Supplementary Material for "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

1 Background of Transformer and Self-attention

In this section, we introduce the Transformer architecture and the self-attention mechanism, which are firstly proposed by Ref. [3]. The transformer is originally an encoder-decoder network for NLP (natural language processing) tasks, which consists of a stack of self-attention layers and feedforward layers. Given sequential input data, Transformers update the hidden states of each element within the sequence by the self-attention mechanism:

$$\text{Attention}(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where the Q, K, V are derived from linear transformations from the hidden states, and d_k is the dimension of hidden states for scaling. By this operation, the hidden states are updated by selectively aggregating the context, weighed by the derived attention scores.

2 nuScenes Autolabeler with MTrans

Besides the KITTI dataset, we also evaluate MTrans on the automatic annotation task with the nuScenes dataset [1] to verify its generality. In this section, we train MTrans with 500 frames of annotated data and then use it to re-label the whole training set. Another object detector, PointPillars [2], is then trained with the auto-generated labels. We also compare the generated 3D boxes with human annotated ones directly.

nuScenes Dataset. The nuScenes dataset is one of the largest open datasets for 3D detection in the field of autonomous driving, including 28,130 frames for training, 6,019 for validation and 6,008 for testing. According to the official challenge settings, similar classes and rare classes are removed, resulting in 10 out of 23 annotated classes for detection. Different from KITTI, nuScenes adopts the evaluation metric of NDS (nuScenes detection score). Moreover, the mAP is calculated by considering the 2D center distance on the ground plane instead of IoU as in KITTI. Each LiDAR frame (360 degrees) corresponds to 6 images captured by individual cameras for different view angles.

Same as in Sec. 5 in the main paper, we preprocess the dataset to extract objects with more than 5 foreground points. nuScenes does not provide 2D boxes, so we project the 3D boxes on the images, and take the outermost corners to build 2D boxes. For each 2D box, the frustum point cloud and its 2D projection are extracted. They are then fed into MTrans for estimating the 3D box. Since this research mainly focuses on the *Car* class, other classes such as pedestrian are excluded.

Implementation Details. Due to nuScenes samples being sparser than KITTI ones, we empirically set the cloud size n' as 350. Additionally, since we project 3D boxes on images, the overlap mask introduced in Sec. 5.1 in the main paper is inaccurate (e.g., some neighbor objects are occluded and hardly visible in the image but still results in a large overlapping region of the 2D box). Therefore, we modify the overlap masks to contain five levels, based on the visibility attribute of nuScenes objects. Consequently, the five possible mask values are set as 1 (no overlap) or 0.75, 0.5, 0.25, 0 denoting overlap with another object that has visibility level of 0-40%, 40-60%, 60-80% or 80-100%, respectively.

Again, 500 LiDAR frames are used to train our MTrans. After training, we use MTrans to re-label the whole training set for the *Car* class. Because the evaluation metric of nuScenes involves all the 10 classes, we keep the human annotations for other classes but replace *Car* annotations with the MTrans-generated ones. The popular object detector PointPillars [2] is re-implemented and trained with the auto-annotated data.

Table 1. Automatic annotation results on nuScenes *val* set. H.Anno for human annotations. Our method achieves comparable accuracy as the original model trained with human annotations.

Method	H.Anno	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	mAP↑	NDS↑
PointPillars	✓	35.98	26.27	37.27	31.57	19.82	41.26	55.54
Ours	500f	38.07	26.56	43.78	35.75	19.74	33.74	50.48

Table 2. MTrans generated 3D boxes in comparison with human annotations.

Method	mIoU↑	Recall(IoU = 0.7)↑	LE↓	Recall(LE = 0.5)↑
Ours (train)	74.25	74.94	0.3479	88.13
Ours (val)	73.91	73.62	0.3448	88.18

Experiment Results. As shown in Tab. 1, compared with the original PointPillars trained with all frames with human annotations, our method requires only 500 annotated frames yet achieves comparable performance. The NDS score of MTrans is 90.89% of the original PointPillars that are trained with tens of thousands of human-annotated frames. In addition to the mAP and NDS, another 5 official metrics are also reported, namely, mATE (mean Average Translation Error), mASE (mean Average Scale Error), mAOE (mean Average Orientation Error), mAVE (mean Average Velocity Error) and mAAE (mean Average Attribute Error). Except for mAOE, our MTrans yields similar performance for all other error terms, where the increases are less than 1%. Considering the enormous scale of the nuScenes dataset, MTrans can significantly save human workload for the annotation process. Moreover, the results also demonstrate that MTrans can yield good performances on different datasets, such as KITTI and nuScenes, proving the generality of our method.

To decouple the influences of the detector network and other classes such as pedestrian, besides training another object detector with the MTrans-generated annotations, we also directly compare MTrans *Car* annotations with manually labeled ground-truths. Results are shown in Tab. 2. Trained with 500 frames, MTrans re-labels the whole training set and validation set of nuScenes. We adopt four metrics, namely the mean IoU, recall with Iou threshold of 0.7, mean location error (LE) and recall with LE threshold of 0.5.

Although no existing autolabelers have studied the dataset of nuScenes for comparsion, we find our MTrans still performs robustly on the large-scale dataset. Compared with ground-truth human annotations, 3D boxes generated by MTrans have 74.25% mIoU and 74.94% recall, which is close to its performances on the KITTI dataset. Moreover, since nuScenes employs location error instead of IoU for calculating mAP, it is worth noticing that the recalls with LE threshold of 0.5 are over 88% on both the *train* and *val* sets. The results demonstrate that MTrans is able to generate high-quality 3D box annotations on different datasets robustly, therefore accelerating the annotation procedure for a broad range of applications.

3 Qualitative Results

We also show the qualitative results of MTrans, including the generated pseudo labels and densified sparse point clouds. The results demonstrate that MTrans is able to effectively address the sparsity issue and produces high-quality 3D boxes.

3.1 Comparison with SOTA Method

In Fig. 1, we compare the generated 3D boxes from our MTrans with the repeatable state-of-the-art baseline of FGR. As shown in the graph, FGR misses some objects and generates no box for them. It turns out that the objects containing fewer points (sparser) are more likely to be missed. Also, the FGR intentionally drops objects with low confidence, which even exacerbate this problem.

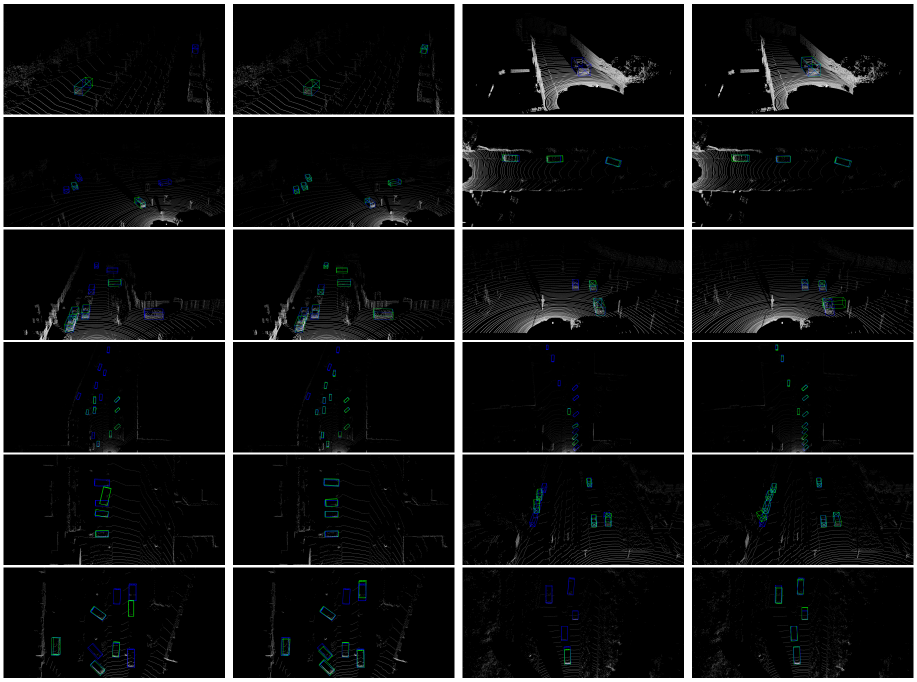

Fig. 1. Comparison with the repeatable SOTA method, FGR. Each pair of the visualization contains results from FGR (left) and MTrans (Right). Blue: ground truth 3D box, Green: generated 3D box. Zoom in for a better view.

On the contrary, MTrans produces obviously higher quality 3D boxes in general. Compared with FGR, no objects are missed due to we force MTrans to generate one 3D box for each 2D weak annotation. However, some boxes are intentionally omitted because there are too few LiDAR points within the area. Compared with FGR, in some cases MTrans has some orientation errors (e.g., row3 & row5 right in Fig. 1). We find those objects are positioned at the border of the 2D image and are truncated. Therefore incomplete frustum point clouds are fed into MTrans, which makes it difficult to predict 3D boxes.

3.2 Visualization of Point Cloud Densification

In this section, we visualize the densified point clouds to justify the effectiveness of MTrans. Given a sparse point cloud and the corresponding image, MTrans performs image-guided interpolation, estimating 3D coordinates for image pixels. By doing so, extra 3D points are generated and hence the original sparse point cloud is densified. As shown in Fig. 2, the sparse point clouds are densified in high fidelity. The distribution of the generated points follows the object shape depicted in the image. Especially for the extreme cases (row1 & row3 right), the car shapes are recovered from the original clouds that are hardly recognizable due

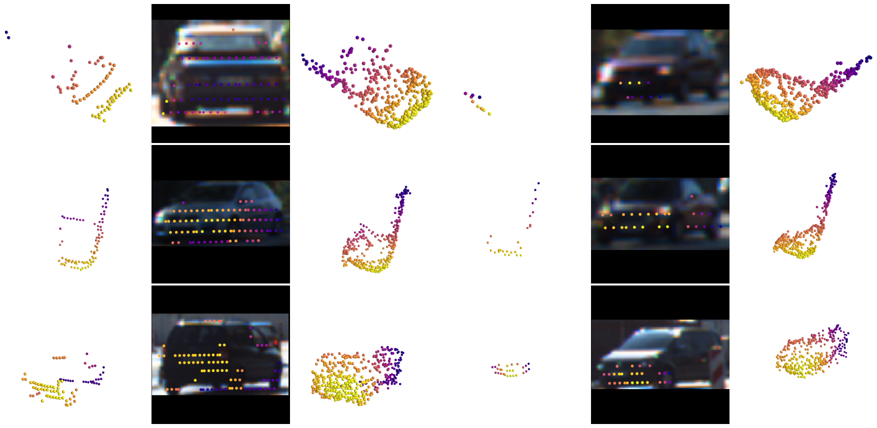

Fig. 2. Qualitative results of the densification of the point clouds. Each row contains the original sparse point cloud (left), the corresponding image with LiDAR points projected (middle) and the densified point cloud (right). Distances of the LiDAR points are denoted by the color.

to the sparsity. The visualization results prove the effectiveness of the deification procedure of our MTrans, which leverages the 2D image information to enrich the sparse point clouds.

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. 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)
3. 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)