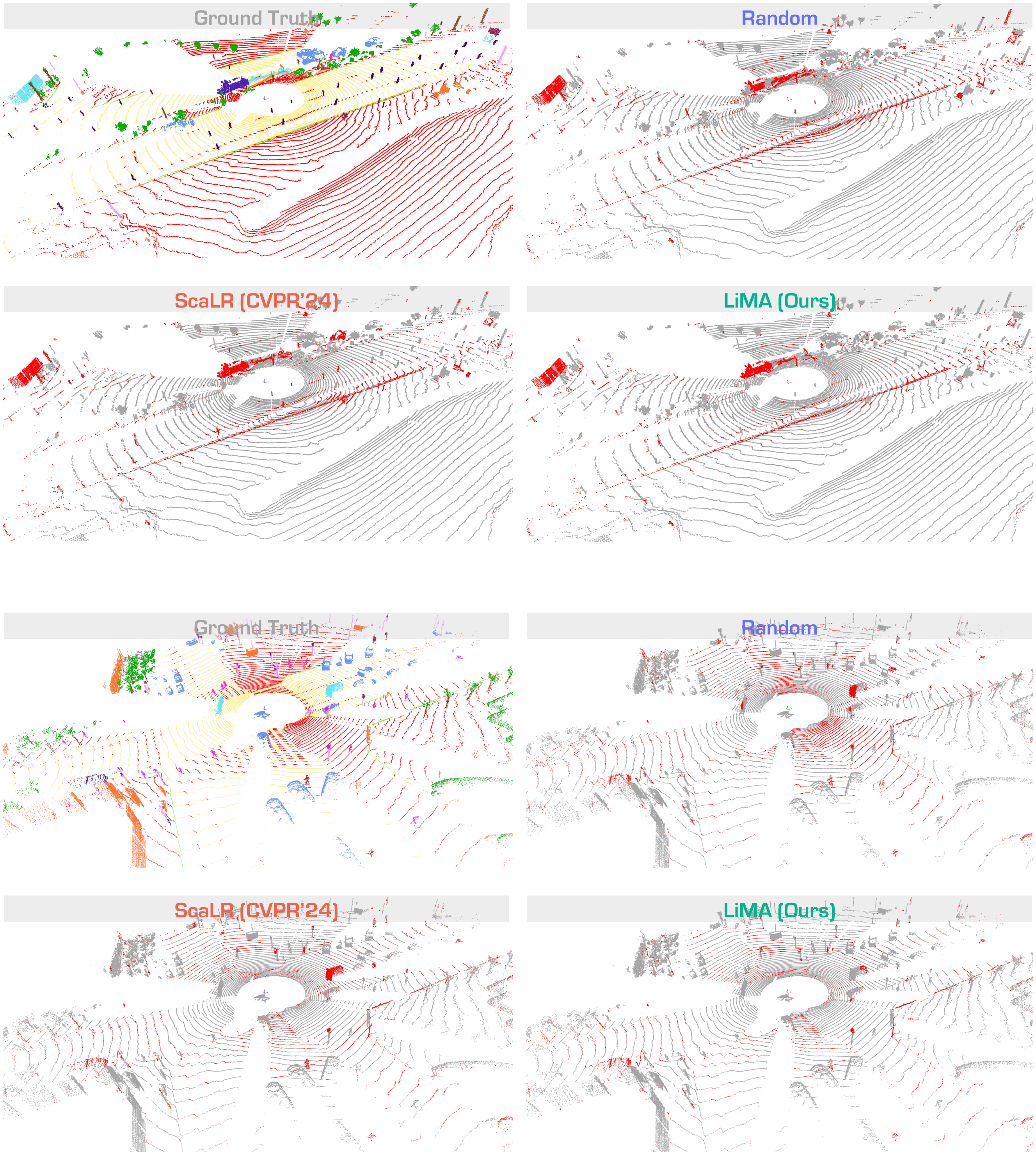

Figure C. **Qualitative assessments** of state-of-the-art methods, pretrained on *nuScenes* [3] and fine-tuned on *Waymo* [21] with 1% annotations. The error maps depict **correct** and **incorrect** predictions in **gray** and **red**, respectively. Best viewed in colors.

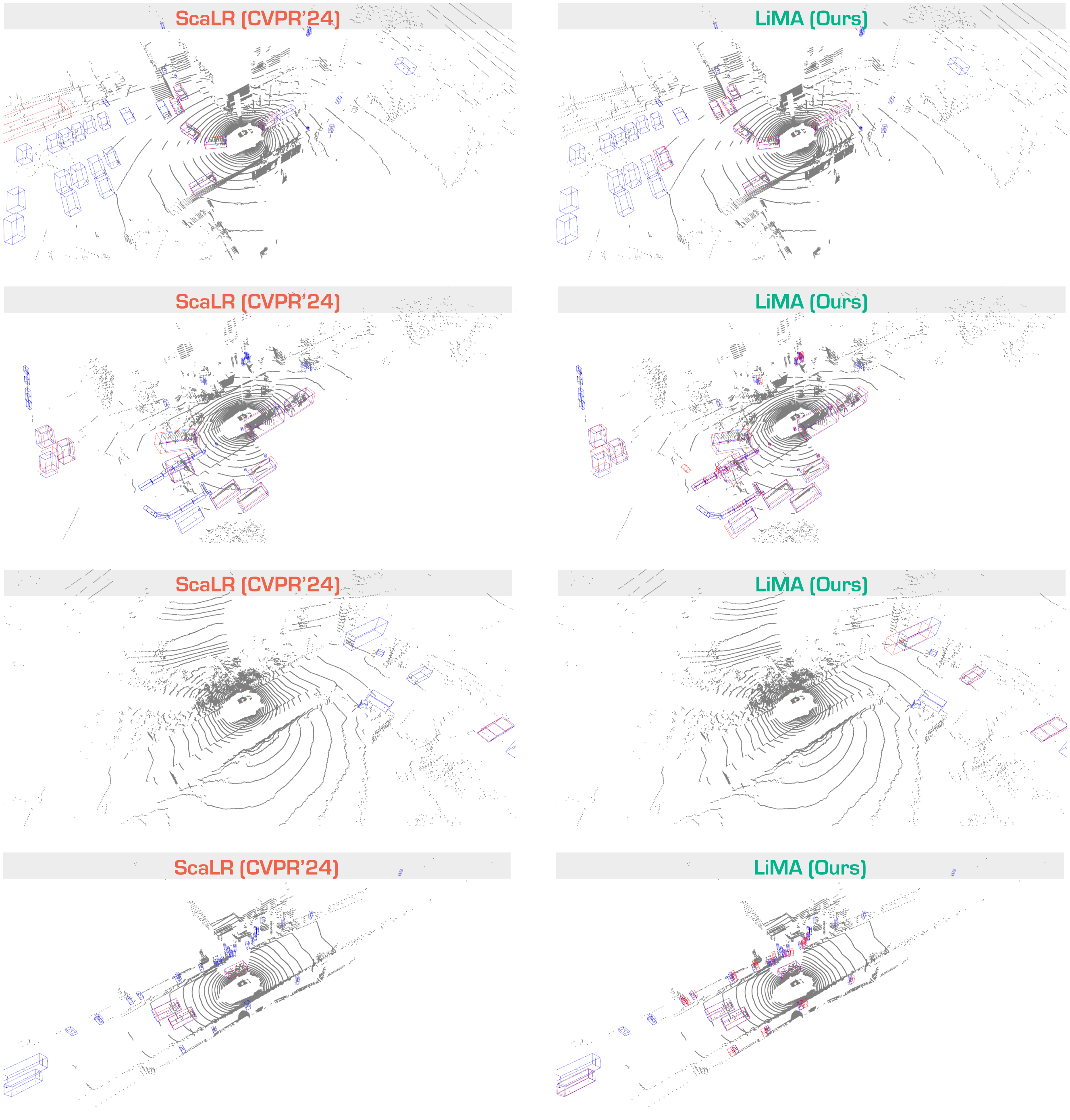

Figure D. **Qualitative assessments** of object detection, pretrained on *nuScenes* [3] and fine-tuned on *nuScenes* [3] with 5% annotations. The **groundtruth** / **predicted** results are highlighted with **blue** / **red** boxes, respectively. Best viewed in colors.

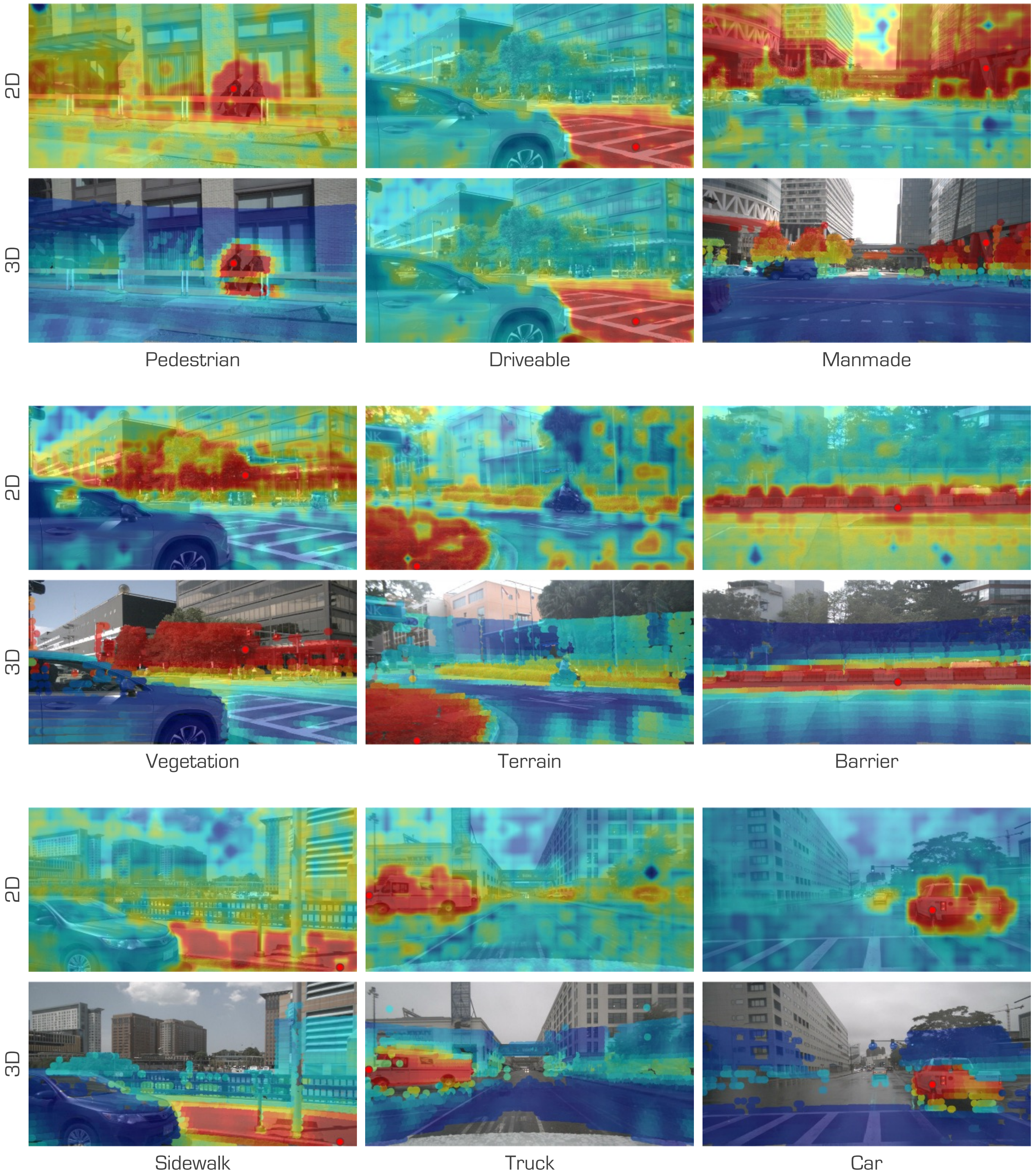

Figure E. **Cosine similarity** between a query point (marked as the red dot) and: (1) image features, and (2) LiDAR point features projected onto the image. Colors range from red (indicating high similarity) to blue (indicating low similarity). Best viewed in colors.

E. Public Resources Used

In this section, we acknowledge the use of the following public resources, during the course of this work.

E.1. Public Codebase Used

We acknowledge the use of the following public codebase, during the course of this work:

MMEngine[1]	Apache License 2.0
MMCV[2]	Apache License 2.0
MMDetection[3]	Apache License 2.0
MMDetection3D[4]	Apache License 2.0
OpenPCSeg[5]	Apache License 2.0

E.2. Public Datasets Used

We acknowledge the use of the following public datasets, during the course of this work:

nuScenes[6]	CC BY-NC-SA 4.0
SemanticKITTI[7]	CC BY-NC-SA 4.0
Waymo Open[8]	Waymo Dataset License
ScribbleKITTI[9]	Unknown
RELLIS-3D[10]	CC BY-NC-SA 3.0
SemanticPOSS[11]	CC BY-NC-SA 3.0
SemanticSTF[12]	CC BY-NC-SA 4.0
SynLiDAR[13]	MIT License
DAPS-3D[14]	MIT License
Synth4D[15]	GPL-3.0 License
Robo3D[16]	CC BY-NC-SA 4.0

E.3. Public Implementations Used

We acknowledge the use of the following implementations, during the course of this work:

nuscenes-devkit[17]	Apache License 2.0
semantic-kitti-api[18]	MIT License
waymo-open-dataset[19]	Apache License 2.0
semantic-poss-api[20]	MIT License
SLidR[21]	Apache License 2.0
DINOv2[22]	Apache License 2.0
Segment-Any-Point-Cloud[23]	CC BY-NC-SA 4.0
torchsparse[24]	MIT License
ScaLR[25]	Apache License 2.0
SuperFlow[26]	Apache License 2.0
FRNet[27]	Apache License 2.0

[1] https://github.com/open-mmlab/mmengine.

[2] https://github.com/open-mmlab/mmcv.

[3] https://github.com/open-mmlab/mmdetection.

[4] https://github.com/open-mmlab/mmdetection3d.

[5] https://github.com/PJLab-ADG/OpenPCSeg.

[6] https://www.nuscenes.org/nuscenes.

[7] http://semantic-kitti.org.

[8] https://waymo.com/open.

[9] https://github.com/ouenal/scribblekitti.

[10] https://github.com/unmannedlab/RELLIS-3D.

[11] http://www.poss.pku.edu.cn/semanticposs.html.

[12] https://github.com/xiaoaoran/SemanticSTF.

[13] https://github.com/xiaoaoran/SynLiDAR.

[14] https://github.com/subake/DAPS3D.

[15] https://github.com/saltoricristiano/gipso-sfouda.

[16] https://github.com/ldkong1205/Robo3D.

[17] https://github.com/nutonomy/nuscenes-devkit.

[18] https://github.com/PRBonn/semantic-kitti-api.

[19] https://github.com/waymo-research/waymo-open-dataset.

[20] https://github.com/Theia-4869/semantic-poss-api.

[21] https://github.com/valeoai/SLidR.

[22] https://github.com/facebookresearch/dinov2.

[23] https://github.com/youquanl/Segment-Any-Point-Cloud.

[24] https://github.com/mit-han-lab/torchsparse.

[25] https://github.com/valeoai/ScaLR.

[26] https://github.com/Xiangxu-0103/SuperFlow

[27] https://github.com/Xiangxu-0103/FRNet

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 1, 4, 9

[2] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13455–13465, 2023. 6, 7

[3] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

[4] Haoming Chen, Zhizhong Zhang, Yanyun Qu, Ruixin Zhang, Xin Tan, and Yuan Xie. Building a strong pre-training baseline for universal 3d large-scale perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19925–19935, 2024. 6, 7

[5] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 1, 4, 6, 7, 8

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[7] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pages 1110–1116, 2021. 1

[8] Alexey A Klokov, Di Un Pak, Aleksandr Khorin, Dmitry A Yudin, Leon Kochiev, Vladimir D Luchinskiy, and Vitaly D Bezuglyj. Daps3d: Domain adaptive projective segmentation of 3d lidar point clouds. *IEEE Access*, 11:79341–79356, 2023. 2

[9] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 2, 3

[10] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, pages 37193–37229, 2023. 6, 7

[11] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021. 4, 6, 7

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[13] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023. 6, 7

[14] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020. 2

[15] Bo Pang, Hongchi Xia, and Cewu Lu. Unsupervised 3d point cloud representation learning by triangle constrained contrast for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5229–5239, 2023. 6, 7

[16] Gilles Puy, Spyros Gidaris, Alexandre Boulch, Oriane Siméoni, Corentin Sautier, Patrick Pérez, Andrei Bursuc, and Renaud Marlet. Three pillars improving vision foundation model distillation for lidar. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21519–21529, 2024. 2, 3, 4, 5, 6, 7

[17] Cristiano Saltori, Evgeny Krivosheev, Stéphane Lathuiliére, Nicu Sebe, Fabio Galasso, Giuseppe Fiameni, Elisa Ricci, and Fabio Poiesi. Gipso: Geometrically informed propagation for online adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 567–585, 2022. 2

[18] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022. 1, 6, 7

[19] Corentin Sautier, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Vincent Lepetit. Bevcontrast: Self-supervision in bev space for automotive lidar point clouds. In *International Conference on 3D Vision*, pages 559–568, 2024. 6, 7

[20] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, pages 369–386, 2019. 2

[21] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 4, 10

[22] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2707, 2022. 1

[23] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2795–2803, 2022. 2

[24] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. 3d semantic segmentation in

the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023. 2

[25] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591, 2020. 6, 7

[26] Xiang Xu, Lingdong Kong, Hui Shuai, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, and Qingshan Liu. 4d contrastive superflows are dense 3d representation learners. In *European Conference on Computer Vision*, pages 58–80, 2024. 6, 7

[27] Sha Zhang, Jiajun Deng, Lei Bai, Houqiang Li, Wanli Ouyang, and Yanyong Zhang. Hvdistill: Transferring knowledge from images to point clouds via unsupervised hybrid-view distillation. *International Journal of Computer Vision*, pages 1–15, 2024. 6, 7

[28] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 6, 7