Label-Guided Auxiliary Training Improves 3D Object Detector

Yaomin Huang[1,*], Xinmei Liu[1,*], Yichen Zhu[2], Zhiyuan Xu[2], Chaomin Shen[1,✉], Zhengping Che[2], Guixu Zhang[1], Yaxin Peng[3], Feifei Feng[2], and Jian Tang[2,✉]

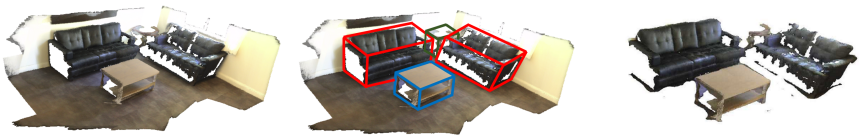
[1] School of Computer Science, East China Normal University
[2] AI Innovation Center, Midea Group
[3] Department of Mathematics, School of Science, Shanghai University
{51205901049,51205901078}@stu.ecnu.edu.cn
{cmshen,gxzhang}@cs.ecnu.edu.cn
{zhuyc25,xuzy70,chezp,feifei.feng,tangjian22}@midea.com
yaxin.peng@shu.edu.cn

Abstract. Detecting 3D objects from point clouds is a practical yet challenging task that has attracted increasing attention recently. In this paper, we propose a Label-Guided auxiliary training method for 3D object detection (LG3D), which serves as an auxiliary network to enhance the feature learning of existing 3D object detectors. Specifically, we propose two novel modules: a Label-Annotation-Inducer that maps annotations and point clouds in bounding boxes to task-specific representations and a Label-Knowledge-Mapper that assists the original features to obtain detection-critical representations. The proposed auxiliary network is discarded in inference and thus has no extra computational cost at test time. We conduct extensive experiments on both indoor and outdoor datasets to verify the effectiveness of our approach. For example, our proposed LG3D improves VoteNet by 2.5% and 3.1% mAP on the SUN RGB-D and ScanNetV2 datasets, respectively. The code is available at https://github.com/FabienCode/LG3D.

1 Introduction

3D object detection is one of the fundamental tasks toward precisely and adaptively understanding the real 3D world. Specifically, 3D object detection processes point clouds (as shown in Fig. 1a) to identify the types of objects and localize their bounding boxes (as shown in Fig. 1b). While challenging and computationally expensive, 3D object detection has attracted wide attention with an increasing amount of excellent works [1,2,3,13,14,16,17,18,25,26,28,35,36]. Existing 3D object detection methods mostly focus on improving the feature extraction in point clouds and making better predictions on objects' locations, such as fusing 2D image and 3D data information [16], leveraging a shape attention

[*] Equal contributions; work done during internships at Midea Group.
[✉] Corresponding authors.



(a) Point Cloud (b) Bounding Boxes (c) Labeled Point Cloud

Fig. 1: An example of (a) original point cloud, (b) bounding box annotations, and (c) the label point cloud extracted from the annotated bounding boxes.

graph convolution operator (SA-GConv) [1] to capture local shape features and relative geometric positions between points, and introducing a strong backbone for better feature learning ability [13]. Nevertheless, one of the most critical issues is that inference speed is typically sacrificed in order to maintain a high performance of 3D object detectors.

Balancing the inference speed and detection performance is challenging due to the nature of point clouds, i.e., the number of points in practical scenarios is huge, which slows down the forward pass. One can bypass such obstacles by applying aggressive sampling strategies, but it severely hurts the quality of detectors. Instead of modifying the architecture of the existing 3D object detectors, in this paper we resolve this issue by introducing a model-agnostic auxiliary training approach, which dramatically improves the detection performance and brings no extra computational cost at test time. Our proposed method is motivated by the assumption that the input labels (i.e., points within bounding boxes) contain rich semantic information if one could find a proper way to extract its latent features. These features can be considered an auxiliary information source, provide supervision to 3D object detectors during training, and, more importantly, can be removed after the training stage. As such, 3D object detectors can be optimized more effectively without hampering the inference speed.

The previous approach adopts learnable modules to extract features from labels in the 2D tasks. For instance, Mostajabi et al. [15] used an auto-encoder on the semantic masks to help the image segmentation model learn better pixel-level features. Similarly, LabelEnc [6] and LGD [34] formulate the bounding box along with its class identity as an extra source of information to supervise the student model. However, despite these previous attempts at learning label information, applying it to 3D detection is non-trivial due to the fundamental difference in input structure between 2D and 3D tasks, i.e., the image in the 2D task versus point clouds in 3D detection. Moreover, besides the categorical information, the point clouds inside the bounding box, i.e., the label point clouds shown in Fig. 1c, contain rich semantic and position information of each target object in the scene, which have been overlooked in the prior work.

Motivated by the above analysis, in this paper, we propose a Label-Guided auxiliary training approach for 3D object detection (LG3D), which serves as an auxiliary network to enhance the feature learning ability of vanilla 3D object de-

tectors. To better utilize the 3D label information, we introduce two novel modules in our method. First, the Label-Annotation-Inducer (LAI) module parameterizes the bounding box label and then maps them to task-specific representations. It aims to fuse the point clouds of particular objects into the sparse, original point clouds input such that the detectors can realize the object's localization, along with other critical but unexplored high-dimensional features, learned particularly by a tiny label encoder. The Label-Knowledge-Mapper (LKM) module is followed up to obtain optimal representations. Despite the simple design of our proposed modules, it tremendously improves the performance of 3D object detectors. It's also worth noting that our proposed LG3D is only used in the training stage and is completely cost-free during the inference.

We summarize our contributions as follows:

- We propose LG3D, a new way to utilize 3D labels by using the label point clouds (i.e., point clouds inside bounding boxes) as an auxiliary network to assist the feature representation learning of the vanilla network.
- Two novel modules, LAI and LKM, are used to fuse label point clouds, annotations, and original point clouds to a single feature embedding, which can effectively compensate for the missing information of target objects caused by data sampling.
- The proposed LG3D can be simply inserted into existing 3D object detectors and removed after training. LG3D improves the state-of-the-art 3D object detectors by a large margin on both indoor and outdoor datasets.

2 Related works

2.1 3D Object Detection

We briefly introduce the 3D object detection approaches in this section, and refer reader to Qian et al. [19] for more detailed description. ImVoteNet [16] proposes to use 2D image RGB, geometric coordinates, semantics, and pixel texture information to assist 3D point clouds object detection. PointPainting [21] proposes to use 2D semantic segmentation information to fuse the transformation matrix of LiDAR information and image information to the point. Cross-modal information fusion is proposed in the PointAugmenting [22] method, point features of corresponding points in 2D images are extracted by mapping between 3D and 2D. BRNet [3] proposes to solve the problem that VoteNet [17] cannot effectively represent the object structure information, adding a back-tracing module for resampling the more informative seed points. HGNet [1] describes the shape of an object by simulating the relative geometric position of the point. H3DNet [35] votes for center points on three dimensions of the bounding box, bounding box surface, and bounding box edges to add more detailed constraints to bounding box predictions. The backtracking module [3] is added based on VoteNet [17] to resample the seed points with richer information. 3DSSD [27] achieves a good balance between accuracy and efficiency by using the fusion sampling strategy in the downsampling process. In GroupFree3D [13], the transformer adaptively

determines the relationship between points and obtains an object proposal by point aggregation. DETR3D [24] uses DETR for 3D object detection, extracting 2D features from multiple camera images, then indexing these 2D features using a set of sparse 3D target queries, using a camera transform matrix to establish connections between 3D positions and multi-view images, and finally connecting 2D feature extraction and 3D box prediction by alternating between 2D and 3D calculations. 3DETR-m [14] improves detection performance by applying mask to self-attention in transformer. However, the calculation cost of object detection increases with the increment in the use of transformers.

Despite the evolutionary development of 3D object detectors, current approaches still require overwhelming computational costs at test time to maintain satisfactory performance. Thus, we provide a novel perspective to harness the semantic information in the label to assist the training of a 3D object detector, which is the first work demonstrating the powerful yet unexplored information in the point clouds that, if handled properly, can significantly boost the existing 3D detector. Our approach is also detector agnostic and robust to different kinds of datasets.

2.2 Auxiliary Task and Knowledge Distillation

Auxiliary Task Auxiliary task [33] is a well-studied topic that aims to assist the model with a lightweight module during training or testing. For example, in SA-SSD [8], the original point cloud features are complemented with down-sampled features with an auxiliary task. While auxiliary training in 3D detection has not raised attention, it has developed fast in 2D object detection. For instance, LabelEnc [6] proposes directly introducing auxiliary intermediate supervision to the trunk to provide feasible supervision in the training stage. It is further modified [34] into a teacher-free approach that incorporates bounding box and class information to the student network.

Knowledge Distillation Knowledge distillation (KD) is another highly closed topic in our approach. It was initially proposed to leverage a large teacher network that transfers its representative knowledge to a compact student network. Its success has spread over numerous domain in computer vision, i.e., image classification [9,37], object detection [32,10], semantic segmentation [12], and image-to-image translation [29,30].

For 3D object detection, SE-SSD [36] uses the idea of knowledge distillation to optimize student networks through a combination of hard and soft targets. Wang et al. [23] uses KD to compensate for the gap between the model of training high-quality input and the model of testing low-quality input in reasoning. Chong et al. [4] leverage point clouds to assist monocular 3D object detection with depth information. More recently, PointDistiller [31] leverages the dynamic graph convolution to transfer the local geometric structure of point clouds.

This work combines two advantages in the auxiliary task and knowledge distillation. Namely, our approach does not require a heavy, cumbersome teacher

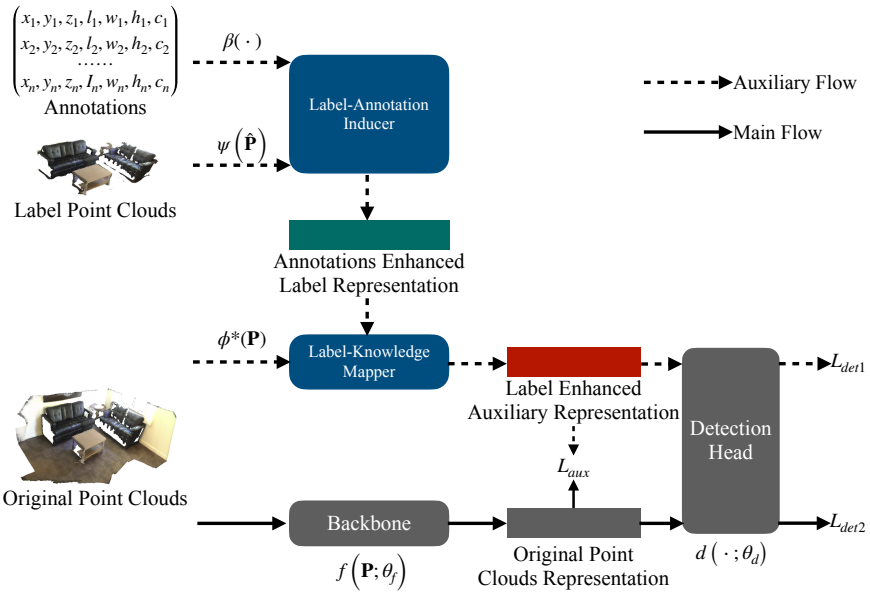


Fig. 2: Our LG3D framework. It includes an LKM module and an LAI module. The whole framework can be simply inserted into a 3D object detection network, and the LKM module shares the detection head with the backbone network. LG3D is removed directly in the inference phase, so it does not increase the computational cost. As shown in the figure, data flow in the training stage contains dotted and solid arrows, while data flow in the inference stage only contains solid arrows.

model to perform distillation. At the same time, we still enjoy the improvement in performance without extra computational cost at test time, which is normally unavoidable in training with the auxiliary task.

3 Method

In this section, we present our method in detail. Fig. 2 gives an overview of our method. In Sec 3.1, we introduce LKM supplements the original point clouds representation with label point clouds to obtain the label enhanced auxiliary representation. In Sec 3.2, LAI encodes the annotations and maps it to a latent semantic space for the annotation enhanced label representation that aim to get a better label enhanced auxiliary representation. In Sec 3.3, a separable auxiliary task uses the label enhanced auxiliary representation to supervise the representation outputs from the backbone with original point clouds.

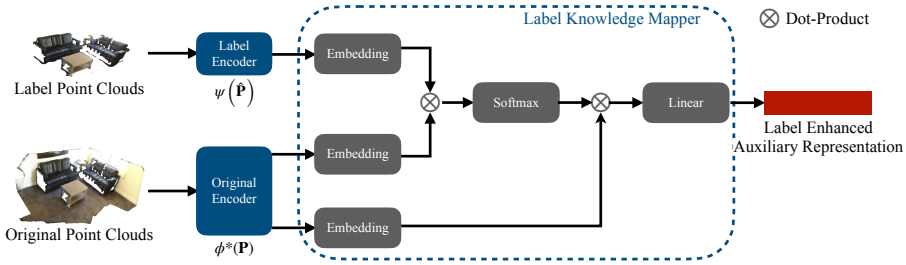


Fig. 3: The dashed box indicates the entire LKM module.

3.1 Label-Knowledge Mapper

The features of the original point clouds are usually extracted by the sampling method. This way, more or fewer point clouds of the object will be lost, affecting the feature extraction. Thus we design the LKM module to induce instance features from label point clouds, then fuse it with the original point clouds representation. This module can well supplement the key information lost during the point clouds samplings, especially for the information loss of small objects.

Original point clouds and label point clouds can be represented as disordered point set $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=1}^{n}$ with $\boldsymbol{p}_i \in \mathbb{R}^d$ and $\hat{\boldsymbol{P}} = \{\hat{\boldsymbol{p}}_j\}_{j=1}^{m}$ with $\hat{\boldsymbol{p}}_j \in \mathbb{R}^d$ respectively, where n and m is the number of original point clouds and label point clouds, respectively. d represents the (x, y, z) coordinate plus extra feature channels such as color, normal, etc. As shown in Fig. 3, in the training stage, we obtain the label point clouds by annotation information in the dataset and feed it into the label encoder. Given a training set $(\boldsymbol{P}, \hat{\boldsymbol{P}})$ and a well-trained original encoder function $\phi^*(\boldsymbol{P})$, instead of fine-tuning the original feature representation to the task-specific label space, we fix $\phi^*(\cdot)$ and learn a separate label encoder $\psi(\hat{\boldsymbol{P}})$ to extract the feature from label point clouds. Then we use a label fusion function $\mathcal{H}((\boldsymbol{P}, \hat{\boldsymbol{P}}), \theta_{\mathcal{H}})$ to fuse the original point clouds and the label point clouds. We find the optimal representation and function by

$$\theta_f^*, \theta_d^* = \arg\min_{\theta_f, \theta_d} \mathbb{L}_{\text{det}}^1(d(\mathcal{H}((\boldsymbol{P}, \hat{\boldsymbol{P}}), \theta_{\mathcal{H}}); \theta_d), y) + \mathbb{L}_{\text{det}}^2(d\left(f\left(\boldsymbol{P}; \theta_f\right); \theta_d\right), y) + \lambda \mathbb{L}_{\text{aux}}, \tag{1}$$

where $y \in \mathbb{R}^{N \times V}$ is the ground-truth label, N is the number of objects, and V is the label length of each objects. $f\left(\boldsymbol{P}; \theta_f\right)$ is the function realized by the backbone. $\mathcal{H}((\boldsymbol{P}, \hat{\boldsymbol{P}}), \theta_{\mathcal{H}}) \in \mathbb{R}^{n' \times C}$ represents the output of the LKM module, where n' is the number of sampled points and C is the number of feature channels. $\mathbb{L}_{\text{aux}}$ represents the auxiliary loss attached to the outputs of the backbone, which is independent of the detection head $d\left(\cdot, \theta_d\right)$ thus it is not affected by the latter's convergence progress. λ is the balanced coefficient.

The design of $\mathbb{L}_{\text{aux}}$ is one of the most important factor of our approach and we will explain its design in Sec 3.3. $\mathbb{L}_{\text{aux}}$ aims to minimize the distance between the

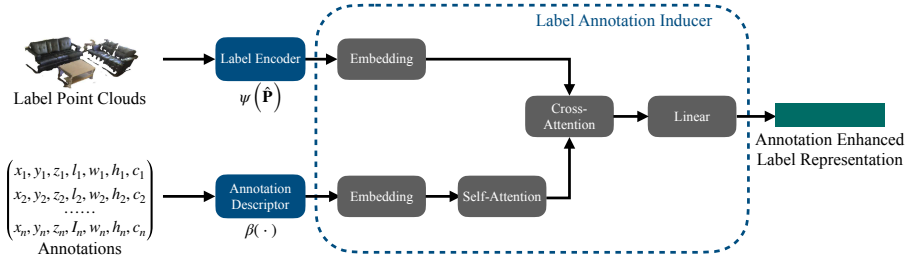


Fig. 4: In the training phase, label point clouds and annotations are fed into the LAI module.

original point clouds feature representation and an ideal representation, which in our method is the label enhanced auxiliary representation. In order to make sure that the original point clouds features can be well combined with the features of the label point clouds, we adopt the attention mechanism[20]. Let the matrix representations of the key ($\boldsymbol{K}_l$) and value ($\boldsymbol{V}_l$) be $\phi^*(\boldsymbol{P};\theta_p) \in \mathbb{R}^{M\times C}$ from the original point clouds representation, query be $\boldsymbol{Q}_l = \psi(\hat{\boldsymbol{P}};\theta_{\hat{P}}) \in \mathbb{R}^{M\times C}$, with the label point clouds representation. Here M and C denotes length and dimensions of query, key and value, respectively. The query, key and value are transformed by linear layers $f_{\mathcal{Q}_l}$, $f_{\mathcal{K}_l}$, $f_{\mathcal{V}_l}$ before conducting attention. To induce the feature from label point clouds, we apply the cross-attention mechanism [20] to fetch original point clouds representation from label point clouds representation. So the ideal representation output of LKM is:

$$\mathcal{H}\left((\boldsymbol{P},\hat{\boldsymbol{P}}),\theta_{\mathcal{H}}\right) = \text{Softmax}\left(\frac{f_{\mathcal{Q}_l}(\boldsymbol{Q}_l)f_{\mathcal{K}_l}(\boldsymbol{K}_l)^\top}{\sqrt{D_k}}\right) f_{\mathcal{V}_l}(\boldsymbol{V}_l), \tag{2}$$

where D_k denotes the dimensions of the key, and Softmax$(\cdot)$ is applied row-wise.

3.2 Label-Annotation-Inducer

Using the proposed Label-Knowledge Mapper described in Sec. 3.1, we obtain the enhanced point clouds representation $\mathcal{H}$. However, the rich information in the annotations has not been fully utilized. To use the label annotations information as an important form of ground truth, we propose the LAI module, as shown in Fig. 4, to complement the features of the label point clouds. Specifically, we extract the label annotations information to obtain an ideal representation $\mathcal{G}$.

Label Embedding In a 3D object detection task, the label information of an object usually contains the center, the size and category, and it may have the head angle. Given an object label, we represent each labeled bounding box of the target object in the point clouds as $\boldsymbol{\alpha}_i = (x_i, y_i, z_i, l_i, w_i, h_i, c_i)$, where i represents the i-th bounding box, (x_i, y_i, z_i) represents the center point of the

i-th bounding box, (l_i, w_i, h_i) represents the length, width and height of the i-th bounding box, and c_i represents the object category corresponding to the i-th bounding box. The initial label representation is

$$\mathcal{A} = \{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_i, \cdots, \boldsymbol{\alpha}_N\}, \quad \boldsymbol{\alpha}_i \in \mathbb{R}^{C_L}, \tag{3}$$

where i indicates the object index, C_L is the array length of the bounding box parameter information and N is the object number.

Annotation Augmentation We perform some information dropping for these determined annotations. When describing bounding boxes, we interpret them with rough scale indices. The centers of the approximate fields (x_i', y_i', z_i') are obtained by random dithering as

$$\begin{aligned} x_i' &= x_i + \eta_x l_i, \\ y_i' &= y_i + \eta_y w_i, \\ z_i' &= z_i + \eta_z h_i, \end{aligned} \tag{4}$$

where η_x, η_y, η_z are sampled from a uniform distribution $\eta \sim U[-B_\eta, B_\eta]$, where B_η is a scale factor whose value is set to be 0.1. Furthermore, we generate fake instances for the recognition task based on the dataset distribution. Note that we need to identify the objectiveness of each instance, that is, to determine the authenticity of a given label. We use the binary cross-entropy function to determine whether the given label is a real label in the point clouds scene or a pseudo label that we add manually and increase the robustness of learning knowledge from the real label by determining the virtual label:

$$\mathbb{L}_{\text{idf}} = -\frac{1}{N}\sum_{i=1}^{N} \delta_{\text{obj}}\left(\boldsymbol{\alpha}_i\right) \log\left(\mathcal{P}_{\text{obj}}\left(\boldsymbol{e}_i\right)\right) + \left(1 - \delta_{\text{obj}}\left(\boldsymbol{\alpha}_i\right)\right) \log\left(1 - \mathcal{P}_{\text{obj}}\left(\boldsymbol{e}_i\right)\right), \tag{5}$$

where $\mathcal{P}_{\text{obj}}(\cdot)$ is a prediction function with a full connection layer and sigmoid function, and $\delta_{\text{obj}}(\boldsymbol{\alpha}_i)$ encodes the binary classification labels, denoting whether the instance is randomly generated $(\delta_{\text{obj}}\left(\boldsymbol{\alpha}_i\right) = 0)$ or manually annotated $(\delta_{\text{obj}}\left(\boldsymbol{\alpha}_i\right) = 1)$.

We then introduce the annotations descriptor $\beta(\cdot)$, which could induce the new label representation $\mathcal{A}$ to the task-specific latent feature space. We adopt a relatively simple multi-layer perceptron [7] as the label annotation encoding module, so the optimization seems not difficult. The new label annotation representation is:

$$\beta_{\mathcal{A}} = \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_i, \cdots, \boldsymbol{e}_N\}, \quad \boldsymbol{e}_i \in \mathbb{R}^{C}, \tag{6}$$

where C is the intermediate feature dimension, and $\boldsymbol{e}_i = \beta(\boldsymbol{\alpha}_i)$ is the encoded label annotation information.

Label Information Interactions Since the annotation representation $\beta_{\mathcal{A}}$ is relatively independent, we first model it globally by a self-attention to obtain

the global annotation representation $\boldsymbol{Q}_\alpha \in \mathbb{R}^{N\times C}$. We then use this as a query condition to apply the cross attention mechanism to the label point clouds' features, so that the label point clouds can be combined with the annotation information to produce better annotation enhanced label representation.

Specifically, given the matrix representations of query $\boldsymbol{Q}_\alpha \in \mathbb{R}^{M\times C}$, key $\boldsymbol{K}_\alpha \in \mathbb{R}^{M\times C}$ and value $\boldsymbol{V}_\alpha \in \mathbb{R}^{M\times C}$ are from the label point clouds representation. Before conducting cross attention, the query, key, and value are transformed by linear layers $f_{\mathcal{Q}_\alpha}$, $f_{\mathcal{K}_\alpha}$, $f_{\mathcal{V}_\alpha}$,

$$\mathcal{F}_A\left(\boldsymbol{Q}_\alpha, \boldsymbol{K}_\alpha, \boldsymbol{V}_a\right) = \text{Softmax}\left(\frac{f_{\mathcal{Q}_a}(\boldsymbol{Q}_\alpha) f_{\mathcal{K}_\alpha}(\boldsymbol{K}_\alpha)^\top}{\sqrt{D_k}}\right) f_{\mathcal{V}_\alpha}(\boldsymbol{V}_\alpha). \tag{7}$$

With the LAI making label point clouds representation perceive label annotations information, the new label fusion function is:

$$\mathcal{G}\left(\boldsymbol{P}, (\hat{\boldsymbol{P}}, \mathcal{A}); \theta_{\mathcal{G}}\right) = \mathcal{G}\left(\phi^*\left(\boldsymbol{P}; \theta_P\right), \left(\psi(\hat{\boldsymbol{P}}; \theta_{\hat{P}}), \beta(\mathcal{A}; \theta_{\mathcal{A}})\right)\right). \tag{8}$$

3.3 Separable Auxiliary Tasks

In Sec. 3.1, we propose to use label point clouds to supplement the original point clouds representation. In Sec. 3.2, the label point clouds representation is further enriched by label annotations information. After these modules, we obtain an ideal representation $\mathcal{G}$ which can supervise the representation outputs from the backbone with original point clouds. We propose a separable auxiliary network using the above modules. It can be simply insert into various 3D object detection networks to improve the detection accuracy during the training.

It is clear that Eq. (1) directly corresponds to a multi-task training paradigm with three loss terms: the first one is label information encoder loss ($\mathbb{L}^1_{\text{det}}$) for the label information embedding; the second term is the common detection loss ($\mathbb{L}^2_{\text{det}}$), which enforces $d\left(\cdot; \theta'_d\right)$ to be a valid detection head; the third loss ($\mathbb{L}_{\text{aux}}$) minimizes the gap between the two latent spaces (namely the outputs of the backbone $f(\cdot; \theta'_f)$ and the label enhanced auxiliary representation $\mathcal{G}((\boldsymbol{P}, (\hat{\boldsymbol{P}}, \mathcal{A})), \theta_{\mathcal{G}})$.

By sharing the detection head for supervision, we ensure the instructive representation quality and consistency with the original point clouds representation. The overall detection loss is:

$$\mathbb{L}_{\text{det}} = \mathbb{L}^1_{\text{det}} + \mathbb{L}^2_{\text{det}} + \mathbb{L}_{idf}. \tag{9}$$

In addition to the common detection loss $\mathbb{L}_{\text{det}}$, we introduce an auxiliary supervision loss $\mathbb{L}_{\text{aux}}$ that uses outputs from LKM directly to supervise the detection backbone, as flow:

$$\mathbb{L}_{\text{aux}} = \min_{\theta_f} \left\| f\left(\boldsymbol{P}, \theta_f\right) - \mathcal{G}\left((\boldsymbol{P}, (\hat{\boldsymbol{P}}, \mathcal{A})), \theta_{\mathcal{G}}\right) \right\|_2, \tag{10}$$

where $\|\cdot\|_2$ is L2-distance to minimize the difference between original point clouds representation and label enhanced auxiliary representation. It is worth

noticing that the gradients of $\mathbb{L}_{\text{aux}}$ only update the backbone module. Above all, the overall loss with a coefficient λ can be summarised as follows:

$$\mathbb{L}_{\text{total}} = \mathbb{L}_{\text{det}} + \lambda \mathbb{L}_{\text{aux}}. \tag{11}$$

In summary, we use label-guided auxiliary training to motivate the underlying network to learn better feature representations. As Fig. 2 shows, during the testing phase, all dotted arrow flow lines are removed, so no additional computational overhead is incurred.

4 Experiments

4.1 Experiment Settings

Dataset To illustrate the generalization of our method, we have conducted experiments on indoor and outdoor datasets. For indoor datasets, the SUN RGB-D dataset consists of 10,355 single-view indoor RGB-D images annotated with over 64,000 3D bounding boxes and semantic labels for 37 categories. The ScanNetV2 dataset is a 3D mesh dataset with about 1,500 3D reconstructed indoor scenes with 40 semantic classes. We follow the commonly-used settings, selecting 10 classes of SUN RGB-D and 18 classes of ScanNetV2. For outdoor datasets, we choose the KITTI dataset for evaluation. The KITTI dataset contains 7481 training samples and 7518 test samples with three categories: Car, Pedestrian and Cyclist.

Data Preparation In the training stage, our network has two different inputs. On the one hand, we feed the full point clouds into the main branch to extract feature representation. On the other hand, we feed the point clouds inside in the bounding box and label information into the auxiliary network. The point clouds are randomly sub-sampled from the raw data of each dataset, i.e., 20,000 points from point clouds in the SUN RGB-D dataset and 40,000 point clouds from a 3D mesh in the ScanNetV2 dataset. Additionally, we perform data augmentation by randomly flipping, rotating, and scaling the point clouds.

Training and Evaluation We implement our LG3D using MMdetection3D [5] framework. For different networks with different datasets, we followed the basic settings in MMdetection3D without additional parameter tuning. The evaluation for indoor datasets follows the same protocol as [17] using mean average precision mAP@0.25 and mAP@0.50. We only evaluate our model on the class 'Car' for the KITTI dataset due to its large amount of data and complex scenarios, just as most state-of-the-art methods test their models. We follow the official KITTI evaluation protocol during the evaluation stage, and the IoU threshold is set to 0.7 for the class 'Car'.

Table 1: Results on indoor datasets.

Method	SUN RGB-D		ScanNetV2	
	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
VoteNet [17]	57.7	–	58.6	33.5
Reimpl. [5]	59.1	35.8	62.9	39.9
VoteNet+Ours	61.7	38.3	65.1	43.0
GroupFree3D [13]	63.0	45.2	69.1	52.8
GroupFree3D+Ours	**64.3**	**47.5**	**70.9**	**54.1**

Table 2: Results on the KITTI dataset.

Method	AP_3d(%)		
	Easy	Moderate	Hard
PointPillars	82.58	74.31	68.99
PointPillars+LG3D	**84.38**	**76.42**	**69.88**
3DSSD	88.36	79.57	74.55
3DSSD+LG3D	**88.96**	**81.47**	**76.72**

4.2 Main Results

Results on Indoor Datasets For indoor datasets, we evaluate our method on VoteNet [17] and GroupFree3D [13]. VoteNet is a classic and representative 3D object detection method, while GroupFree3D is the state-of-the-art method on indoor datasets. Results are presented in Table 1. The results show that our method significantly improve both frameworks. Compared with the baseline of VoteNet, our method achieves performance gains of 2.6% on the SUN RGB-D with mAP@0.25 and 2.5% with mAP@0.5. As for ScanNetV2, our model achieves performance gains of 2.2% and 3.1% on mAP@0.25 and mAP@0.5, respectively. Similarly, our method works for Group-Free 3D, which achieves performance gains of 1.8% mAP@0.25 and 1.3% mAP@0.5.

Results on the Outdoor KITTI Dataset To fully illustrate the generalization of our approach, we have added our module to 3DSSD [27] and PointPillars [11] and carried out experiments on the KITTI dataset. The comparison results on the KITTI test set are shown in Table 2. Compared with the baseline, our LG3D outperforms its original version. In terms of the main metric, i.e., AP on "moderate" instances, our method outperforms PointPillars and 3DSSD by 2.11% and 1.9%, respectively.

Table 3: Ablation study results of the LKM and LAI modules.

	SUN RGB-D		ScanNetV2	
	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
Baseline (Reimpl.)	59.1	35.8	62.9	39.9
Baseline+LKM	61.1	37.3	64.0	41.3
Baseline+LKM+LAI	**61.7**	**38.3**	**65.1**	**43.0**

Table 4: Ablation study results of the two-stage training strategy.

Method	mAP@0.25	mAP@0.5
Baseline	62.9	39.9
One-Stage	64.4	42.4
Two-Stage	**65.1**	**43.0**

4.3 Ablation Studies

In this section, we discuss the designed choices in LG3D and investigate their independent impact on final metrics in ablation studies. If not specified, all models are designed on VoteNet of ScanNetV2.

Label-Guided Module To better understand the role of our method, we conduct experiments to evaluate the contribution of each sub-task. Specifically, our LKM module comprises a label point clouds encoder and a supervision loss $L2$-distance. As shown in Table 3. Even if the LKM module alone is used to supplement the original representation with label point clouds, our method achieves some performance gains. When the LAI module is used to further complement the label point clouds representation, our method achieves a further performance improvement, which shows that our main modules significantly contribute to the overall network. For more ablation studies on label annotation augmentation strategies, please refer to the appendix.

Two Steps Training In our method, we use a two-step training strategy. First of all, we load the well-trained function $\phi^* \left(\boldsymbol{P}; \boldsymbol{\theta_P}\right)$, but do not freeze its parameters. We use a joint optimization method to optimize it together with $\psi(\hat{\boldsymbol{P}}; \theta_{\hat{\mathbb{P}}})$ and $\beta\left(\mathcal{A}; \theta_{\mathcal{A}}\right)$. By the first step, we obtain optimized $\phi^{*'}\left(\boldsymbol{P}; \theta_{\boldsymbol{P}}\right)$, $\psi^*(\hat{\boldsymbol{P}}; \theta_{\hat{\boldsymbol{P}}})$ and $\beta^*\left(\mathcal{A}; \theta_{\mathcal{A}}\right)$. In the second step, we load the functions obtained in the first step, freeze all parameters, and perform the second training step to obtain the final optimized $f\left(\cdot; \theta_f\right)$ and $d\left(\cdot; \theta_d\right)$. We show the ablation in Table 4.

Training with More Epochs The performance gain is from the proposed module, not the long training epochs. To verify that, we conduct additional

Table 5: Performance comparisons with different numbers of training epochs.

Dataset	Method	mAP@0.25	mAP@0.50	# of Epochs
ScanNetV2	VoteNet	62.90	39.90	36
		62.50	40.10	72
	VoteNet+LG3D	**65.10**	**43.00**	72
	GroupFree3D	69.10	52.80	80
		68.50	52.80	160
	GroupFree3D+LG3D	**70.90**	**54.10**	160
SUN RGB-D	VoteNet	59.10	35.80	80
		59.20	35.70	160
	VoteNet+LG3D	**61.70**	**38.30**	160

experiments to train both methods with equivalent epochs (72 on VoteNet and 160 on GroupFree3D). The results in Table 5 indicate that training baseline detectors for a long time are not helpful, which validates the effectiveness of our approach.

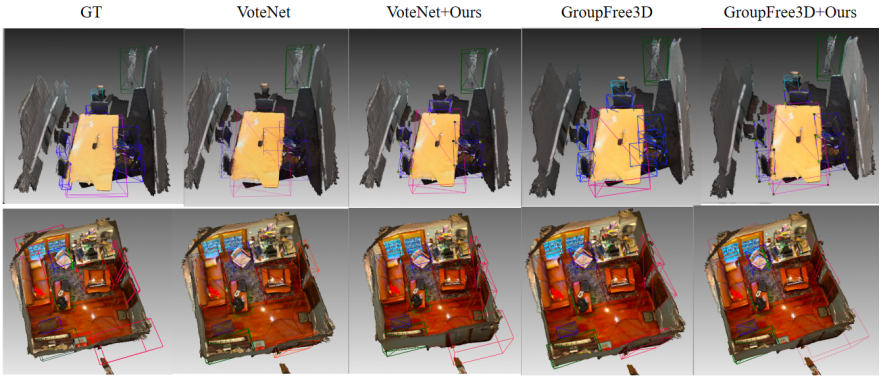


Fig. 5: Result visualizations on the ScanNetV2 dataset. GT means ground truth. The bounding box color denotes the object category.

4.4 Qualitative Results and Discussion

Fig. 5 shows several representative results on ScanNetV2. Our method has a good improvement effect on the detections of missing small objects and imprecision large objects. Fig. 6 shows that our method is particularly effective in detecting small objects and has an improvement for other objects. In addition, due to the

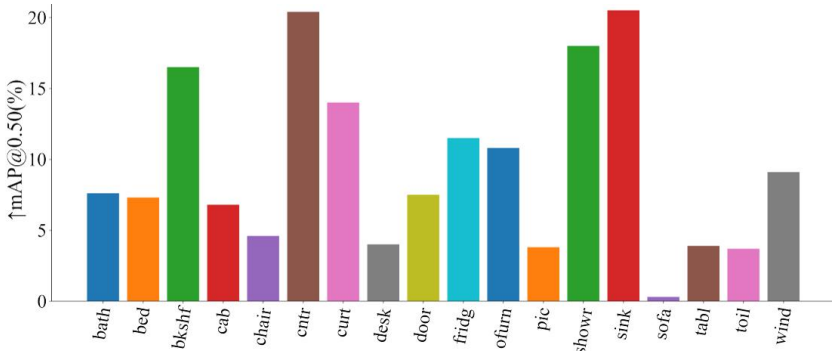


Fig. 6: The mAP@0.5 score improvement by LG3D applied to VoteNet of each category on the ScanNetV2 dataset.

limitation of the ball query radius during feature processing, the perceptual field is naturally limited, especially for objects with large aspect ratio differences, such as shower curtains and curtains. The initial VoteNet is affected by the backbone's natural limitation and the clustering operation in the detection head part by presetting the 3D spherical boundary. Our LAI module can effectively complement the effect of this problem by using the size information in the label annotations. The results show that the performance on shower curtains and curtains in the ScanNetV2 dataset are improved by about 15%. For more results on per-category performance comparisons, please refer to the appendix.

5 Conclusion

In this work, we have designed a novel label-guided auxiliary approach for 3D object detection networks to facilitate the training process. A novel point clouds label encoding module is introduced to map real labels into potential embeddings that serve as auxiliary intermediate supervision of the detection backbone during training. A knowledge refinement-like idea is used to simplify our auxiliary module. The processed label information is fed into the upper branch auxiliary network for encoding in the experiments. The distance information represented by the 3D features is used to directly optimize the feature embedding of the detection backbone. Experiments show that this method greatly improves the detection performance of the original network while maintaining the detection speed of the original network.

Acknowledgement

This work was done when Yaomin Huang and Xinmei Liu took internships at Midea Group. This work was supported in part by National Science Foundation of China (61731009 and 11771276) and Shanghai Pujiang Program (21PJ1420300).

References

1. Chen, J., Lei, B., Song, Q., Ying, H., Chen, D.Z., Wu, J.: A hierarchical graph network for 3D object detection on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 392–401 (2020)
2. Chen, Q., Sun, L., Wang, Z., Jia, K., Yuille, A.: Object as hotspots: An anchor-free 3D object detection approach via firing of hotspots. In: European Conference on Computer Vision. pp. 68–84. Springer (2020)
3. Cheng, B., Sheng, L., Shi, S., Yang, M., Xu, D.: Back-tracing representative points for voting-based 3D object detection in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8963–8972 (2021)
4. Chong, Z., Ma, X., Zhang, H., Yue, Y., Li, H., Wang, Z., Ouyang, W.: Monodistill: Learning spatial features for monocular 3d object detection. In: International Conference on Learning Representations (2022)
5. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d` (2020)
6. Hao, M., Liu, Y., Zhang, X., Sun, J.: Labelenc: A new intermediate supervision method for object detection. In: European Conference on Computer Vision. pp. 529–545. Springer (2020)
7. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
8. He, C., Zeng, H., Huang, J., Hua, X.S., Zhang, L.: Structure aware single-stage 3D object detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11873–11882 (2020)
9. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015)
10. Kang, Z., Zhang, P., Zhang, X., Sun, J., Zheng, N.: Instance-conditional knowledge distillation for object detection. Advances in Neural Information Processing Systems **34**, 16468–16480 (2021)
11. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
12. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2604–2613 (2019)
13. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3D object detection via transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2949–2958 (2021)
14. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3D object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
15. Mostajabi, M., Maire, M., Shakhnarovich, G.: Regularizing deep networks by modeling and predicting label structure. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5629–5638 (2018)
16. Qi, C.R., Chen, X., Litany, O., Guibas, L.J.: Imvotenet: Boosting 3D object detection in point clouds with image votes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4404–4413 (2020)

17. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep Hough voting for 3D object detection in point clouds. In: proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9277–9286 (2019)
18. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: 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)
19. Qian, R., Lai, X., Li, X.: 3d object detection for autonomous driving: a survey. Pattern Recognition p. 108796 (2022)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
21. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: PointPainting: Sequential fusion for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4604–4612 (2020)
22. Wang, C., Ma, C., Zhu, M., Yang, X.: PointAugmenting: Cross-modal augmentation for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11794–11803 (2021)
23. Wang, Y., Fathi, A., Wu, J., Funkhouser, T., Solomon, J.: Multi-frame to single-frame: Knowledge distillation for 3d object detection. In: The Workshop on Perception for Autonomous Driving at the European Conference on Computer Vision (2020)
24. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
25. Wang, Z., Zhao, Z., Jin, Z., Che, Z., Tang, J., Shen, C., Peng, Y.: Multi-stage fusion for multi-class 3d lidar detection. In: IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021. pp. 3113–3121 (2021)
26. Xu, Q., Zhou, Y., Wang, W., Qi, C.R., Anguelov, D.: SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15446–15456 (2021)
27. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3DSSD: Point-based 3D single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11040–11048 (2020)
28. Yoo, J.H., Kim, Y., Kim, J., Choi, J.W.: 3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection. In: European Conference on Computer Vision. pp. 720–736. Springer (2020)
29. Zhang, L., Chen, X., Dong, R., Ma, K.: Region-aware knowledge distillation for efficient image-to-image translation. arXiv preprint arXiv:2205.12451 (2022)
30. Zhang, L., Chen, X., Tu, X., Wan, P., Xu, N., Ma, K.: Wavelet knowledge distillation: Towards efficient image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12464–12474 (2022)
31. Zhang, L., Dong, R., Tai, H.S., Ma, K.: Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2022)
32. Zhang, L., Ma, K.: Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In: International Conference on Learning Representations (2020)

33. Zhang, L., Yu, M., Chen, T., Shi, Z., Bao, C., Ma, K.: Auxiliary training: Towards accurate and robust models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 372–381 (2020)
34. Zhang, P., Kang, Z., Yang, T., Zhang, X., Zheng, N., Sun, J.: Lgd: Label-guided self-distillation for object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3309–3317 (2022)
35. Zhang, Z., Sun, B., Yang, H., Huang, Q.: H3DNet: 3D object detection using hybrid geometric primitives. In: European Conference on Computer Vision. pp. 311–329. Springer (2020)
36. Zheng, W., Tang, W., Jiang, L., Fu, C.W.: SE-SSD: Self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14494–14503 (2021)
37. Zhu, Y., Wang, Y.: Student customized knowledge distillation: Bridging the gap between student and teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5057–5066 (2021)