Multimodal Transformer for Automatic 3D Annotation and Object Detection

Chang Liu, Xiaoyan Qian, Binxiao Huang, Xiaojuan Qi, Edmund Lam, Siew-Chong Tan, and Ngai Wong

The University of Hong Kong, Pokfulam, Hong Kong
{lcon7, qianxy10, huangbx7}@connect.hku.hk
{xjqi,elam,sctan,nwong}@eee.hku.hk

Abstract. Despite a growing number of datasets being collected for training 3D object detection models, significant human effort is still required to annotate 3D boxes on LiDAR scans. To automate the annotation and facilitate the production of various customized datasets, we propose an end-to-end multimodal transformer (MTrans) autolabeler, which leverages both LiDAR scans and images to generate precise 3D box annotations from weak 2D bounding boxes. To alleviate the pervasive sparsity problem that hinders existing autolabelers, MTrans densifies the sparse point clouds by generating new 3D points based on 2D image information. With a multi-task design, MTrans segments the foreground/background, densifies LiDAR point clouds, and regresses 3D boxes simultaneously. Experimental results verify the effectiveness of the MTrans for improving the quality of the generated labels. By enriching the sparse point clouds, our method achieves 4.48% and 4.03% better 3D AP on KITTI moderate and hard samples, respectively, versus the state-of-the-art autolabeler. MTrans can also be extended to improve the accuracy for 3D object detection, resulting in a remarkable 89.45% AP on KITTI hard samples. Codes are at `https://github.com/Cliu2/MTrans`.

Keywords: 3D Autolabeler, 3D Object Detection, Multimodal Vision, Self-attention, Self-supervision, Transformer

1 Introduction

3D detection technologies have seen rapid development in recent years, with considerable importance for tasks such as autonomous driving and robotics. Large-scale datasets can often improve the performance and generality of deep neural networks, hence new datasets such as nuScenes [1] and Waymo Open Dataset [20] include millions of annotated objects and hundreds of thousands of LiDAR scans. Nonetheless, the annotation procedure is extremely labor-intensive, especially for 3D point clouds [12,26]. Therefore, it is imperative to explore ways to accelerate the annotation procedure.

Compared with 3D annotations, 2D bounding boxes are much easier to obtain. Hence, we investigate the generation of 3D annotations from weak 2D boxes. Despite its substantial practical value, only a few existing works study the automatic annotation problem. Taking point clouds and images as the multimodal inputs, FGR [26] converts 2D boxes to 3D boxes with a non-learning algorithm. Alternatively, WS3D [11] leverages center clicks as weak annotations, and directly regresses 3D boxes from cylindrical proposals centered by the clicks. Both methods leverage the image information to reduce the problem space and achieves state-of-the-art performances on the KITTI dataset [5]. Nevertheless, their methods overlook the pervasive sparsity issue of point clouds. As shown on the left of Fig. 1, even though the problem domain is narrowed down by the weak 2D annotations, it is still challenging to generate a 3D box if the points are too sparse, due to the ambiguity in orientation and scale. Consequently, although these methods generate human-comparable annotations for easy samples, they suffer obvious accuracy drops for hard samples that are usually sparse, by 9.13% and 6.47% in 3D AP, respectively.

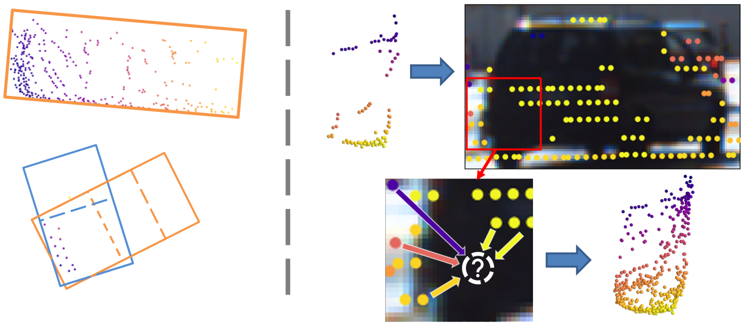

Fig. 1. Left: The sparsity issue of point cloud for 3D box annotation. Compared with the top object, the bottom is ambiguous in orientation and scale. Both are from the Bird's Eye View of LiDAR scans. **Right:** Interpolating 3D points by referring to context points based on 2D semantic and geometric relationships. Point's depth shown in color.

The sparsity problem remains a fundamental challenge for automatic annotations. Besides the sparsity issue, point clouds lack color information. Therefore, similar-shaped objects (e.g., traffic light post and pedestrian) are difficult to distinguish, which further complicates the problem [23]. Fortunately, images provide dense fine-grid RGB information, which is an effective complement to the sparse point clouds. Motivated by this, we investigate point cloud enrichment with image information. As shown on the right of Fig. 1, 3D LiDAR points can be projected onto the image, and therefore an image pixel's 3D coordinates can be estimated by referring to the context LiDAR points, based on their 2D spatial and semantic relationships. This way, new 3D points can be generated from 2D pixels, and hence densify the sparse point cloud.

To this end, we propose the end-to-end multimodal transformer (MTrans) which alleviates the sparsity problem and outputs accurate 3D box annotations given weak 2D boxes. In order to leverage the dense image information to enrich sparse point clouds, we present the multimodal self-attention module, which efficiently extracts multimodal 2D and 3D features, modeling points' geometric and semantic relationships. To train MTrans, we also introduce a multi-task design, so that it learns to generate new 3D points to reduce the sparsity of point clouds while predicting 3D boxes. The multi-task design also includes a foreground segmentation task to encourage MTrans to uncover point-wise semantics. The multimodal self-attention and multi-task design work collectively and enable the MTrans to efficiently exploit multimodal information to address the sparsity issue, and hence facilitate more accurate 3D box prediction. Moreover, to exploit unlabelled data for the point generation task, a *mask-and-predict* self-supervision strategy can be carried out. With a small amount of annotated data (e.g., 500 frames) and optional unlabeled data, our method can generate high-quality 3D box annotations for training other 3D object detectors. Comprehensive experiments are conducted, and our MTrans outperforms all the other baseline methods significantly. Using only 500 annotated frames, PointRCNN trained with MTrans can achieve 99.62% of its original performance trained with full human annotations, which is 3.65% higher in mAP than the state-of-the-art autolabeler. Furthermore, our point generation approach for handling the sparsity issue also has direct value for 3D object detection tasks. Extended for object detection, MTrans surpasses similar 2D-box-aided methods of F-PointNet [14] and F-ConvNet [25] by 19.6% and 10.8% mAP, respectively. Our contributions are threefold:

1. We propose a novel autolabeler, MTrans, which generates high-quality 3D boxes from weak 2D bounding boxes, thus greatly saving manual work.
2. MTrans effectively alleviates the pervasive sparsity issue for 3D point clouds, and it can also be extended for 3D object detection tasks.
3. Comprehensive experiments are conducted to justify our method. MTrans outperforms existing state-of-the-art autolabelers by large margins. Visualization results are also provided for interpretability.

2 Related Work

2.1 Multimodal 3D Object Detection

Several previous studies investigate multimodal networks, utilizing images and LiDAR point clouds for a more comprehensive perception. Most of them extract features from images and point clouds in different branches, followed by fusing the information for final decision [2,6,33,27]. MV3D [2] proposes the deep fusion manner to repetitively exchange information from the different views including the image, LiDAR's front view and Bird's Eye View (BEV), during the inference. Later, EPNet [6] adopts Feature Pyramid Network to encode LiDAR's front view and RGB images, and fuses the two feature pyramids with the gating mechanism. Similarly, ACMNet [33] and 3D-CVF [31] are based on gating fusion as well. PI-RCNN [27] presents a hybrid PACF fusion module. To align different views of the image and point clouds during fusion, AVOD [7] introduces the 3D anchor grid to generate 3D proposals and applies fusion only for the proposed region. Alternatively, Pi-RCNN [27] and PointPainting [23] directly extract point cloud features with point-based backbones (e.g., PointNet [15] and PointNet++ [16]), and augment them with image features through LiDAR-image calibration.

Rather than fusing different modalities, some other research employs images to narrow down the problem space. PointPainting [23] performs image semantic segmentation and highlights the points corresponding to the interested 2D region. Similarly, Ref. [10] performs instance segmentation on images to filter the 3D point clouds. F-PointNet [14] and F-Convnet [25] leverage 2D bounding boxes on images to locate a frustum region of the point cloud, and hence lowers the difficulty and improves accuracy. CLOCs [13] generates 2D box proposals from images to filter 3D proposals, exploiting the consistency of the two modalities to improve the overall detection accuracy.

The approaches above demonstrated the value of multiple modalities. The dense image data can complement the sparse point clouds. Nevertheless, the number of points remains unchanged, hence the sparsity problem is not substantially solved. Also, the multi-branch design introduces extra computation and memory overheads, leaving it still an open question on how to fully utilize multimodal information [24].

2.2 Single-modal Detectors Addressing the Sparsity Problem

Noticing the sparsity problem, some researchers try to enrich the sparse clouds to improve detection accuracy. SPG [29] firstly locates foreground regions, where an object is likely to exist, and then generates semantic points with a voxel-based point generation module. Specifically, the network predicts whether each voxel is in the foreground or background, and then generates a new point for each foreground voxel. BtcDet [28] densifies the point clouds within voxel-based proposals as well. Instead of directly generating new points, BtcDet finds a best matched prior object for each proposal, and densifies the proposal by mirroring the original points as well as aggregating the points from the prior object.

Both SPG and BtcDet alleviate the sparsity problem, resulting in state-of-the-art performances on the KITTI detection dataset. The two works demonstrate the value of enriching the sparse point cloud for the 3D object detection task. Nonetheless, they are all trained with a large amount of annotated data, which are not accessible for the automatic annotation task. Additionally, they only leverage single modality, omitting the dense RGB information from images.

2.3 Autolabelers for 3D Object Detection

Although autolabelers have considerable potential for real-world applications, only a few works have investigated this problem. FGR [26] is a non-learning approach that takes 2D bounding boxes to locate frustum sub-point-clouds and removes ground points with the RANSAC algorithm. After that, they heuristically calculate the tightest 3D bounding box that can wrap all remaining points for each object. In contrast, other deep-learning approaches train their networks with a small amount of human annotations, and generate 3D boxes from inexpensive weak annotations.

CC [21] extracts frustum sub-clouds in the same way as FGR, but adopts PointNets [15] for segmenting and regressing 3D boxes. SDF [32] uses predefined CAD models to estimate the 3D geometry of cars detected in 2D images. VS3D [17] generates 3D proposals based on cloud density with an unsupervised UPM module. Points in the proposals together with cropped images are fed into a classification/regression network for the final 3D boxes. Later, WS3D [12] and WS3D(2021) [11] utilize center clicks as weak 2D annotations. Human annotators are asked to click on the objects' centroids on images, followed by adjusting the position in BEV of LiDAR scans. Points within a cylindrical proposal centered by each click are processed by a backbone network for the 3D pseudo box.

The above autolabelers utilize image modality and weak annotations to automatically generate 3D boxes for point clouds. However, they mainly use images to coarsely locate objects but do not address the sparsity problem of point clouds. Consequently, their performances are still sub-optimal compared with human annotations, having noticeable accuracy drops, especially for moderate and hard samples, which are usually far and sparse.

3 MTrans for Automatic Annotation

In this section, we first define the automatic annotation problem, and then introduce the proposed MTrans as an autolabeler, including multimodal self-attention mechanism, multi-task design, and self-supervised point generation.

Automatic Annotation. Given the LiDAR scan and its corresponding image, this research aims to generate 3D bounding box annotations for interested objects (i.e., Cars) from weak annotations of 2D bounding boxes. LiDAR points and image pixels are connected by the LiDAR-image projection based on the calibration parameters. Each point in the LiDAR point cloud can be projected to a pixel on the corresponding image. The autolabeler is trained with a small amount of human annotated data (e.g., 500 frames) and then used to label the remaining data. Another object detection network can be trained with the auto-generated 3D labels, saving manual work significantly.

Overview. As shown in Fig. 2, to address the sparsity problem and generate high quality 3D annotations, our MTrans takes multimodal inputs of LiDAR point clouds and their corresponding images (Sec. 3.1). The multimodal inputs are encoded and fused as point-level embedding vectors for further processing (Sec. 3.2). With the proposed multimodal self-attention module, object features are extracted based on the points' semantic and geometric relationships (Sec. 3.3). Then the extracted features are used for multiple tasks of foreground segmentation, point generation, and 3D box regression, encouraging the MTrans

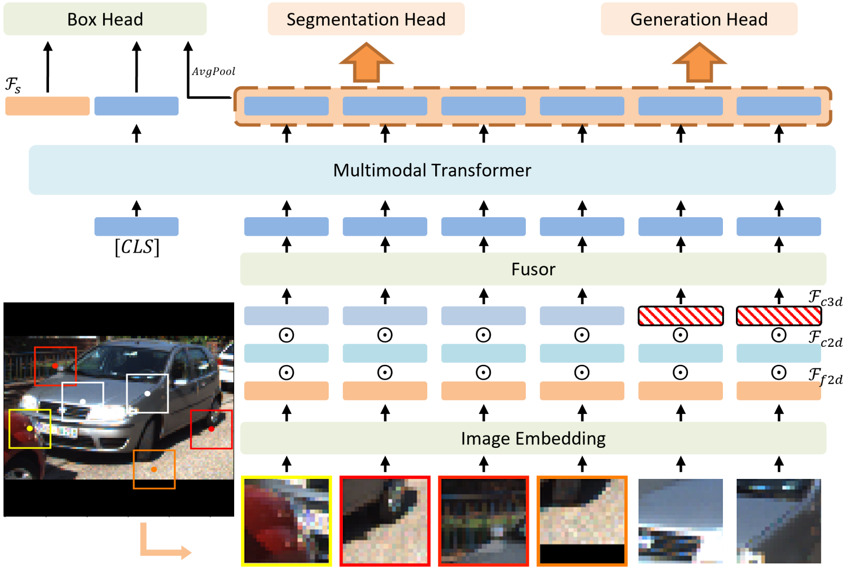

Fig. 2. Workflow of the MTrans. The network models both 3D and 2D information from the LiDAR scans and images. The frustum points and the image are fed into the network for simultaneous foreground segmentation, point generation, and box regression. $\mathcal{F}_{f2d}$, $\mathcal{F}_{c2d}$ and $\mathcal{F}_{c3d}$ are the embeddings for image patches, 2D coordinates and 3D coordinates. $\mathcal{F}_s$ is the global image feature derived by averaging $\mathcal{F}_{f2d}$.

to comprehensively utilize the multimodal information and alleviate the sparsity problem (Sec. 3.4). Additionally, a self-supervised training strategy is proposed to leverage unlabeled data for the point generation task (Sec. 3.5).

3.1 Data Preparation

Our MTrans generates one 3D bounding box for each object in a weak 2D box. Using the LiDAR-image calibration parameters, a 3D point (x, y, z) can be projected onto the image plane (u, v) by the mapping function f_{cal}. Therefore, we can extract the frustum sub-cloud $\mathcal{P}_F$ corresponding to a 2D box:

$$\mathcal{P}_F = \{(x, y, z) \mid f_{cal}(x, y, z) \in \mathcal{B}_{2D}\}, \tag{1}$$

where $\mathcal{B}_{2D}$ is the region cropped by the 2D box. We use n to represent the frustum cloud size (i.e., the number of points). The points' 2D projections are defined as:

$$\mathcal{C}_{2D} = \{(u, v) \mid (u, v) = f_{cal}(x, y, z),\ (x, y, z) \in \mathcal{P}_F\}, \tag{2}$$

A shown in Fig. 2, centered by the 2D projections, image patches of shape $k \times k$ are extracted, denoted as $\mathcal{I}_p$. Thus, every LiDAR point has three categories of features, $\mathcal{P}_F^{(i)}$, $\mathcal{C}_{2D}^{(i)}$ and $\mathcal{I}_p^{(i)}$.

As mentioned in Sec. 1, in order to alleviate the sparsity problem, we generate 3D points from the image pixels, by referring to the n real LiDAR points as *context*. Since the number of real LiDAR points, n, varies from sample to sample, we randomly drop points or sample pixels from images as *padding*, resulting in fixed n' elements. Besides the padding points, we further sample another m pixels uniformly from the image, called *target* points. The padding and target points only have 2D features ($\mathcal{C}_{2D}^{(i)}$ and $\mathcal{I}_p^{(i)}$) without known 3D coordinates ($\mathcal{P}_F^{(i)}$).

3.2 Feature Embeddings and Object Representation

Same as vanilla transformers [22,4], we encode the object features as embedding vectors. For the 2D position, we use sinusoidal position embedding [22] to encode u and v into a half-length vector respectively, and then concatenate them for the 2D position embedding $\mathcal{F}_{c2d}$. As the 3D coordinates are continuous, a multilayer perceptron (MLP) is used to encode the (x, y, z) coordinates into 3D position embeddings $\mathcal{F}_{c3d}$. Lastly, we extract image patch features $\mathcal{F}_{f2d}$ with a 3-layer convolutional sub-network from $\mathcal{I}_p$. Specifically, we empirically set the patch size as 7×7 and kernel size as 3×3. All of $\mathcal{F}_{c2d}$, $\mathcal{F}_{c3d}$ and $\mathcal{F}_{f2d}$ have shapes of $((n' + m), d)$, where d is the embedding vector length, for the $n' + m$ points. Note that for the padding and target points, they have no known 3D coordinates. Therefore, their $\mathcal{F}_{c3d}$ features are replaced by a trainable embedding vector $\mathcal{E}$.

Afterwards, we fuse the three kinds of features by concatenating them along the d-axis, and decrease the channel from $3d$ back to d by a fully-connected layer. As shown in Fig. 2, we introduce an object-level token $[CLS]$, which is a randomly-initialized trainable embedding vector. The total $n' + m + 1$ element representations of the object, $\mathcal{F} \in \mathbb{R}^{(n'+m+1)\times d}$, are then input into the MTrans for feature extraction with the proposed multimodal self-attention mechanism.

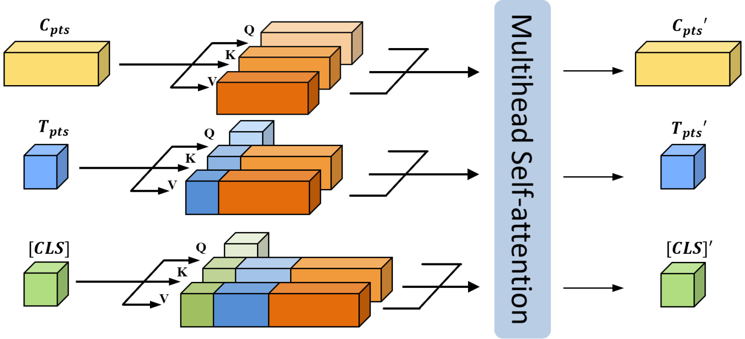

Fig. 3. Multimodal Self-attention. C-pts with known 3D coordinates attend to each other, while each of the T-pts attends to itself as well as the C-pts. $[CLS]$ attends to all available C-pts and T-pts for object-level representation. For better visual effect, we show one T-pt doing self-attention in the middle row. Best viewed in color.

3.3 Multimodal Self-attention

Following the vanilla transformer [22,4], we manipulate the element representations with the multi-head self-attention mechanism. The element representations are transformed into query Q, key K, and value V. The values V are recombined based on attention scores calculated by multiplying Q and K. Owing to the space limitation, we do not repeat the technical details of the multi-head self-attention mechanism, but only stress on the modifications for our MTrans.

Different from the original transformer, under our problem setting, some elements contain incomplete information (i.e., the padding and target points have only 2D information but no 3D information). Therefore, we treat the original frustum sub-cloud points with full information as the context points (*C-pts* in Fig. 3), and recover 3D coordinates for the padding and target elements (*T-pts*). Specifically, the T-pts look up C-pts based on their position and semantic relationships, and interpolate their 3D coordinates to enrich the sparse point cloud. During our multimodal self-attention, the C-pts only attend to each other but not T-pts, while each T-pt can also attend to itself, and $[CLS]$ attends to all available points to aggregate object-level information.

As shown in Fig. 3, the C-pts perform vanilla self-attention, and have representations updated as C-pts$'$. For each T-pts, a one-element Q is built from its element representation, while K and V are from the concatenation of itself and the C-pts. Similarly, $[CLS]$ has Q from itself, but K and V from the concatenation of $[CLS]$, T-pts, and C-pts. With this asymmetric design, T-pts update themselves by absorbing 3D information from C-pts, and $[CLS]$ encodes a global representation of the object. For our MTrans, we stack 4 multimodal self-attention layers, and vanilla feed-forward-layer [22] is adopted between each pair of the self-attention layers.

3.4 Multi-task Prediction Heads

After the self-attention layers, the transformed element representations are fed into multiple prediction heads for the multi-task supervision. Firstly, a 2-layer MLP $\mathcal{H}_{seg}$ performs binary segmentation for each point. Points within the ground-truth 3D bounding box are regarded as foreground, otherwise background. The segmentation loss $\mathcal{L}_{seg}$ is the summation of Cross-Entropy loss and Dice loss. Since T-pts have no 3D coordinates, only C-pts are supervised.

Another 2-layer MLP head $\mathcal{H}_{xyz}$ predicts the 3D coordinates for the point generation task. Since there are no ground-truth values for the padding and target points, we adopt a mask-and-predict strategy to train this task with self-supervision, which will be introduced in detail in the next section. Smooth L_1 loss is used for the point generation task as $\mathcal{L}_{xyz}$. Intuitively, the foreground is more critical than background, so we only calculate $\mathcal{L}_{xyz}$ for the foreground points predicted by $\mathcal{H}_{seg}$. Moreover, we multiply the loss with a weight of 0.1 if the point is a context point due to the leaked 3D coordinates, otherwise 1.

The third prediction head $\mathcal{H}_{box}$ directly regresses 3D bounding boxes from the object-level representations, which is derived by concatenating the $[CLS]$

representation, the averaged element representations, and the averaged image feature of $\mathcal{F}_s$, as shown in Fig. 2. The output box is a 7-element vector for the location ((x, y, z) coordinates), dimension (length, width, height), and rotation along the z-axis. We use the dIoU loss [35,9,34] for the box loss $\mathcal{L}_{box}$.

As the IoU (intersection over union) metric is direction-invariant (i.e., rotating a box for 180 degrees does not change the IoU), we further introduce a direction head for the binary direction classification. Specifically, orientation within $[-\pi/2, \pi/2)$ is regarded as the front, and $[-\pi, -\pi/2) \cup [\pi/2, \pi]$ as the back. Cross Entropy loss is used for the direction loss $\mathcal{L}_{dir}$.

All the heads are trained simultaneously, and the overall loss is given by:

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{xyz} + \lambda_{box}\mathcal{L}_{box} + \mathcal{L}_{dir}, \tag{3}$$

where the λ_{box} is a coefficient and we empirically set it as 5.

3.5 Self-supervised Point Generation

One major motivation of the proposed MTrans is to enrich the sparse point clouds with image information. As mentioned above, beyond the context points, some extra T-pts (i.e., padding and target points) are appended with unknown 3D information. The point generation task, supervised by the loss $\mathcal{L}_{xyz}$, aims to uncover the 3D coordinates for these T-pts based on their 2D geometric and semantic relationships with the context points. Due to the lack of ground truths, we adopt a self-supervised strategy for this task. Specifically, we randomly mask a portion ($0 \sim 95\%$) of the context points, by replacing their $\mathcal{F}_{c3d}$ with the trainable embedding vector $\mathcal{E}$. The masked points are then treated as T-pts and supervised by the $\mathcal{L}_{xyz}$. With this partial supervision, we managed to train the MTrans for the point generation task. During the inference, no context points are masked and the model recovers the 3D coordinates for the real T-pts.

4 Extension of MTrans to 3D Object Detection

Inspired by F-PointNet [14] and F-ConvNet [25], we can also convert the MTrans into a 3D object detector. Similarly, our model lifts 2D detection results up to 3D bounding boxes. The architecture and workflow remain the same as in Sec. 3. However, to be compatible for the detection metric of Average Precision (AP), we add another prediction head $\mathcal{H}_c$ for the confidence of the generated 3D box. The confidence is indicated by the estimated IoU score of each generated 3D box. Instead of using a small amount of data for the automatic annotation task, the detector version of the MTrans is trained with full data. The overall loss is also updated as:

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{xyz} + \lambda_{box}\mathcal{L}_{iou} + \mathcal{L}_{dir} + \mathcal{L}_c, \tag{4}$$

where the new term $\mathcal{L}_c$ is the Smooth L_1 loss for confidence head $\mathcal{H}_c$.

5 Experimental Evaluations

The proposed MTrans is evaluated for two tasks, the automatic annotation and 3D object detection. We also conduct ablation studies to justify the contributions of each module.

KITTI Dataset. KITTI [5] is a benchmark dataset for 3D detection for autonomous driving. We follow the official practice to split the dataset into training and validation sets with 3,712 and 3,769 frames. The standard evaluation metric of AP with the IoU threshold being 0.7 is adopted. Same as FGR [26], we focus on the Car class and filter out objects with less than 5 foreground LiDAR points, due to some samples having too few points to possibly draw a 3D box.

5.1 Implementation Details

For the MTrans, 4 multimodal self-attention layers with the hidden size of 768 and 12 heads are used. As an autolabeler, the model is trained with 500 or 125 annotated frames for 300 epochs. The remaining frames are also used as unlabeled data for self-supervision on the point generation task. Adam optimizer is employed with a learning rate of 10^{-4}. We also use standard image augmentation methods of auto contrast, random sharpness and color jittering, as well as point cloud augmentation methods of random translation, scaling and mirroring. Due to some 2D boxes having overlaps, an overlap mask of shape is added as a new channel of the image, where the overlapped areas have values of 0, and otherwise 1. For each object, the cloud size n' is set to 512 for context plus padding points, and m is 784 for the number of image-sampled target points.

5.2 Automatic Annotation Results

To evaluate our MTrans as an autolabeler, we employ two popular backbone object detection networks, PointRCNN [19] and PointPillars [18], and retrain them with the auto-generated 3D annotations. To examine the influence of different data scales, we also train them with all frames, 500 frames, and 125 frames of human-annotated data, respectively. As shown in Tab. 1, the accuracy drops dramatically when the data are insufficient, especially for the moderate and hard samples. With 125 frames of annotated data, PointPillars and PointRCNN suffer mean accuracy drops of 16.9% and 6.23% in AP.

Meanwhile, the proposed MTrans is also trained with 500 frames and 125 frames of annotated data, and then re-annotates the whole KITTI *train* set. As shown in Tab. 1, trained with the MTrans-generated annotations, the two networks both significantly outperform their counterparts. For PointPillars, our method brings 7.06% and 12.42% absolute average increases in AP on the *val* set for 500 and 125 frames of data, respectively. Impressively, using only 500 and 125 frames, PointRCNN trained with MTrans can achieve 99.62% and 97.24% of the original model trained with full data on the *val* set. On the *test* set,

Table 1. Automatic annotation results on KITTI *val* set and *test* set.

Method	Fully Supervised	AP_{3D} Val			AP_{3D} Test		
		Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [8]	✓	86.10	76.58	72.79	82.58	74.31	68.99
PointPillars [8]	500f	80.65	67.65	66.14	-	-	-
PointPillars [8]	125f	71.36	58.29	55.11	-	-	-
Ours + PointPillars	500f	86.69	76.56	72.38	77.65	67.48	62.38
Ours + PointPillars	125f	83.70	71.66	66.67	-	-	-
PointRCNN [19]	✓	88.99	78.71	78.21	86.96	75.64	70.70
PointRCNN [19]	500f	88.54	76.85	70.12	-	-	-
PointRCNN [19]	125f	85.63	73.60	67.98	-	-	-
Ours + PointRCNN	500f	88.72	78.84	77.43	83.42	75.07	68.26
Ours + PointRCNN	125f	87.64	77.31	74.32	-	-	-
Compare with other autolabelers							
WS3D [12]	BEV Centroid	84.04	75.10	73.29	80.15	69.64	63.71
WS3D(2021) [11]	BEV Centroid	85.04	*75.94*	*74.38*	*80.99*	*70.59*	*64.23*
FGR [26]	2D Box	*86.67*	73.55	67.90	80.26	68.47	61.57
Ours (PointRCNN)	2D Box	**88.72**	**78.84**	**77.43**	**83.42**	**75.07**	**68.26**

MTrans-trained PointRCNN also yields 97.19% of the original model. The above results demonstrate that our MTrans can generate high-quality 3D annotations, producing object detection networks comparable to their counterparts trained with human annotations. Automatically raising 2D boxes into 3D annotations, our method greatly saves human efforts for the annotation procedure.

We also compare our method with existing state-of-the-art autolabelers, FGR [26] and WS3D [11,12]. Although they also generate 3D boxes from weak 2D annotations, we want to stress the difference in task design for the three approaches. FGR uses 2D boxes, same as in MTrans, but is a non-learning algorithm, while WS3D uses center-clicks on the image and BEV, which might provide extra 3D information. Using the same PointRCNN backbone, our method surpasses all the above baselines significantly. Especially for the moderate and hard samples, MTrans improves the 3D AP by 4.48% and 4.03% respectively. This observation aligns with the motivation of enriching the sparse point cloud for higher quality of generated labels, since moderate and hard samples usually suffer more from the sparsity issue than easy samples.

5.3 Detection Results

As mentioned in Sec. 4, with an extra prediction head for the box confidence, the proposed MTrans can also be applied for 3D object detection task. Same as F-PointNet [14] and F-ConvNet [25], we train our MTrans with full data, and use ground-truth 2D boxes for frustum proposals. We evaluate our method on the KITTI *val* set, using metrics of AP_{3D} and AP_{3D} R40 (40 recall thresholds).

Table 2. Detection results on KITTI *val* set.

Method	Modality	Extra Ref. Signal	AP$_{3D}$			AP$_{3D}$ R40		
			Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [2]	LiDAR+RGB	✗	71.29	62.68	56.56	-	-	-
F-PointNet [14]	LiDAR+RGB	2D box	83.76	70.92	63.65	-	-	-
F-ConvNet [25]	LiDAR+RGB	2D box	89.02	78.80	77.09	-	-	-
VPFNet [36]	LiDAR+RGB	✗	-	-	-	93.42	88.76	86.05
SECOND [30]	LiDAR	✗	87.43	76.48	69.10	90.97	79.94	77.09
Voxel-RCNN [3]	LiDAR	✗	89.41	84.52	78.93	92.38	85.29	82.86
PV-RCNN [18]	LiDAR	✗	89.35	83.69	78.70	92.57	84.83	82.69
SPG [29]	LiDAR	✗	89.81	84.45	79.14	92.53	85.31	82.82
BtcDet [28]	LiDAR	✗	-	-	-	93.15	86.28	83.86
Ours	LiDAR+RGB	2D box	**97.82**	**89.89**	**89.45**	**98.83**	**93.57**	**90.95**

Shown in Tab. 2, our method achieves impressive performances of 92.38% and 94.46% for mAP and mAP(R40). The detection accuracy is significantly higher than other state-of-the-art baselines. However, same as F-PointNet [14] and F-ConvNet [25], our method utilizes the ground-truth 2D boxes as extra reference signals, which reduces the problem difficulty. Nonetheless, in the fair comparison with F-PointNet and F-ConvNet under identical task and experiment settings, our method still boosts the mAP by 19.61% and 10.75%, respectively.

Table 3. Ablation Results.

	MM	Seg	Gen	ExtraPts	SelfSup	Metric			
						mIoU	Recall	mAP	mAP$_{R40}$
Vanilla	✗	✗	✗	✗	✗	60.07	41.53	48.57	46.89
Multimodal	✓	✗	✗	✗	✗	65.57	53.30	58.59	57.33
Seg Loss	✓	✓	✗	✗	✗	69.45	61.53	67.84	69.42
XYZ Loss	✓	✗	✓	✗	✗	67.61	57.76	63.62	63.82
Self-supervise	✓	✗	✓	✗	✓	68.42	59.82	65.25	66.89
Densify	✓	✗	✓	✓	✗	75.37	74.95	80.82	83.47
Full Generate	✓	✗	✓	✓	✓	76.84	78.48	83.98	85.72
Ours	✓	✓	✓	✓	✓	**77.50**	**81.22**	**86.24**	**90.30**
Ours(all frames)	✓	✓	✓	✓	✓	81.01	87.65	92.36	94.43

5.4 Ablation Study

Ablation studies are conducted to verify the technical contributions of the proposed method. We train the MTrans with 500 annotated frames of the KITTI *train* set, and evaluate the model on the *val* set. The generated 3D boxes are compared with the ground-truth human annotations, and three metrics are employed, namely, the mean IoU (mIoU), recall with IoU threshold of 0.7, and mean average precision (mAP, mAP$_{R40}$) over the Car class.

As shown in Tab. 3, the vanilla transformer, which directly processes 3D LiDAR points $\mathcal{F}_{c3d}$, cannot model the problem well with the small amount of data with only 500 frames. The *Multimodal* variant introduces extra modality of images, and improves the mIoU and recall significantly by 5.5% and 11.77%, respectively. *Seg Loss* and *XYZ Loss* introduce the multi-task design, and again improve the accuracy significantly by 12.09 and 6.49% mAP$_{R40}$, demonstrating the contribution of the auxiliary losses. For *Densify*, extra target points are evenly sampled from the image, and their 3D coordinates are estimated to densify the point cloud a step further. Alternatively, *Self-supervise* utilize unlabeled data for the point generation task, also improving the accuracy, resulting in 68.42% mIoU. Combining all the point cloud densification techniques, *Full Generate*'s mIoU and recall both surpass 70%. Leveraging all the modules, our MTrans archives impressive 81.22% Recall and 86.24% mAP using only 500 frames of annotated data, which is even comparable with state-of-the-art object detectors trained with full annotations.

The ablation studies justify the contributions of our four innovations, namely, the multimodal network, auxiliary multiple tasks, point generation, and self-supervision. The multimodal inputs can effectively boost accuracy with negligible modification on the vanilla transformer (*Multimodal* variant). Moreover, the introduction of the auxiliary multi-task design (*Seg Loss* and *XYZ Loss*) raises the mIoU by 2.04% and 3.88%, showing that the model is encouraged to understand the problem more comprehensively from multiple aspects. The largest gain, however, comes from adding extra target points from the image, which increases the mIoU by 7.76% and mAP by 20.35% (*Densify*). This observation well supports our motivation that image-guided interpolation can enrich the sparse point cloud for better annotations. Combining all the techniques, our final MTrans yields a surprising 77.5% mIoU and 81.22% recall, with only 500 annotated frames. Nonetheless, the gains from these techniques tend to be saturated when combining all of the modules. Compared with the mAP accuracy for all-data-trained MTrans at the bottom of Tab. 1, we hypothesize that the main constraint is the amount of training data.

6 Visualization of Attention on Context Points

In Fig. 4 we visualize the attention distribution over context points. Specifically, for each context point, we average the attention scores it received from all other points. The attention scores of all context points sum to 1. As shown in the figure, points that belong to the car tends to receive more attention than background points, which aligns with our intuition, since background points provide little reference value of labeling 3D boxes. Moreover, the attention distribution also demonstrates a coherent spatial pattern in both the 2D image contents and

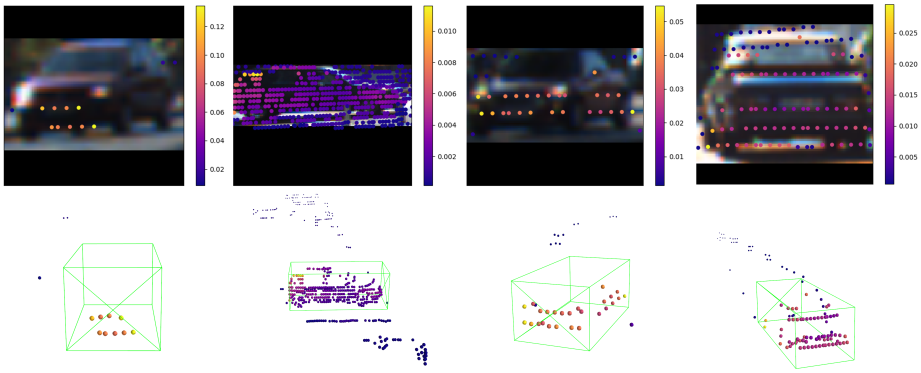

Fig. 4. Attention score visualization results. Context points and their average attention scores are plotted. Each point's color represents the amount of attention it receives from other points. The attention scores sum to 1 for each data sample. It is observed that the foreground points tend to receive more attention than background points, which provides an intuitive interpretation for the multimodal attention MTrans.

the 3D space. This observation suggests that multimodal self-attention extracts object features based on 3D & 2D geometric, and 2D semantic relationships.

Another interesting phenomenon is that the foreground points closer to object edges tend to receive more attention. As shown in the figures, yellow points are presented near box corners for all four samples. This is also reasonable since corner points usually provide more critical information for 3D box prediction. The observation provides good interpretability for our MTrans.

7 Conclusion

This work has proposed the MTrans, an autolabeler that generates 3D boxes from weak 2D box annotations, given the LiDAR point clouds and corresponding images. Our method effectively alleviates the pervasive sparsity problem of point clouds. Specifically, we introduce the multimodal self-attention mechanism, together with auxiliary multi-task and self-supervision design, which leverage image information to generate extra 3D points. Experimental results demonstrate that our method can outperform existing autolabelers significantly, generating 3D box annotations comparable to human annotations. Moreover, the proposed MTrans can also be applied for the 3D object detection task, achieving state-of-the-art performances. Beyond LiDAR + monocular image, an interesting future direction would be to incorporate more modalities to improve the detection accuracy and robustness, such as radar and/or stereo images. It would also be valuable to investigate the domain transfer problem for the autolabelers, which could save human effort further. We hope this study can inspire future works to solve the sparsity problem as well as to develop automatic annotation workflows to reduce manual workloads.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
2. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
3. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1201–1209 (2021)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
6. Huang, T., Liu, Z., Chen, X., Bai, X.: Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision. pp. 35–52. Springer (2020)
7. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8. IEEE (2018)
8. 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. pp. 12697–12705 (2019)
9. lilanxiao: Differentiable iou of oriented boxes. `https://github.com/lilanxiao/Rotated_IoU` (2021)
10. McCraith, R., Insafutdinov, E., Neumann, L., Vedaldi, A.: Lifting 2d object locations to 3d by discounting lidar outliers across objects and views. arXiv preprint arXiv:2109.07945 (2021)
11. Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., Van Gool, L.: Towards a weakly supervised framework for 3d point cloud object detection and annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
12. Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., Dai, D.: Weakly supervised 3d object detection from lidar point cloud. In: European Conference on Computer Vision. pp. 515–531. Springer (2020)
13. Pang, S., Morris, D., Radha, H.: Clocs: Camera-lidar object candidates fusion for 3d object detection. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10386–10393. IEEE (2020)
14. 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)
15. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)

17. Qin, Z., Wang, J., Lu, Y.: Weakly supervised 3d object detection from point clouds. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 4144–4152 (2020)
18. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
19. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019)
20. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)
21. Tang, Y.S., Lee, G.H.: Transferable semi-supervised 3d object detection from rgb-d data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1931–1940 (2019)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
23. 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)
24. Wang, W., Tran, D., Feiszli, M.: What makes training multi-modal classification networks hard? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12695–12705 (2020)
25. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1742–1749. IEEE (2019)
26. Wei, Y., Su, S., Lu, J., Zhou, J.: Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 4348–4354. IEEE (2021)
27. Xie, L., Xiang, C., Yu, Z., Xu, G., Yang, Z., Cai, D., He, X.: Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12460–12467 (2020)
28. Xu, Q., Zhong, Y., Neumann, U.: Behind the curtain: Learning occluded shapes for 3d object detection. arXiv preprint arXiv:2112.02205 (2021)
29. 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)
30. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
31. 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: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16. pp. 720–736. Springer (2020)
32. Zakharov, S., Kehl, W., Bhargava, A., Gaidon, A.: Autolabeling 3d objects with differentiable rendering of sdf shape priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12224–12233 (2020)

33. Zhao, S., Gong, M., Fu, H., Tao, D.: Adaptive context-aware multi-modal network for depth completion. IEEE Transactions on Image Processing (2021)
34. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12993–13000 (2020)
35. Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R.: Iou loss for 2d/3d object detection. In: 2019 International Conference on 3D Vision (3DV). pp. 85–94 (2019)
36. Zhu, H., Deng, J., Zhang, Y., Ji, J., Mao, Q., Li, H., Zhang, Y.: Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion. arXiv preprint arXiv:2111.14382 (2021)