First Place Solution to the Unified Model for Multi-task Learning of SSLAD2022

Tiancheng Wen[1], Ningzi Wang[1], Handong Wang[1], Mingfeng Wang[1,2]
Zongtai Li[1,3], and Feiyang Tan[1]

[1] MEGVII Technology
{wentiancheng, wangningzi, wanghandong, tanfeiyang}@megvii.com
[2] The Hong Kong Polytechnic University
mingfeng.wang@connect.polyu.hk
[3] IIIS, Tsinghua University
li-zt20@mails.tsinghua.edu.cn

Abstract. In this report, we present our winning solution to the unified model for muilti-task challenge of SSLAD 2022 at ECCV 2022. A multi-modal multi-task BEV based model is proposed, which individually adopts element-wise addition and two stage finetune strategy to tackle the multi-modal and multi-task problem. The overall training can be divided into ONCE pretrain and AutoScenes finetune, and series of techniques such as semi-supervised label correction and module-wise EMA is applied to bridge the gap between the two datasets. Besides, we design a cascaded segmentation head to improve lane divider performance. Our final model finally achieves 0.73 NDS and 0.67 mIOU, wining the first place in the unified model track in SSLAD 2022.

Keywords: BEV perception, multi-task learning, multi-modal learning, object detection, road segmentation

1 Introduction

Conception system for autonomous driving is responsible for providing various information based on multi-modal sensor collected data. The Self-supervised Learning for Next-Generation Industry-level Autonomous Driving Challenge at ECCV 2022 proposes AutoScenes dataset in its "Unified Model for Multi-task Learning" track. The AutoScenes dataset provides more than 3000 frames of autonomous driving scene with both lidar and camera data. The algorithms developed on the AutoScenes are proposed to be capable of both object detection and road segmentation task with multi-modal data as input. NDS (nuScences[1] detection score) and mIOU (mean insertion over union) are the evaluation metrics to these two tasks.

In this technical report, we present out solution to the challenge, which obtains 0.73 NDS and 0.67 mIOU on AutoScenes dataset. The method basically follows BEVFusion[4] pipeline and consists of ONCE pretrain and AutoScenes finetune.

2 Methods

2.1 Shared Backbones and Fusion Module

We adopt the lidar feature generator from ONCE model[8], which firstly carries out voxelization, then splits the point cloud into voxels with predefined voxel size in three directions, and finally extracts feature from each voxel by averaging the feature vectors of all the points belonging to this voxel. After obtaining voxelized features, we employ the same ResNet-like 3D backbone with the ONCE model. Finally, the 3D feature map we get is flatten into 2D BEV representation.

We adopt LSS[6] with VoVNet99-eSE[3] as camera backbone, which firstly generates 2D feature in image space and then performs projection transform and voxel pooling to produce 2D BEV feature.

Following BEVFusion[4], we fuse the lidar feature and camera feature in BEV representation via element-wise addition rather than concatenation. The two fusion methods share similar performance according to our observation on AutoScenes dataset.

2.2 Detection Head

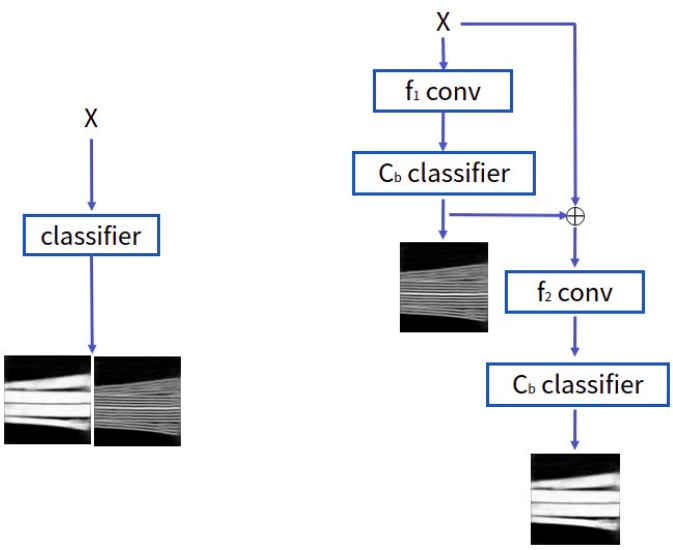
Once dataset and AutoScenes dataset share same annotated categories. Our detection head mainly inherits from ONCE model[8]. We change the subtask designs to 5 different heads. We use the same structure as ONCE model, each subtask consists of a shared convolutional layer and three separate detection heads, namely classification head, regression head, and IoU prediction head. The label assignment method is also same as ONCE model.

2.3 Segmentation Head

Our segmentation head includes a grid transformer to aggregate and represent the BEV features on a fixed-size rendered map, and a classifier to distinguish between two kinds of road surfaces.

Grid transformer This module is built for interval reduction. Due to the need of segmentation tasks, we need to express the BEV features with a fixed rendering size. However, the input scope(256×256) of feature and the output scope(200×200)of semantic feature are distinctly different. We build two grid networks to describe different rendering sizes and perform the transformation of features through bilinear interpolation. Then the transformed features are fed into the classifier for the segmentation tasks.

Semantic classifier Drivable area and lane divider may overlap in the bird's-eye view segmentation map, Thus, we split this task into two binary semantic segmentation tasks, one for each category. A more traditional approach to perform segmentation tasks is based on the [4], which directly uses a multi-channel output classifier to predict the segmentation results of two road surfaces



a: Conventional segmentation head b: The proposed cascaded head

Fig. 1: The conventional and the proposed segmentation head.

simultaneously. Given the transformed BEV feature X and a unified classifier C, the segmentation results S of two kinds of roadway can be represented as:

$$(S_a, S_b) = C(X) \tag{1}$$

where subscripts a and b denote the drivable area and lane divider respectively.

However, during the model training process, we find that the segmentation head performs bad on lane divider, this is because the representation of the divider in the bird's eye view is very steep and rugged, which makes the training of the network more difficult. To alleviate such issue, we set up two classifiers and connected them in a cascade way, and the classifier C_b is placed at the end to aid the training of C_a, as Fig. 1b shows, Furthermore, to avoid filtering useful features, we also introduce a residual structure, the computational process can be written as follows:

$$Z_b = f_1(X) \tag{2}$$

$$Z_a = f_2(X \oplus Z_b) \tag{3}$$

$$S_a = C_a(Z_a), S_b = C_b(Z_b) \tag{4}$$

where Z is the latent feature of roadways, f_1 and f_2 are basic convolution blocks, and the $\oplus$ denotes the concatenation.

2.4 Training Strategy

We use trainval split of ONCE[5] dataset for pretrain and then finetune multi-task model on AutoScenes due to the limited scale of AutoScenes. Note that the

trainval split is annotated with detection task only, which means the segmentation head of our model is unpretrained with ONCE.

An antagonism between detection and segmentation is observed when directly training the model with two tasks simultaneously. The reasons are in two folds. First, the two tasks require feature in different height. Road segmentation focuses on the ground surface but detection embraces feature above the ground. However, the BEV scheme compresses the height dimension and loses the difference. Second, based on the ONCE pretrained detection model, the segmentation head of AutoScenes model needs to be trained from scratch while the pretrained detection head and shared backbone layers needs a light finetune. And It breaks the mIOU performance to freeze the preatrained layers and finetune segmentation head only.

To address the antagonism, we maintain a individual channel attention for each task, trying to bring height knowledge into channel dimension, which is proved ineffective on minor-scale AutoScenes. Then we switch to the second point and build a two-stage joint training strategy. The key idea is to separate the update step of pretrained layers and unpretrained segmentation layers. In the first stage we finetune the pretrained layers with a small learning rate, including the detection head, and train the segmentation head with a large learning rate. In the second stage, we finetune the segmentation head with a small learning rate, and finetune other layers with a much smaller learning rate.

We further introduce different EMA(Expotional Moving Average)[2] decay parameter to the pretrained layers and the unpretrained segmentation layer. A relative large EMA decay of the pretrained layers can help slow down the training step, making it move around the optimal state obtained by prior training and provide a proper feature to both the new-added segmentation head and the pretrained detection head. In this way, the antagonism between the two tasks is eliminated.

3 Experiments

3.1 Implementation Details

We use ONCE[5] dataset for detection pre-train and then train multi-task model on AutoScenes dataset. Most Settings are based on [8], we adjust some setting to fit AutoScenes.

Data Augmentations We follow the data augmentation strategies used in 3D object detection in ONCE detection pre-train, including global rotation $\mathcal{U}\left(\frac{-\pi}{4}, \frac{\pi}{4}\right)$, global scaling $\mathcal{U}$ (0.95, 1.05), global translation $\mathcal{U}$ (-0.2m, 0.2m), randomly flipping along the x-axis and y-axis and GT sampling. In fine-tune stage, we keep these augmentations except GT sampling.

Voxelization Details The point cloud range is limited to [(-54.0, 54.0), (-54.0, 54.0), (-5.0, 3.0)] respect to x, y, z-axis during training and testing process. The voxel size along x, y, z-axis is set to [0.075m, 0.075m, 0.2m], and the max number of voxels is 120000 in training and 160000 in testing.

Point Cloud Intensity Difference The point cloud input of two dataset has significant difference on LiDAR intensity, in ONCE dataset intensity has a distribution in [0, 242] and over 90% are lower than 1, but in AutoScenes the distribution in [0, 65535] and over 90% in [0, 32767], which is much smoother than in ONCE. We use a log transform smoothing the intensity distribution in ONCE dataset, and rescale the intensity in both dataset to [0, 1] distribution.

Detection Labels AutoScenes dataset has 29 scenes in training split and 16 scenes in validation split, 2.4k labeled frames in total, as the frequency is 5 Hz, about 0.5k seconds is labeled, and in some frame the detection labels is not very accurate. Considering ONCE labels include 8k seconds, which indicates more diversity than AutoScenes, and the classes in two dataset can match, we choose 15 scenes in AutoScenes train split that our ONCE detection model behaves better, pseudo labelling those scenes. Other scenes remain unchanged, and then use those data for domain adaptation.

Box Orientation The detection labels in AutoScenes has slightly difference in definition of orientation with ONCE, that all orientation of boxes are in $[0, \pi]$, which only consider the box orientation but not the heading angle of object. We still train the model in normal definition that using heading angle as orientation so the orientation is in $[0, 2\pi]$, only change it to AutoScenes definition before evaluation.

Training-Time Hyperparameters The multi-task training consists of two stages as introduced in 2.4. In the first stage, we set the learning rate and ema decay as 1×10^{-4} and 0.999 for the pretrained modules, and the parameters of unpretrained segmentation head is set as 3×10^{-3} and 0.95. In the second stage, it is set as 1×10^{-5} and 0.9999 for pretrained layers, and 1×10^{-3} and 0.999 for the segmentation head.

Test-Time Augmentation (TTA) We use the following TTA strategies: double flip (original, horizontally flipped, vertically flipped, horizontally + vertically flipped), rotation with $6.25°$ and double flip, rotation with $-6.25°$ and double flip. (The rotation is along z-axis.)

Evaluation All evaluation based on official code on github[7]. According to detail evaluation code in detection part, we remove all out-of-range boxes before evaluation.

3.2 Results on Validation Split

Detection-only Baseline Experiments Table 1 demonstrates the baseline experiments of detection-only lidar model, evaluate on validation split. Pseudo labelling greatly increase AP in all classes except truck. ONCE pre-train model shows good performance, and finetune on pseudo labelling AutoScenes train split further improves it.

Modality-fusion & Multitask Experiments As shown in Table 2, when the model only uses lidar data for training, our vanilla model achieves 45.5% mAP

	mAP	AP_car	AP_truck	AP_bus	AP_bicycle	AP_ped
original label, train from scratch	0.179	0.492	0.0	0.143	0.111	0.151
pseudo labelling, train from scratch	0.418	0.793	0.035	0.403	0.419	0.43
once pretrain, without finetune	0.584	**0.834**	0.375	0.528	0.617	0.565
once pretrain, finetune on pseudo labelling	**0.616**	0.832	**0.379**	**0.563**	**0.625**	**0.683**

Table 1: Ablation Study on Once Pre-train and Pseudo Labelling

Methods	Pre-trained	Det.	Seg.	Modality	mAP	mIoU
Vanilla Model	✓	✓		L	0.606	/
Vanilla Model	✓		✓	L	/	0.563
Vanilla Model		✓	✓	L	0.455	0.544
Vanilla Model	✓	✓	✓	L	0.523	0.558
Fusion Model	✓	✓	✓	L+C	0.568	0.530
Fusion Model + module-wise EMA	✓	✓	✓	L+C	0.604	0.566

Table 2: Experiments about multi modality and multi task

and 54.4% mIoU. But after training with the pretrained model, Our model significantly boosts its performance by 6.8% mAP and 1.4% mIoU, demonstrating the significance of pretrained model when joint multi task training.

However, when compared with pure detection task training or pure segmentation task training, the multi-task joint training could not effectively improve the results of each sub task, or even antagonize each other. As analyzed in 2.4, this is probably because on the current dataset, the features required for detection and segmentation do not meet the envisaged complementary relationship. And with the proposed two-stage finetune strategy, the joint training model can achieve 56.8% mAP and 53.0% mIoU. Besides, the module-wise EMA can further boost model's performance, and it finally outperform previous experiments 2% mAP and 3.6% mIoU.

4 conclusions

In this report, we present a generic framework for multi-task multi-sensor 3D perception and win first place on the unified model for multi-task learning of the SSLAD2022 Challenge at ECCV 2022. We propose two stage finetune to integrate the learning of object detection and road segmentation. However, the fusion of element-wise addition is naive and the multi-modal input doesn't bring satisfying gains, especially on road segmentation task. Therefore, it is an important topic to discuss the work of camera data in mutil-modal models.

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. Klinker, F.: Exponential moving average versus moving exponential average. Mathematische Semesterberichte **58**(1), 97–107 (2011)
3. Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 0–0 (2019)
4. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. arXiv (2022)
5. Mao, J., Niu, M., Jiang, C., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, C., et al.: One million scenes for autonomous driving: Once dataset (2021)
6. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision. pp. 194–210. Springer (2020)
7. SSLAD2021: Autoscenes-eval (2022), https://github.com/SSLAD2021/AutoScenes-eval
8. Yao, Z., Huang, T., Liu, L., Wang, B., Jiang, T., Sun, J., Wang, X., Yao, H., Li, Z.: First place solution to the 3d object detection of the sslad2021 challenge (2021)