2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

LidarAugment: Searching for Scalable 3D LiDAR Data Augmentations

Zhaoqi Leng[1*], Guowang Li[1], Chenxi Liu[1], Ekin Dogus Cubuk[2], Pei Sun[1], Tong He[1], Dragomir Anguelov[1] and Mingxing Tan[1]

***Abstract*— Data augmentations are important for training high-performance 3D object detectors that use point clouds. Despite recent efforts on designing new data augmentations, perhaps surprisingly, most current state-of-the-art 3D detectors only rely on a few simple data augmentations. In particular, different from 2D image data augmentations, 3D data augmentations need to account for different representations of input data and require being customized for different models, which introduces significant overhead. In this paper, we propose *LidarAugment*, a practical and effective data augmentation strategy for 3D object detection. Unlike previous methods, which require tuning all augmentation policies in an exponentially large search space, we propose to factorize and align the search space of each data augmentation, which cuts down the 20+ hyperparameters to 2, and significantly reduces the search complexity. We show LidarAugment can be easily adapted to different model architectures with different input representations by a simple 2D grid search, and consistently improve a range of detectors including both convolution-based UPillars/StarNet/RSN and transformer-based SWFormer. Furthermore, LidarAugment mitigates overfitting and enables 3D detectors to scale up to larger capacities. When combined with the latest 3D detectors, LidarAugment achieves a new state-of-the-art 74.8 mAPH L2 on the Waymo Open Dataset.**

I. Introduction

Data augmentations are widely used in training deep neural networks. In particular, for autonomous driving, many data augmentations are developed to improve data efficiency and model generalization. However, most recent 3D object detectors have only employed a limited set of basic data augmentation operations, such as rotation, flip and ground-truth sampling [1], [2], [3], [4], [5], [6], [7], in contrast to 2D image recognition and detection models that use more sophisticated data augmentations [8], [9], [10], [11], [12], [13]. This paper aims to explore the feasibility of using more advanced 3D data augmentations to improve modern 3D object detectors, particularly for high-capacity models.

One of the main challenges in adopting advanced 3D data augmentations is that these augmentations are sensitive to input representations and model capacity. Different input representations, such as range-image-based models and point-cloud-based models, require different types of data augmentation. High-capacity 3D detectors are more prone to overfitting and require stronger overall data augmentation compared to lite models with fewer parameters. Therefore, tailoring each augmentation for different models is necessary. However, the search space scales exponentially with respect to the number of hyperparameters, resulting in a significant search cost. While recent studies [15], [16] attempt to address these challenges by using efficient search algorithms, they often rely on a fixed search space and complex search algorithms, such as population-based search [17], to find a data augmentation strategy for a model. However, our studies reveal that the search spaces used in prior works are suboptimal, and without a systematic way to define a good search space, the potential of a model cannot be fully realized, despite using complex search algorithms.

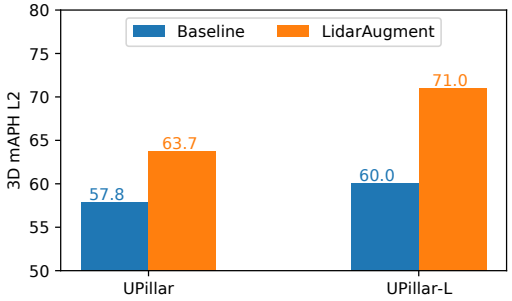


Fig. 1: **Model scaling with LidarAugment on WOD.** The baseline augmentations are adopyed from the prior art of [14]. When we scale up UPillars to UPillars-L, our LidarAugment improves both models, and the performance gains are more pronounced for the larger model, owing to its customizable regularization. More results in Table IV.

In this paper, we propose LidarAugment, a simplified search-based approach for 3D data augmentations. Unlike previous methods that rely on complex search algorithms to explore an exponentially large search space, our approach defines a simplified search space that contains a variety of data augmentations but has minimal (i.e. two) hyperparameters, making it easy to customize a diverse set of 3D data augmentations for different models.

Specifically, we construct the LidarAugment search space by first factorizing a large search space based on operations and exploring each sub search space with a per-operation search. Then, we normalize and align the sub search space for each data augmentation to form the LidarAugment search space, which contains only two shared hyperparameters: the normalized magnitude $m \in [0, \infty)$ and the probability of applying each data augmentation policies $p \in [0, 1]$. The LidarAugment search space significantly simplifies prior works [15] by reducing the number of hyperparameters to two, a 15$\times$ reduction in number of hyperparameters.

Despite only having two hyperparameters, the LidarAugment search space contains a variety of existing 3D data augmentations, such as drop/paste 3D bounding boxes, rotate/scale/dropping points, and copy-paste objects and back-

[1] Waymo Research, [2] Google Brain, * lengzhaoqi@waymo.com

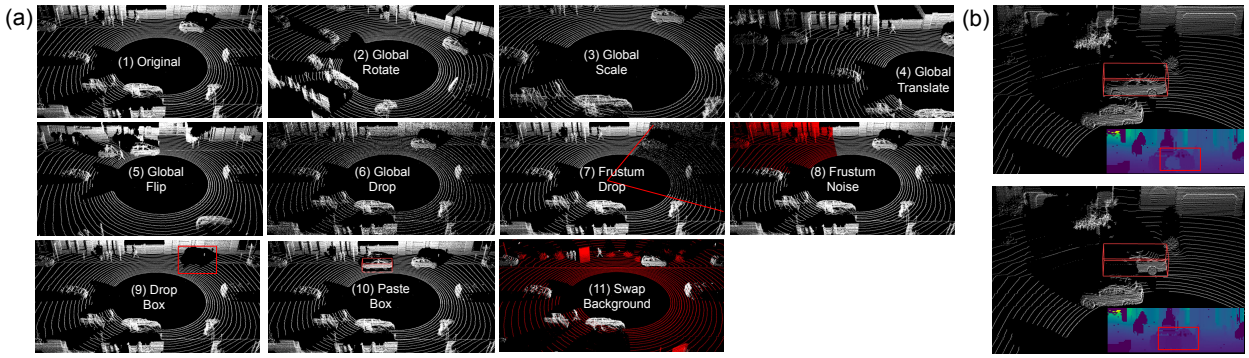


Fig. 2: **Visualizing LidarAugment.** (a) displays all data augmentation operations used in LidarAugment. For non-global operations, we highlight the augmented parts in red (boxes). (b) demonstrates how occlusion introduced by data augmentation, such as pasting a car object, is handled by removing overlapping rays in range view based on distance. We present point clouds and the corresponding range images with (bottom) and without (top) removing overlapping rays in the range view.

grounds. In addition, LidarAugment supports coherent augmentation across both point and range view representations, which generalizes to multi-view 3D detectors.

We perform extensive experiments on the Waymo Open Dataset [18] and demonstrate that LidarAugment is effective and generalizes well to different model architectures (convolutions-based and transformer-based), input views (3D point view and range image), and temporal scales (single and multi frames). Notably, LidarAugment advances state-of-the-art (SOTA) transformer-based SWFormer by 1.4 mAPH on the test set. Furthermore, LidarAugment provides customizable regularization, allowing us to scale up 3D object detectors to much higher capacity without overfitting. As summarized in Figure 1, LidarAugment consistently improves UPillars models, and the performance gains are particularly large for high-capacity models. Our contributions can be summarized as:

1) **New insight:** we reveal that common 3D data augmentation search spaces are suboptimal and should be tailored for different models.
2) **LidarAugment:** we propose LidarAugment that simplifies the search space for 3D data augmentations by utilizing only two hyperparameters while supporting 10 augmentation policies (15× reduction compares to prior works), offering diverse yet practical augmentations. In addition, we develop a new method to coherently augment both point and range-view input representations.
3) **State-of-the-art performance:** LidarAugment consistently improves both convolution-based UPillars/StarNet/RSN and attention-based SWFormer and demonstrates new state-of-the-art results on Waymo Open Dataset. In addition, LidarAugment enables model scaling to achieve much better quality for high-capacity 3D detectors.

II. Related works

Data augmentation. Data augmentation is widely used in training deep neural networks. For 3D object detection from point clouds, various global and local data augmentations, such as rotation, flip, pasting objects, and frustum noise, have been proposed to improve model performance [19], [1], [20], [2], [4], [21], [15], [22], [23], [24]. However, the effectiveness of 3D data augmentations are sensitive to model architectures and capacity, which often require extensive manual tuning. Therefore, most existing 3D object detectors [2], [6], [25], [26], [14] adopt only a few simple augmentations, such as flip and pixel shift [27], [28].

Several recent works attempt to use range images for multi-view 3D detection, but very few augmentations are developed for range images. [5] attempts to paste objects in the range image without handling occlusions. Our Paste Box augmentation support coherently augmenting both range-view and point-view input data while handling occluded objects in a simple way (more details in Figure 2), which enables more realistic augmented scenes and enriches the data augmentations for multi-view 3D detectors.

Learning data augmentation policies. Designing effective data augmentation typically requires manual tuning and domain expertise. Several search-based approaches have been developed to improve 2D images-based models, such as AutoAugment [9], RandAugment [12], and Fast AutoAugment [29]. Our LidarAugment is inspired by RandAugment in the sense that we aim to construct a simplified search space. However, unlike 2D image augmentations, where a search space works well for many models, we reveal that existing search space for 3D detection tasks are suboptimal, which motivates us to propose the first systematic method for defining search spaces for 3D detection tasks.

On the other hand, for 3D detection, PPBA [15] and PointAugment [16] propose efficient learning-based data augmentation frameworks for 3D point clouds. However, both approaches require users to run complex algorithms on an exponentially large but not well-designed search space. In contrast, our LidarAugment framework provides a systematic approach to design a simple yet more effective search spaces with only two hyperparameters.

III. LidarAugment

In this section, we first introduce data augmentation policies used in LidarAugment. Next, we analyze the perfor-

mance of each data augmentation policy on Waymo Open Dataset [18]. Finally, we propose a systematic approach to progressively design 3D augmentation search space.

A. Data augmentations for point clouds and range images.

3D point cloud and 2D range image are two different representations of LiDAR data. Despite being the native representation of LiDAR data, data augmentations for range image is not well studied in recent literatures compared to point clouds. In this section, we first review data augmentations for point clouds and then introduce a new method for coherently applying data augmentation to both point clouds and range images.

Augmenting point clouds. We follow the implementation of data augmentation policies described in recent studies [1], [15], [30], which contain global operations (rotate, scale, translate, flip, and drop points) and local operations (drop boxes, paste boxes, swap background, drop points and add feature noise in a frustum), as shown in Figure 2 (a).

Augmenting range images. Different from sparse 3D point representation, pixels in range image are compact. Data augmentations, such as pasting objects and swap background, disturb the compact structure of range representation. To coherently augment both 3D point view and 2D range view, we propose a novel approach that leveraging the bijective property between point clouds and range images to augment both, while accounting for occlusion.

First, we transform the range image pixels to point cloud based on their (x, y, z) coordinates. To preserve the bijective mapping between a pixel in a range image and a point in the corresponding point clouds, we concatenate the (row, column) index of each pixel in the range image as additional features before scattering pixels to 3D. After performing data augmentation in the point representation, we transform the augmented point clouds back to the range view by scattering each point to a pixel in a 2D image based on its (row, column) index.

Leveraging the compactness of range images. Coherently augmenting both range and point views leads to more realistic augmented scenes. Since each pixel in a range image corresponds to a unique ray from the LiDAR, overlapping pixels in the range view represent that the same light ray penetrates trough multiple surfaces. In such cases, we compare the distance between overlapping pixels in the range view and keep the pixel closest to the ego vehicle. This effectively removes occluded points in both the range and point views, as shown in Figure 2 (b).

B. Effects of each data augmentation.

In this section, we evaluate the effects of each data augmentation policy on Waymo Open Dataset [18]. To benchmark the policies, we propose a UPillars architecture that builds upon the popular PointPillars [2] and incorporates recent advances in architecture design, such as U-Net backbone [31], and the CenterNet detection head [32].

Datasets and training. The Waymo Open Dataset [18] contains 798 and 202 training and validation sequences. For our experiments, we trained UPillars using batch size of 64, the Adam optimizer [33], and a cosine decay learning rate with a max learning rate of 3e-3 and a total of 80000 steps.

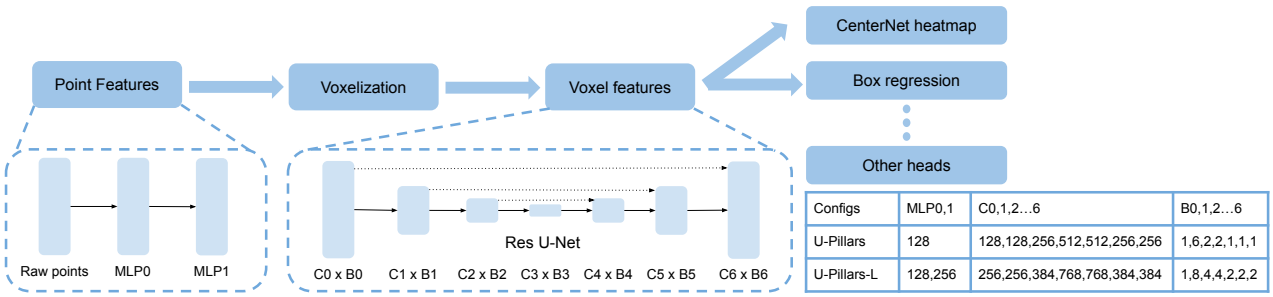
Policy	Hyperparameters	WOD (Veh./Ped.)	mAP L1
No Aug	-	-	60.2
Drop Box	Probability Number of boxes	p/p $2m/2.8m$	66.0 (+5.8)
Paste Box	Probability Number of boxes	$1.4p/p$ $3.2m/4.4m$	66.6 (+6.4)
Swap Background	Probability	$0.6p$	63.6 (+3.4)
Global Rot	Probability Max rotation angle	$1.4p$ $0.22\pi m$	73.3 (+13.1)
Global Scale	Probability Scaling factor	p $0.036m$	66.0 (+5.8)
Global Drop	Probability Drop ratio	p $1-0.18m$	64.9 (+4.7)
Frustum Drop	Probability Theta angle width Phi angle width R distance Drop ratio	p $0.1\pi m$ $0.1\pi m$ $75-7.5m$ $1-0.1m$	64.1 (+3.9)
Frustum Noise	Probability Theta angle width Phi angle width R distance Max noise level	$0.6p$ $0.14\pi m$ $0.14\pi m$ $75-10.5m$ $0.14m$	65.1 (+4.9)
Global Translate	Probability Stdev. of noise (x, y)	$1.4p$ $0.66m$	67.5 (+7.3)
Global Flip	Probability	p	69.0 (+8.8)

TABLE I: **Aligned search spaces and performance.** The search space for each hyperparameter on Waymo Open Dataset (WOD) for UPillars is listed. The global hyperparameters (p, m) control all data augmentation policies. After aliging the search space, the optimal (p, m) for each data augmentation are $(0.5, 5)$. We clip the probability of each policy to [0, 1], the minimum R distance to 0, te maximum rotation angle to [0, π), the maximum flip probability to 0.5, and the ratio of dropped points to [0, 0.8]. The theta angle and phi angle are clipped to [0, π) and [0, 2π), respectively.

Effect of each data augmentations. We factorize the LidarAugment search space into per-policy sub search space and show the performance of UPillars when trained using only one data augmentation policy on the Waymo Open Dataset (WOD) in Table I. Interestingly, on WOD, we find that global rotation is the most effective data augmentation, whereas pasting ground truth bounding boxes is commonly regarded as the most effective data augmentation [1] on KITTI [34]. A closer look at the statistics of the two datasets reveals that KITTI LiDAR frames, on average, contain about five objects, whereas, WOD frames, on average, contain more than 50 objects. Thus, pasting ground truth objects has a larger impact on KITTI, due to the significantly lower object density, than on WOD. On the other hand, we find that a smaller global rotation angle of $\pi/4$ is commonly used when training on KITTI, but we observe that a much stronger rotation of π is preferred for WOD.

C. Defining LidarAugment search space.

As noted in the previous section, different from RandAugment for 2D images [12], naively using the same search space across different datasets is suboptimal, which is a



Configs	MLP0,1	C0,1,2…6	B0,1,2…6
U-Pillars	128	128,128,256,512,512,256,256	1,6,2,2,1,1,1
U-Pillars-L	128,256	256,256,384,768,768,384,384	1,8,4,4,2,2,2

Fig. 3: **UPillars architecture.** The input points are processed by two full connected layers with channel size (MLP0, MLP1) before voxelized into pillars. The bird-eye-view pillars are then processed by a Res U-Net, which consists of (Ci, Bi) channels and blocks at each resolution. The CenterNet detection head, box regression head, and other attributes regression heads are applied to the output of the U-Net. For both models, we use the same voxel size of 0.32m and range of 81.92m.

unique challenge for 3D detection tasks. To address this challenge, we propose a method to factorize the entire search space and align each data augmentation based on its optimal hyperparameters.

Align the search space. Using global hyperparameters to control all data augmentations requires normalizing the search domain of each hyperparameter. Without normalization, the same global magnitude could lead to an overly aggressive application of one data augmentation and an insufficient application of another. To align the search domain of each data augmentation policy, we train a UPillars model on a given dataset using only one data augmentation policy at a time. Based on the optimal hyperparameter values, we rescale the search domain of each hyperparameter in a data augmentation policy from $[0, arbitrary\_value]$ to $[0, optimal\_value]$, so that the optimal value for each hyperparameter corresponds to the same global magnitude or probability hyperparameter. Since each data augmentation policy has multiple parameters, we perform a small-scale 2D grid search to scale the probability and magnitudes of all hyperparameters in each sub search space, to reduce the cost of the process.

Here, we provide an example on aligning the search space for a specific data augmentation, Global Translate. We define the initial search domain for the probability of applying Global Translate to be $\{0.3, 0.5, 0.7, 0.9\}$, and the domain for the magnitude of translation noise to be $\{0.9, 1.5, 2.1, 2.7, 3.3, 3.9\}$. If the optimal values $(p_{\text{noise}}, m_{\text{noise}}) = (0.7, 3.3)$, we rescale the search domain to $(p_{\text{noise}}, m_{\text{noise}}) = (1.4p, 0.66m)$ so that when the global hyperparameters $(p, m) = (0.5, 5)$, the hyperparameters for Global Translate are optimal. We apply a similar normalization process to align the search space for each data augmentation policy. Details of all the hyperparameters are listed in Table I. The LidarAugment pseudocode is shown in Figure 4.

```
augmentations = [
    DropBox, PasteBox, SwapBackground, GlobalRot,
    GlobalScale, GlobalDrop, FrustumDrop ,
    FrustumNoise, GlobalTranslate, GlobalFlip]
def lidaraugment(m, p, input_frame):
    for aug in augmentations:
        aug.set_magnitude(m)
        aug.set_probability(p)
        input_frame = aug.transform(input_frame)
    return input_frame
```

Fig. 4: **Pseudo Python code for LidarAugment.**

IV. Experiments

In this section, present our experimental setup and results. First, we introduce the datasets and models used in our experiments. Next, we demonstrate that LidarAugment significantly improves the performance of both convolution-based and attention-based models. Then, we present our results on model scaling, followed by ablations studied on different models and datasets.

A. Experimental setup

We evaluate LidarAugment on Waymo Open Dataset [18] (WOD) where the main metric is mAPH L2, and also perform additional ablation studies on nuScene [35]. We evaluate LidarAugment on a variety of 3D object detectors, as well as different model sizes. To ensure fair comparison, we follow the original training settings for each model, and only replace the baseline augmentations with our proposed LidarAugment. For training UPillars models, we use Adam optimizer [33] with a cosine learning rate schedule, a max learning rate of 1e-3, a total of 16e4 steps, and a batch size of 64.

B. LidarAugment achieves new state-of-the-art results

In Table II, we compare the validation set results on Waymo Open Dataset. Our LidarAugment significantly improves both convolution-based and transformer-based models. Notably, by scaling up the basic UPillars, our LidarAugment achieves 71.0 mAPH of L2 on UPillars-L, which is 1.9 AP better than the previous best convolution-based 3D detector PVRCNN++ [36]. It is worth mentioning that the latest transformer-based SWFormer [14] already uses 4 strong data augmentation policies, i.e., rotation (probability 0.74, yaw angle uniformly sampled from $[-\pi, \pi]$), random flip (probability 0.5), randomly scaling the world (scaling factor uniformly sampled from $[0.95, 1.05]$, and randomly drop points (drop probability 0.05), where the rotation angle and flip probability are maxed out. Despite that, LidarAugment still outperforms SWFormer by 1.9 AP, establishing a new state-of-the-art result for single-modal models without ensemble or test time augmentation on Waymo Open Dataset.

Method	Type	mAPH L2	Vehicle AP/APH 3D L1	Vehicle AP/APH 3D L2	Pedestrian AP/APH 3D L1	Pedestrian AP/APH 3D L2
P.Pillars [2] †	conv	51.9	63.3/62.7	55.2/54.7	68.9/56.6	60.4/49.1
CenterPoint [25]	conv	67.1	76.6/76.1	68.9/68.4	79.0/73.4	71.0/65.8
RSN_3f [6]	conv	68.1	78.4/78.1	69.5/69.1	79.4/76.2	69.9/67.0
PVRCNN++ [36]	conv	69.1	79.3/78.8	70.6/70.2	81.8/76.3	73.2/68.0
UPillars-L†	conv	60.0	69.5/69.0	61.5/61.0	70.4/66.1	63.0/59.0
UPillars-L(+LA)	**conv**	**71.0**	**79.5/79.0**	**71.9/71.5**	**81.5/77.3**	**74.5/70.5**
SST_1f [26]	attn	63.4	74.2/73.8	65.5/65.1	78.7/69.6	70.0/61.7
SST_3f [26]	attn	69.5	77.0/76.6	68.5/68.1	82.4/78.0	75.1/70.9
SWFormer_3f [14]	attn	70.9	79.4/78.9	71.1/70.6	82.9/79.0	74.8/71.1
SWFormer_3f (+LA)	**attn**	**72.8**	**80.9/80.4**	**72.8/72.4**	**84.4/80.7**	**76.8/73.2**

TABLE II: **WOD validation-set results.** *LA* denotes our LidarAugment, *conv* denotes convolutional networks, and *attn* denotes attention-based transformer models. LidarAugment improves both types of models and achieves the best results among each category. † model is trained using augmentations shown in prior art [14].

Table III presents the test-set results for the latest models. Our LidarAugment outperforms all prior works by a large margin and improves the test-set L2 mAPH by 1.4 AP compared to the latest SWFormer.

Method	mAPH L2	Vehicle AP/APH 3D L1	Vehicle AP/APH 3D L2	Pedestrian AP/APH 3D L1	Pedestrian AP/APH 3D L2
P.Pillars [2] †	55.1	68.6/68.1	60.5/60.1	68.0/55.5	61.4/50.1
M3DETR [37]	61.0	77.7/77.1	70.5/70.0	68.2/58.5	60.6/52.0
CenterPoint [25]	69.1	80.2/79.7	72.2/71.8	78.3/72.1	72.2/66.4
RSN_3f [6]	69.7	80.7/80.3	71.9/71.6	78.9/75.6	70.7/67.8
PVRCNN++ [36]	71.2	81.6/81.2	73.9/73.5	80.4/75.0	74.1/69.0
SST_TS_3f [26]	72.9	81.0/80.6	73.1/72.7	83.1/79.4	76.7/73.1
SWFormer_3f [14]	73.4	82.9/82.5	75.0/74.7	82.1/78.1	75.9/72.1
SWFormer_3f(+LA)	**74.8**	**84.0/83.6**	**76.3/76.0**	**83.1/79.3**	**77.2/73.5**

TABLE III: **WOD test-set results.** LidarAugment (LA) significantly improves performance of SWFormer and sets a new state-of-the-art mAPH L2. † reimplemented in [6].

C. LidarAugment enables better model scaling

Scaling up model capacity is a common approach to achieve better performance, but large 3D object detectors often suffer from overfitting. Table IV shows scaling results, where UPillars-L is a larger model with more layers and channels than UPillars, detailed in Figure 3.

We evaluate the impact of scaling model size on the performance of UPillars with strong augmentations used in the latest SWFormer as Baseline (see subsection IV-B). As shown in Table IV, with Baseline augmentations, increasing the capacity of the model from UPillars to UPillars-L without LidarAugment does not significantly improve the performance. In fact, several metrics, such as Veh/Ped L1 AP, even become worse (e.g., 69.3/70.3 for UPillar-L vs 72.1/72.3 for UPillars). We observe the training loss of UPillars-L is much smaller compared to loss of UPillars, indicating severe overfitting.

On the other hand, with LidarAugment, UPillars-L achieves much better performance, especially on the most challenging metric, i.e., +7.3AP for 3D L2 mAPH as shown in Figure 1.

Surprisingly, despite the baseline UPillar (mAPH=57.8) is much worse than latest 3D detectors, the performance of the scaled UPillar-L (+ LidarAugment) is competitive with the latest SWFormers, i.e., their mAPH are 71.0 vs. 72.8. This suggests the potential for exploring higher performance 3D detectors by scaling up model capacity using LidarAugment.

	Veh/Ped AP L1 UPillars	Veh/Ped AP L1 UPillars-L	Veh/Ped APH L2 UPillars	Veh/Ped APH L2 UPillars-L
BaseAugment	72.1/72.3	69.3/70.3	63.5/52.1	61.0/59.0
LidarAugment	**77.1/77.5**	**79.5/81.6**	**68.5/58.9**	**71.5/70.5**

TABLE IV: **UPillars scaling results on WOD**.

D. LidarAugment supports different representations

In contrast to 2D image models, 3D detectors are more diverse and could use different input representations due to the additional dimensionality and sparsity of point cloud data. In addition to UPillars and SWFormer, which are both pillar-based architectures that take 3D sparse points as inputs, we further demonstrate that LidarAugment generalizes to other input representations. First, StarNet [38] is a point-based detector that directly processes raw points in 3D to detect objects. RSN, on the other hand, utilizes the multi-view property of point clouds and takes both range images and 3D sparse points as inputs. However, due to the lack of multi-view data augmentations in prior works, RSN only utilize two simple augmentations, i.e., random flip and rotation.

Model	Augmentation	Vehicle	Pedestrian
StarNet [38]	baseline	58.2	71.9
	+LidarAugment	**61.6**	**74.2**
RSN-1frame [6]	baseline	75.2	77.2
	+LidarAugment	**75.8**	**79.0**
RSN-3frame [6]	baseline	77.0	79.1
	+LidarAugment	**77.7**	**80.6**
SWFormer [14]	baseline	79.4	82.9
	+LidarAugment	**80.9**	**84.4**

TABLE V: **LidarAugment improves various models**. Startnet is a point-based detector. RSN is a range image and pillar-based detector. Results are WOD L1 AP.

LidarAugment is a general method that supports augmenting different views of point clouds, including range images, as explained in subsection III-A. Table V shows the performance of LidarAugment on point-based StarNet, range image based RSN, and transformer-based SWFormer. In general, our LidarAugment improves all types of 3D detectors, sometimes by a large margin.

E. Ablation studies: comparing to other approaches.

In this section, we show that LidarAugment outperforms other common data augmentation approaches on UPillars.

Manually tuned data augmentation. Due to the complexity of search space scales exponentially with respect to the number of parameters, commonly used data augmentation strategies often consist of few data augmentation operations. Here, we compare two sets of data augmentation strategies used in training high-performance 3D detectors. First, we adopt the random flip (probability 0.5) and rotation (probability 0.5, yaw angle uniformly sampled from $[-\pi/4, \pi/4]$) data augmentations used in training RSN [6]. Then, we benchmark a more advanced and stronger data augmentation strategy used in training SWFormer [14], which is detailed in subsection IV-B. Our results show that both data augmentation strategies significantly improved UPillars' performances, about +10 AP for Vehicle and Pedestrian 3D L1 AP, when compared to the no augmentation baseline, shown in Table VI. However, tuning the data augmentation hyperparameters is challenging, as searching 4 values for each hyperparameter would result in more than 1000 searches of 5 hyperparameters.

UPillars (AP Level 1)	Hparams	Vehicle	Pedestrian
No Augmentation	-	58.0	62.4
Rotate & Flip [6]	3	70.8	70.0
Rotate, Flip, Scale, Drop points [14]	5	**72.1**	72.3
PPBA [15]	29	71.6	**72.6**
LidarAugment	**2**	**77.1** (+5.0)	**77.5** (+4.9)

TABLE VI: **LidarAugment outperforms common data augmentation strategies.** UPillars L1 APs on Waymo Open Dataset *validation set* are reported. LidarAugment requires the least number of hyperparameters (Hparams) but achieves the best results compared to manually designed and automl-based data augmentations strategies.

AutoML-based data augmentation. To address the challenge of the exponentially large search space, population-based training has been proposed for online tuning of hyperparameters [17], [11], [15]. We follow the implementation of progressive-population based data augmentation (PPBA) [15] and use the same sets of data augmentation policies and search space. We set population size to 16, generation step to 4000, perturbation and exploration rate to 0.2. Our results, shown in Table VI, indicate that PPBA significant outperforms the no augmentation baseline. Despite PPBA introducing significantly more data augmentation policies, it performs on par with manually tuned data augmentations, which only contains four policies.

Upon inspection of the search domain of each hyperparameter, we find the search space of PPBA is suboptimal. For example, the maximum rotation angle for global rotation in the PPBA search space is $\pi/4$, a common value used for the KITTI dataset. However, $\pi/4$ is insufficient compared to the tailored max rotation angle π used in our LidarAugment. Surprisingly, a single well-tuned global rotation augmentation achieves L1 mAP 73.3, shown in Table I, outperforming PPBA with an L1 mAP 72.1 over vehicle and pedestrian tasks. Although PPBA algorithm is more efficient than grid search and contains diverse augmentation policies, the suboptimal search domain of the rotation angle restricts the performance of PPBA, highlighting the importance of tailoring the search space for 3D detection.

LidarAugment In contrast, LidarAugment mitigates both the curse of dimensionality and suboptimal search domain issues by aligning and scaling the magnitude and probability of each data augmentation policy. This significantly reduces the search complexity (only two hyperparameters) while allowing exploration of a larger hyperparameter space. As indicate in Table VI, LidarAugment significantly outperforms both manually designed and AutoML-based data augmentation strategies by about 5 AP for both vehicle and pedestrian detection tasks and only requires a simple grid search of two hyperparameters.

F. Generalize to nuScenes dataset

To further validate our method, we evaluate LidarAugment on a different dataset: nuScenes [35]. For simplicity, we use the same training settings as in the Waymo Open Dataset, but reduce the voxel size to 0.25 and the total training steps by half for faster training. We also use the same baseline augmentation as SWFormer and redefine the LidarAugment search space for nuScenes, as described in subsection III-C. Table VII shows LidarAugment is a general approach that outperforms the baseline augmentation by a large margin on nuScenes.

UPillars	mAP	NDS
Rotate, Flip, Scale, Drop points [14]	40.6	48.2
LidarAugment	**46.7** (+6.1)	**53.4** (+5.2)

TABLE VII: **nuScenes validation-set results**.

V. Conclusion

In this paper, we propose *LidarAugment*, a scalable and effective 3D augmentation approach for 3D object detection that significantly simplifies the search process for augmentations and outperforms existing methods by a large margin. LidarAugment enables exploring a large search space while reducing the search complexity to only two hyperparameters. Extensive studies show that LidarAugment generalizes to convolution and attention-based architectures, as well as point-based and range-based input representations. More importantly, LidarAugment opens up exciting new research opportunities, such as model scaling in 3D detection. With LidarAugment, we demonstrate new state-of-the-art 3D detection results on the challenging Waymo Open Dataset.

REFERENCES

[1] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[2] 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*, 2019, pp. 12 697–12 705.

[3] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[5] Z. Liang, M. Zhang, Z. Zhang, X. Zhao, and S. Pu, "Rangercnn: Towards fast and accurate 3d object detection with range image representation," *arXiv preprint arXiv:2009.00206*, 2020.

[6] P. Sun, W. Wang, Y. Chai, G. Elsayed, A. Bewley, X. Zhang, C. Sminchisescu, and D. Anguelov, "Rsn: Range sparse net for efficient, accurate lidar 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5725–5734.

[7] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, "Range conditioned dilated convolutions for scale invariant 3d object detection," in *Conference on Robot Learning*. PMLR, 2021, pp. 627–641.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[11] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2731–2741.

[12] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[13] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[14] P. Sun, M. Tan, W. Wang, C. Liu, F. Xia, Z. Leng, and D. Anguelov, "Swformer: Sparse window transformer for 3d object detection in point clouds," *European Conference on Computer Vision (ECCV)*, 2022.

[15] S. Cheng, Z. Leng, E. D. Cubuk, B. Zoph, C. Bai, J. Ngiam, Y. Song, B. Caine, V. Vasudevan, C. Li *et al.*, "Improving 3d object detection through progressive population based augmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 279–294.

[16] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Pointaugment: an auto-augmentation framework for point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6378–6387.

[17] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846*, 2017.

[18] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[19] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[20] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[21] Y. Chen, V. T. Hu, E. Gavves, T. Mensink, P. Mettes, P. Yang, and C. G. Snoek, "Pointmixup: Augmentation for point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 330–345.

[22] J. S. Hu and S. L. Waslander, "Pattern-aware data augmentation for lidar 3d object detection," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2703–2710.

[23] J. Choi, Y. Song, and N. Kwak, "Part-aware data augmentation for 3d object detection in point cloud," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3391–3397.

[24] M. Reuse, M. Simon, and B. Sick, "About the ambiguity of data augmentation for 3d object detection in autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 979–987.

[25] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[26] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang, "Embracing single stride 3d object detector with sparse transformer," *arXiv preprint arXiv:2112.06375*, 2021.

[27] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 677–12 686.

[28] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, "Range conditioned dilated convolutions for scale invariant 3d object detection," *arXiv preprint arXiv:2005.09927*, 2020.

[29] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] Z. Leng, S. Cheng, B. Caine, W. Wang, X. Zhang, S. Jonathon, M. Tan, and A. Dragomir, "Pseudoaugment: Learning to use unlabeled data for data augmentation in point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 279–294.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[32] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[35] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[36] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection," *arXiv preprint arXiv:2102.00463*, 2021.

[37] T. Guan, J. Wang, S. Lan, R. Chandra, Z. Wu, L. Davis, and D. Manocha, "M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 772–782.

[38] J. Ngiam, B. Caine, W. Han, B. Yang, Y. Chai, P. Sun, Y. Zhou, X. Yi, O. Alsharif, P. Nguyen *et al.*, "Starnet: Targeted computation for object detection in point clouds," *arXiv preprint arXiv:1908.11069*, 2019.