D-Aug: Enhancing Data Augmentation for Dynamic LiDAR Scenes

Jiaxing Zhao, Peng Zheng and Rui Ma

***Abstract*—Creating large LiDAR datasets with pixel-level labeling poses significant challenges. While numerous data augmentation methods have been developed to reduce the reliance on manual labeling, these methods predominantly focus on static scenes and they overlook the importance of data augmentation for dynamic scenes, which is critical for autonomous driving. To address this issue, we propose D-Aug, a LiDAR data augmentation method tailored for augmenting dynamic scenes. D-Aug extracts objects and inserts them into dynamic scenes, considering the continuity of these objects across consecutive frames. For seamless insertion into dynamic scenes, we propose a reference-guided method that involves dynamic collision detection and rotation alignment. Additionally, we present a pixel-level road identification strategy to efficiently determine suitable insertion positions. We validated our method using the nuScenes dataset with various 3D detection and tracking methods. Comparative experiments demonstrate the superiority of D-Aug.**

***Index Terms*—Data augmentation, deep learning, LiDAR point cloud, dynamic scenes**

I. Introduction

DUE to its precise range-sensing ability for capturing 3D geometric information, LiDAR sensors have found widespread use in various applications, particularly in the field of Autonomous Driving (AD). Recently, the most promising approach for processing LiDAR data involves training deep neural networks in various downstream applications [1], [2], [3], [4], [5], [6]. However, this approach requires plenty of labeled data [7], especially in 3D object detection and tracking tasks [8], [9], [10], [11], [12], [13], [14], [15]. Unfortunately, the manual collection and labeling of such data are time-consuming and labor-intensive [16], impeding comprehensive analysis and understanding of LiDAR point clouds. To alleviate the burden of data labeling, data augmentation [17], [18] has emerged as a prevailing method. It aims to effectively reduce the need for data labeling by enriching the training set through transformations of existing data.

Some global data augmentation methods [19], [20], [21], [22], [23], [24], [25], [26] involve manipulating the entire LiDAR dataset on a global scale, such as random scaling, flipping, and rotation. Conversely, other methods [19], [22], [27], [28] focus on object-level augmentation. For instance, LiDAR-Aug [16] integrates objects rendered from Computer-Aided Design (CAD) models into the original LiDAR point clouds. Despite the variety of data augmentation methods, they predominantly focus on statically augmenting the current frame, overlooking the continuity of augmented objects across consecutive frames. Ensuring this continuity is crucial for object detection and tracking tasks, and addressing this aspect is essential to maintain the realism of augmented data.

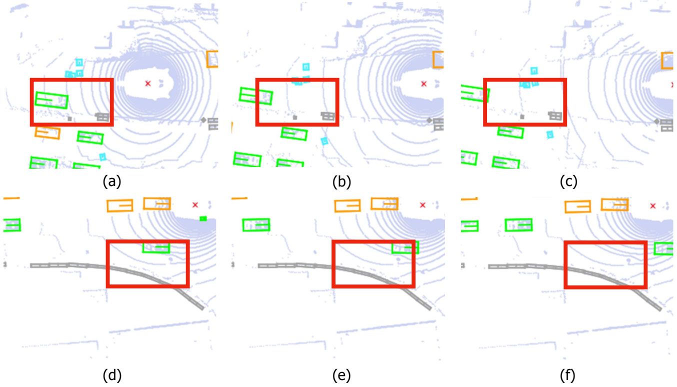

Fig. 1. Illustration of augmented LiDAR data. The three figures in the first row as well as those figures in the second row , display the augmented point clouds for three successive frames from two distinct scenes. The orange and green bounding boxes represent the original and inserted objects, respectively. Notably, the areas within the red rectangles emphasize the relative movement between the inserted objects and the stationary obstacles (grey boxes).

To address the aforementioned limitations, we introduce D-Aug, a novel LiDAR data augmentation method tailored for dynamic scenes. Unlike previous approaches, our method focuses on improving the continuity of objects across successive frames. We introduce a pixel-level road identification technique to locate available insertion positions within the scene, ensuring they align with the actual traffic flow. Additionally, we employ a dynamic collision detection algorithm to guarantee that inserted objects remain collision-free in dynamic scenes. In Figure 1, we illustrate augmented LiDAR data, where we use stationary obstacles as references to highlight how the augmented objects maintain dynamic continuity in successive frames.

The key contributions of our method are outlined as follows:

1) We propose D-Aug, a novel LiDAR data augmentation method tailored for dynamic scenes, ensuring the continuity of inserted objects across successive frames.
2) Our proposed method is evaluated on the publicly available nuScenes [29] dataset, demonstrating significant improvements in 3D object detection and tracking performance compared to various baselines.

Corresponding authors: Peng Zheng and Rui Ma.

Jiaxing Zhao, Peng Zheng, and Rui Ma are with the School of Artificial Intelligence, Jilin University, Changchun 130000, China (email: zhaojx9921@mails.jlu.edu.cn; zhengpeng22@mails.jlu.edu.cn; ruim@jlu.edu.cn)

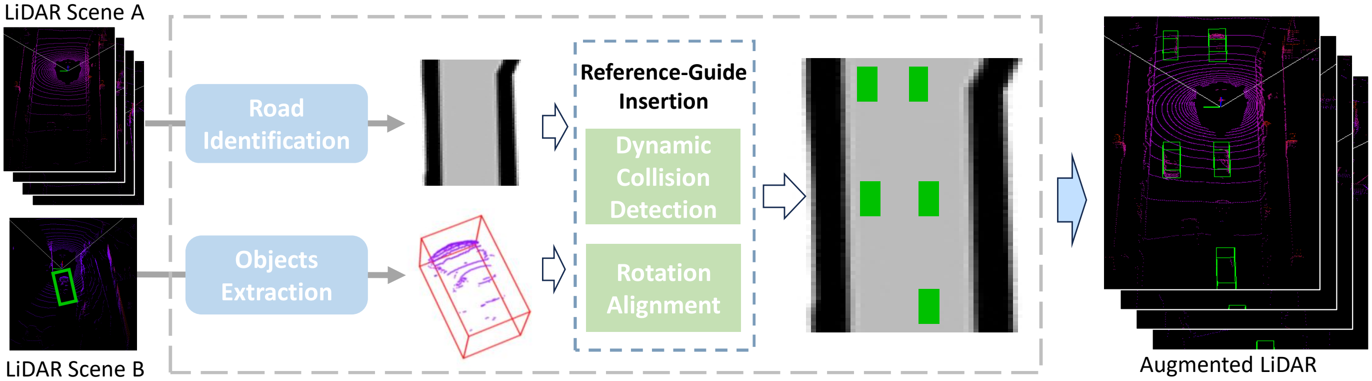

Fig. 2. Overview of D-Aug. Consecutive point cloud frames A are processed: available insertion positions are first determined through road identification. Subsequently, objects are extracted from point cloud B by calculating direction vectors. Finally, these extracted objects are inserted into each frame of A using a reference-guided insertion approach, which incorporates dynamic collision detection and rotation alignment. Notably, the inserted objects are rotated to align with the traffic flow in the dynamic scene, resulting in augmented dynamic scenes.

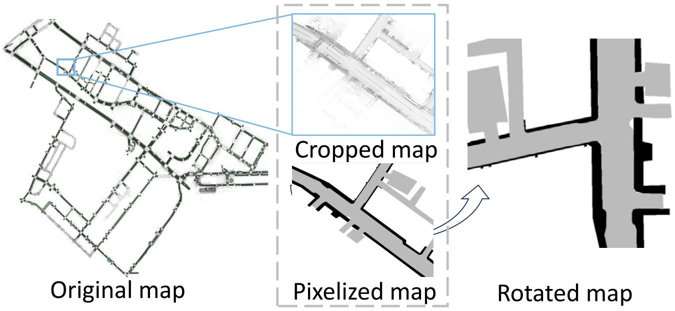

Fig. 3. Illustration of pixel-level road identification. The map is cropped, pixelized, and rotated to facilitate road identification based on pixel values. The grey areas represent the roads where the objects can be inserted.

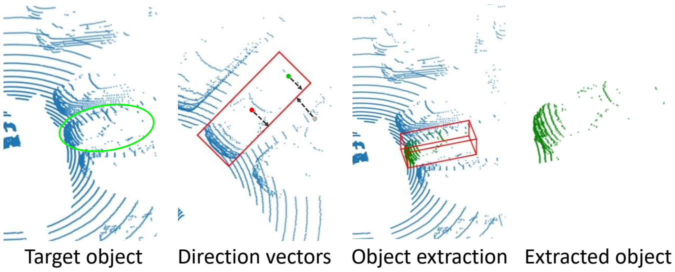

Fig. 4. Illustration of object extraction. Points within a bounding box are identified by calculating direction vectors. Points sharing the same direction vectors as the center point of the bounding box are extracted as objects.

II. Proposed Method

The following subsections present our approach: first, a pixel-level road identification method is introduced, which proves to be more efficient than the region-based segmentation provided by the nuScenesMap-API. Next, we describe the extraction of objects from a given point cloud. Finally, we discuss the insertion of extracted objects using reference-guided insertion. An overview of the proposed method is illustrated in Fig. 2.

A. Pixel-Level Road Identification

Before inserting objects, it is essential to find a suitable position to avoid overlap with existing objects in the scene. One possible solution is using nuScenesMap-API, which provides a fine-grained layer-based method to identify roads. However, it tends to be slow and inefficient when dealing with large-scale scenes. To address this issue, we introduce a pixel-level road identification method. Specifically, our method encompasses several key steps, including map cropping, pixelization, and road identification, as shown in Fig. 3.

Processing the entire scene is time-consuming, and LiDAR data far from the vehicle is often irrelevant for most applications. Hence, we crop the map for efficiency. Given LiDAR data, ego-pose, map, and the corresponding map mask, we pixelize the map and convert the position of the vehicle into pixel coordinates on the map. This pixelization is achieved by assigning distinct colors to different classes, where the class of each pixel is provided in the map mask. This approach enables us to determine the class for each pixel through a single query, rather than individual queries for each class in the map mask. Specifically, we assign ”grey” to ”road”. Subsequently, we crop the map with the vehicle’s pixel coordinates as the center. The cropped map is then rotated to align with the LiDAR data. Since the map is pixelized and our assignment of ”grey” to ”road”, roads can be directly identified based on the pixel values. During insertion, we validate the inserted position against the map. Specifically, we project the bounding box of inserted objects onto the map. The insertion position is deemed valid only if the projected bounding box does not intersect areas other than the road.

B. Objects Extraction

For effective point cloud insertion at the object level, precise object extraction is imperative. This process involves retrieving all point clouds associated with an object along with their corresponding bounding boxes. While the nuScenes [29] dataset provides bounding box information, extracting the point cloud is intricate due to the mismatch between LiDAR data recorded in the local coordinate system and the bounding boxes recorded in the global coordinate system. By utilizing rotation quaternions, translation vectors, and ego-poses from

TABLE I
EVALUATION RESULTS OF THE 3D DETECTION TASK ON THE NUSCENES VALIDATION DATASET

Method	mAP	NDS	Car	Tru	Bus	Tra	C.V.	Ped	Mot	Byc	T.C.	Bar
CBGS [30]	50.10	61.61	81.2	51.5	66.4	37.3	16.3	77.2	38.4	**17.7**	**57.3**	**57.6**
+ D-Aug	**50.68**	**61.77**	**82.1**	**52.8**	**67.5**	**39.1**	16.3	**77.7**	**40.2**	16.4	**57.3**	57.5
PointPillars [31]	43.30	**57.50**	80.6	48.0	62.5	33.7	10.7	70.9	30.0	**4.6**	44.0	48.0
+ D-Aug	**43.98**	57.46	**80.9**	**50.1**	**63.1**	**34.2**	**10.8**	**71.2**	**32.2**	3.9	**45.2**	**48.2**
CenterPoint [32]	59.10	66.69	85.3	57.3	71.5	**37.9**	17.1	85.0	58.9	41.4	**69.5**	67.1
+ D-Aug	**59.68**	**67.16**	**85.5**	**58.6**	**71.7**	**37.9**	**18.4**	**85.2**	**60.0**	**42.2**	68.9	**68.4**
VoxelNeXt [33]	60.45	66.72	**83.9**	57.1	70.5	38.5	19.3	**84.8**	**63.3**	49.2	69.8	**68.0**
+ D-Aug	**60.64**	**67.07**	**83.9**	**57.6**	**70.7**	**39.3**	**22.2**	84.7	62.2	**49.8**	**70.1**	65.9
TransFusion-L [34]	63.80	68.88	86.4	52.2	72.6	44.0	**26.6**	86.5	70.4	**57.0**	**73.9**	68.4
+ D-Aug	**64.44**	**69.11**	**86.8**	**54.8**	**75.0**	**45.9**	26.1	**86.7**	**71.4**	55.1	73.1	**69.6**

TABLE II
COMPARISONS OF 3D TRACKING TASK.

Method	AMOTA ↑	AMOTP ↓	MOTA ↑	Recall ↑
CenterPoint	65.4	**57.3**	56.2	69.0
+ D-Aug	**66.4**	57.4	**56.4**	**69.7**

the calibrated sensor, we transform the point cloud from the local to the global coordinate system, aligning it with the bounding box. To extract the point cloud within the bounding box, we utilize direction vectors from points to the bounding box faces, as illustrated in Fig. 4. A direction vector is defined as a vector starting from the given point and perpendicular to the specified face. Given a bounding box B, we compute direction vectors from its center point c to each face F_i. For each point p in the point cloud P, we calculate its direction vectors in the same manner. Points inside the bounding box share direction vectors with the center point.

C. Reference-Guided Objects Insertion

Randomly selecting insertion positions can lead to various issues, such as objects floating above the ground or misaligned orientations with the traffic flow, particularly in dynamic scenes. Intuitively, the position of existing objects can guide the insertion. Hence, we introduce a reference-guided insertion algorithm: Initially, an object is randomly chosen as a reference. Next, we search for available insertion positions within a set distance around the reference object. The availability of a position is determined by two factors: ensuring the inserted object remains grounded and avoiding collisions with other objects. If no suitable position is found, other objects are chosen as references for the search algorithm. Upon identifying a suitable insertion position, the extracted objects are first rotated to align with the traffic flow within the scene, facilitating smooth insertion. Finally, the objects are inserted into the target scene, ensuring consistency with its overall layout.

Collision detection in dynamic data augmentation poses more challenges than in static augmentation, as it must account for potential collisions not only in the current frame but also in subsequent frames of the target scene. By considering the velocity vector and position of bounding boxes in the current frame, collision relationships can be computed for each frame. The search for insertion positions continues until no collisions are detected, at which point the searched position is considered an available insertion position.

In a frame S_i from the set $\mathcal{S} = \{S_0, S_1, \ldots, S_K\}$ containing a set of bounding boxes $\mathcal{B}i = \{B_i^1, B_i^2, \ldots, B_i^n\}$, the collision detection function $\Pi(\mathcal{S}, B, \mathbf{V})$ is defined as follows, where B represents the bounding box of an inserted object:

$$\Pi(\mathcal{S}, B, \mathbf{V}) = \begin{cases} 1, & \text{if } \exists i \in \{0, \ldots, x\}{:}\exists B' \in \mathcal{B}_i{:} \\ & \quad \kappa(\text{Move}(B, i \cdot \mathbf{V}), B') = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, $\text{Move}(B, i \cdot \mathbf{V})$ represents moving the bounding box B along the velocity vector $i \cdot \mathbf{V}$, and $\kappa(B_1, B_2)$ indicates whether bounding boxes B_1 and B_2 collide. Specifically, two bounding boxes are considered to collide if their projections on the XY plane intersect.

III. EXPERIMENTS

A. Experimental Settings

1) Dataset.: The nuScenes [29] dataset comprises 1,000 driving sequences, with 700, 150, and 150 sequences allocated for training, validation, and testing, respectively. Each sequence spans approximately 20 seconds at 20 frames per second (FPS). The dataset provides calibrated vehicle attitude information and bounding box annotations across 10 classes with long-tailed distributions.

2) Metrics.: For the 3D detection task, mean Average Precision (mAP) and nuScenes Detection Score (NDS) are employed as evaluation metrics. Unlike the conventional 3D Intersection over Union (IoU), mAP is computed based on the aerial view, representing the average of AP across various classes. NDS integrates mAP with additional metrics such as translation, scale, orientation, and velocity into a weighted average. For further details about NDS, please refer to [29]. In the 3D tracking task, MOTA measures the overall accuracy of multi-object tracking. AMOTA [35], which averages MOTA across various IoU thresholds, offers a more comprehensive representation of the tracking capabilities. Meanwhile, AMOTP provides a complementary measure by concentrating on tracking precision. Additionally, the recall metric is considered in 3D tracking experiments.

TABLE III
ABLATION STUDIES ON THE OBJECT INSERTION METHOD

Method	mAP ↑	NDS ↑	AMOTA ↑	AMOTP ↓
Random	59.39	66.94	65.9	57.2
Reference-guide	**59.68**	**67.16**	**66.4**	57.4
- road identification	59.63	66.89	65.8	**56.7**

TABLE IV
ABLATION STUDIES ON THE ROAD IDENTIFICATION

Method	Cumtime ↓	Percall ↓
Layer filtering	1112.04s	0.881s
Pixel-level	**39.79s**	**0.011s**

The 'Cumtime' indicates the total time spent within a data augmentation process, while 'Percall' indicates the average time spent per call.

TABLE V
ABLATION STUDIES ON THE VOXEL SIZE

Size	Method	mAP ↑	NDS ↑	AMOTA ↑	AMOTP ↓
0.075	CenterPoint	59.10	66.69	65.4	**57.3**
	+D-Aug	**59.68**	**67.16**	**66.4**	57.4
0.100	CenterPoint	55.15	64.16	61.1	64.4
	+D-Aug	**56.15**	**64.61**	**62.6**	**61.0**
0.200	CenterPoint	51.68	60.61	58.6	**65.4**
	+D-Aug	**52.16**	**61.05**	**59.2**	65.6

TABLE VI
ABLATION STUDIES ON THE QUANTITY OF INSERTED OBJECTS

Num	mAP ↑	NDS ↑	AMOTA ↑	AMOTP ↓
1	59.46	66.58	65.7	58.5
3	59.36	66.75	60.4	62.2
5	**59.68**	**67.16**	**66.4**	**57.4**
8	59.38	66.92	65.3	58.4

3) Baselines.: To showcase the effectiveness of our method, we apply our proposed D-Aug to several state-of-the-art (SOTA) methods for 3D object detection and tracking: CBGS [30], PointPillars [31], CenterPoint [32], VoxelNeXt [33], and TransFusion-L [34]. Unfortunately, some of these methods [30], [31], [33], [34] do not provide code for the 3D tracking task. Therefore, we exclusively conduct experiments related to the 3D tracking task using CenterPoint.

B. Comparisons

1) 3D Detection: The comparisons for the 3D detection task are presented in Table I. The results highlight performance improvements in most cases when D-Aug is employed. Notably, enhancements are observed across nearly all metrics for CenterPoint [32]. Significantly, enhancements are observed across all baseline methods, particularly in classes such as "Car", "Tru", "Bus", and "Tra", which share a common characteristic: they typically exhibit higher speeds than other classes. It is evident that dynamic objects derive greater benefits from D-Aug augmentation. Additionally, employing D-Aug with TransFusion-L [34] yields the best performance in both mAP and NDS.

2) 3D Tracking: To further illustrate the efficacy of D-Aug, we conduct comparative experiments on the 3D tracking task. As depicted in Table II, the results indicate performance enhancement, particularly in AMOTA, where D-Aug achieves a 1.0% increase over the original method. The results solidify the efficacy of D-Aug in the 3D tracking task, as it accounts for the continuity within dynamic scenes.

C. Ablation studies

To assess the efficacy of each proposed component, we perform ablation studies on both 3D object detection and tracking tasks. Our ablation studies are conducted exclusively on CenterPoint since it provides code for the 3D tracking task.

1) Object Insertion Method: In object insertion, we propose a reference-guided insertion algorithm instead of randomly choosing an insertion position within the scene. Additionally, pixel-level road identification is introduced to ensure the validity of the insertion position. We validate our proposed insertion method through ablation studies, as shown in Table III. We also conduct an efficiency comparison between pixel-level road identification and the layer filtering method offered in nuScenesMap-API, with the results shown in Table IV.

2) Voxel Size: To efficiently manage substantial volumes of point cloud data, conversion into voxels is essential, as they offer a discrete, grid-based representation. While smaller voxels preserve finer details, they demand increased computational resources. To underscore the versatility of D-Aug, we conduct experiments with varying voxel sizes, as depicted in Table V. Encouragingly, the results demonstrate consistent enhancements across different voxel sizes, affirming the generalizability of our approach.

3) Quantity of Inserted Objects: The quantity of inserted objects can significantly impact performance. Therefore, we vary the number of inserted objects from 1 to 8 and analyze the results. As depicted in Table VI, we find that 5 objects yield optimal performance. This finding suggests that an excessive number of inserted objects may introduce unrealistic elements, while too few objects might not fully exploit the benefits of data augmentation.

IV. CONCLUSION

This paper introduces D-Aug, a specialized LiDAR data augmentation method designed for dynamic scenes. D-Aug entails extracting objects from LiDAR data and seamlessly inserting them into dynamic scenes while preserving their coherence across consecutive frames. Validating insertion positions is ensured through precise pixel-level road identification and reference-guided insertion strategy. Our experiments validate the effectiveness of D-Aug in enhancing performance across nuScenes detection and tracking benchmarks. Nonetheless, occlusion within the point cloud remains a challenge, suggesting that addressing post-insertion occlusion represents a promising direction for future work.

References

[1] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, *Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution*, p. 685–702. Jan 2020.

[2] J. Tu, P. Wang, and F. Liu, "Pp-rcnn: Point-pillars feature set abstraction for 3d real-time object detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul 2021.

[3] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[5] C. Qi, L. Yi, H. Su, and L. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Cornell University - arXiv,Cornell University - arXiv*, Jun 2017.

[6] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, *SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation*, p. 1–19. Jan 2020.

[7] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 2702–2719, Oct 2020.

[8] T. Sadjadpour, J. Li, R. Ambrus, and J. Bohg, "Shasta: Modeling shape and spatio-temporal affinities for 3d multi-object tracking," Nov 2022.

[9] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. M. Alvarez, "Focalformer3d: focusing on hard instance for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8394–8405, 2023.

[10] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1090–1099, 2022.

[11] J. Liu, L. Bai, Y. Xia, T. Huang, B. Zhu, and Q.-L. Han, "Gnn-pmb: A simple but effective online 3d multi-object tracker without bells and whistles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1176–1189, 2022.

[12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 918–927, 2018.

[13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

[14] C. Zheng, X. Yan, H. Zhang, B. Wang, S. Cheng, S. Cui, and Z. Li, "Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8111–8120, 2022.

[15] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European Conference on Computer Vision*, pp. 680–696, Springer, 2022.

[16] J. Fang, X. Zuo, D. Zhou, S. Jin, S. Wang, and L. Zhang, "Lidar-aug: A general rendering-based augmentation framework for 3d object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[17] M. Hahner, D. Dai, A. Liniger, and L. Van Gool, "Quantifying data augmentation for lidar based 3d object detection," *arXiv preprint arXiv:2004.01643*, 2020.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations,International Conference on Learning Representations*, Jan 2015.

[19] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, p. 3337, Oct 2018.

[20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[21] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[22] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[23] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[24] M. Hahner, D. Dai, A. Liniger, and L. Gool, "Quantifying data augmentation for lidar based 3d object detection.," *Cornell University - arXiv,Cornell University - arXiv*, Apr 2020.

[25] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "St3d: Self-training for unsupervised domain adaptation on 3d object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *Cornell University - arXiv,Cornell University - arXiv*, Feb 2020.

[27] P. Šebek, Š. Pokornỳ, P. Vacek, and T. Svoboda, "Real3d-aug: Point cloud augmentation by placing real objects with occlusion handling for 3d detection and segmentation," *arXiv preprint arXiv:2206.07634*, 2022.

[28] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao, "Polarmix: A general data augmentation technique for lidar point clouds," Jul 2022.

[29] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[30] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[31] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

[32] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[33] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking,"

[34] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1090–1099, 2022.

[35] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10359–10366, IEEE, 2020.