MamboNet: Adversarial Semantic Segmentation for Autonomous Driving

Jheng-Lun Liu[2], Augustine Tsai[1,3] , Chiou-Shann Fuh[2] , and Fay Huang[3]

[1] The Institute for Information Industry, Taipei, Taiwan
[2] Department of CSIE, National Taiwan University, Taipei, Taiwan
[3] Department of CSIE, National Ilan University, Yilan, Taiwan
atsai@iii.org.tw, a09911911@gmail.com , fuh@csie.ntu.edu.tw , fay@niu.edu.tw

Abstract. Environment semantic maps provide essential information for autonomous vehicles to navigate in complex road scenarios. In this paper, an adversarial network to complement the conventional encoder-decoder semantic segmentation network is introduced. A newly proposed adversarial discriminator is piggybacked to the segmentation network, which is used to improve the spatial continuity and label consistency in a scene without explicitly specifying the contextual relationships. The segmentation network itself serves as a generator to produce an initial segmentation map (pixel-wise labels). The discriminator then takes the labels and compare them with the ground truth data to further update the generator in order to enhance the accuracy of the labeling result. Quantitative evaluations were conducted which show significant improvement on spatial continuity.

Keywords: Generative Adversarial Network (GAN), Semantic Segmentation, Autonomous Driving

1 Introduction

Surrounding understanding is critical to the safety of autonomous vehicles. The ability to recognize the drivable areas and dynamic objects on the road enables the safe navigation. Conventionally, camera frames are used to detect pedestrians, cars, motorcycles, roads, and sidewalks in pixel-level. The goal of this task is to produce semantic segmentations by assigning each input data point, namely a pixel, a unique class label. With the advancements of LiDAR sensor technology in recent years, many commercial products can detect points beyond 200 meters. In this paper, we tackled the semantic segmentation task using a rotating LiDAR scanners. Comparing to solely using camera frames, 3D point clouds obtained by LiDAR provide a richer spatial and geometry information. However, the unstructured and sparse nature of the 3D data presents another level of challenges.

The major contribution of this paper is a novel method which can efficiently improve 3D LiDAR point cloud segmentation. We complemented an end-to-end encoder decoder segmentation pipeline with an adversarial network which is derived from Generative Adversarial Network (GAN)[1]. The network improves the spatial continuity

and label consistency without explicitly specifying the contextual information. The adversarial network was only applied during model training, and was removed during the online inference stage. The complexity of the overall architecture is kept in minimum.

2 Related Work

Semantic segmentation is one of the most important deep learning applications. In 2D image segmentation, U-Net [2] pioneered the encoder-decoder CNN architecture adoption, they transferred the entire feature map from encoders to the corresponding decoders and concatenates them to up-sampled (via deconvolution) decoder feature maps. In order to reduce memory requirements, Kendall [3] proposed to store the max pooling indices instead of concatenation with fewer parameters for decoder reconstruction.

Nowadays, 360-degree revolving LiDAR is the most common laser scanner for autonomous driving. In order to address 3D point cloud segmentation using aforementioned 2D segmentation paradigm, common approach is to spherically project the 3D point cloud data onto 2D range image plane. Leading the online frame-rate processing for practical applications, Wu [4] proposed a light weighted model derived from SqueezeNet to process data in 2D image plane. SqueezeSegV2 [5] extended V1 with Contextual Aggregation Module (CAM) [6] to mitigate LiDAR sensor data drop out issues.

A synthetic point cloud generation using GTA-V game engine with intensity rendering was also proposed to augment the training data. Due to nonhomogeneous spatial distribution of point cloud, SqeeuzeSegV3 [7] proposed Spatial-Adaptive Convolutions (SAC) which may change the weights according to the input data location. Miliotos [8] extended Wu [4] 3 label classes to 19 classes and replace extended the label classes from three to nineteen, and replaced the 2D CRF to 3D GPU-based nearest neighbor search acting directly on the full, un-ordered point cloud. This last step helps the retrieval of labels for all points in the cloud, even if they are occluded in the range image.

Cortinhal [9] transformed the deep network with Bayesian treatment by introducing uncertainty measures, epistemic and aleatoric noises. Luc [10] introduced an adversarial network to discriminate the predicted segmentation maps either from the ground truth or segmentation network to mitigate the higher order label inconsistencies. Souly [11] introduced a semi-supervised segmentation using weakly labelled data for the generator. In this paper, the proposed MamboNet was inspired by many of these approaches and mostly by Luc's adversarial network.

3 Method

A. 3D to 2D projection

The projection method as mentioned in [4,5,7,8,9] has been applied for data pre-processing. Each raw 3D point cloud in 360-degree surrounding is spherical projected onto a 2D grid point on a range image as illustrated in Fig. 1. A 3D point (*x*, *y*, *z*) with respect to the world coordinate system originated at the sphere center is projected to the image with coordinates of (θ_{loc} , ϕ_{loc}), which is calculated as follows:

$$\theta = \arcsin \frac{z}{\sqrt{x^2+y^2+z^2}}, \theta_{loc} = \lfloor \theta / \Delta\theta \rfloor$$
$$\phi = \arcsin \frac{y}{\sqrt{x^2+y^2}}, \phi_{loc} = \lfloor \phi / \Delta\phi \rfloor \quad (1)$$

Here, $\Delta\theta$ and $\Delta\phi$ are quantization steps. Each grid point represents a five-dimensional feature vector: three for its associated 3D location (*x*, *y*, *z*), one for the intensity value, and the other for the range value.

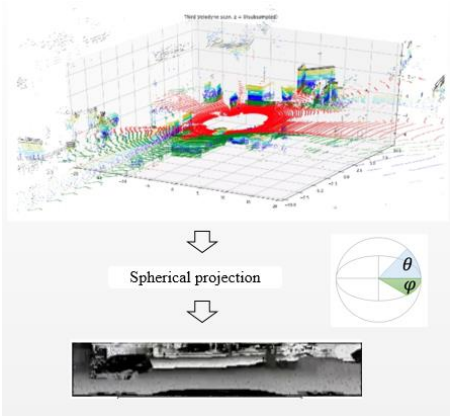

Fig. 1. Spherical projection

B. Architecture

The main objective of applying adversarial network is to enforce the spatial continuity and label consistency. Conventional encoder-decoder network [3] creates a segmentation map (pixel-wise labeling), and then follows up with conditional random field (CRF) to impose pixel grouping constraints. We replaced CRF with a discriminator which is only used during the training and can be dropped in inference to maintain minimum network complexity and it is similar to bag of freebies in [11]. Our ad-

versarial network (shown in Fig. 2) is similar to [10], the discriminator takes two inputs, namely, predicted and ground truth maps. Both maps are concatenated with the same 2D input data. The predicted map is generated by the encoder-decoder semantic segmentation network.

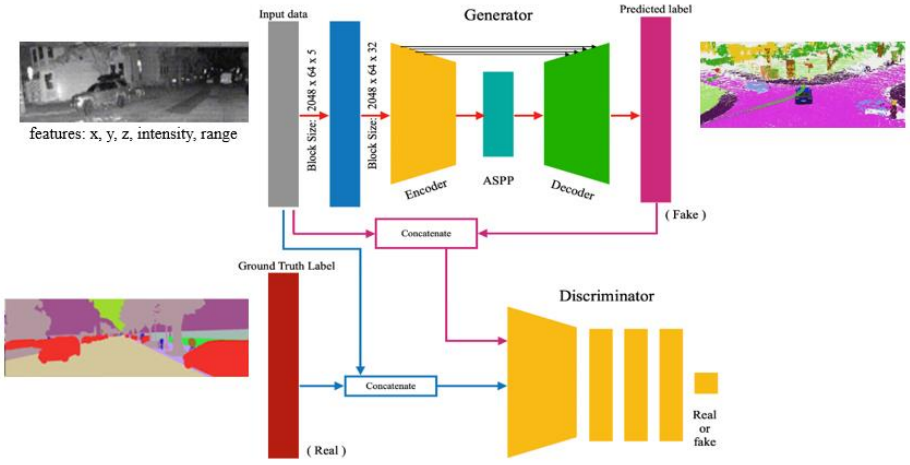

Fig. 2. Overall network architecture

A detailed version of the generator is shown in Fig. 3, each yellow block of the encoder is an Inception [13] like module with a group of mixed kernel sizes and dilation rates. Each block has three parallel convolution layers, the outputs are concatenated and then summed up with forth convolution layer. Between encoder and decoder, an Astrous Spatial Pyramid Pooling (ASPP) [2] module is inserted for exploiting multi-scale features and enlarging the receptive field. ASSP is employed to capture small street objects, such as pedestrian and cyclists. In decoder, the conventional transpose convolution layer is replaced with the low computation pixel-shuffle layer, similar to super resolution [14]. It can leverage low resolution feature map to generate up-sampled feature maps by converting information of the channel dimension to the spatial dimension. The operation is to convert a feature map of (H x W x Cr^2) to (Hr x Wr x C), where H, W, C and r are the height, width, number of channel, and up-sampling factor.

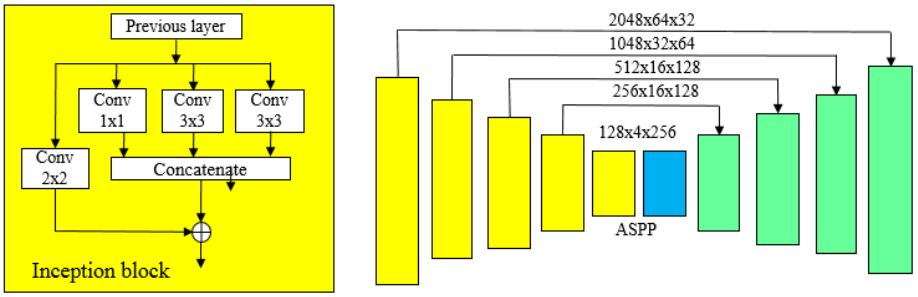

Fig. 3. Details of the generator

The discriminator is a VGG based convolutional network shown in Fig. 4. The data size is 2048 x 64 x 6. The first two dimensions are the image width and height, and the third dimension includes *x*, *y*, *z*, intensity, range, and class label. Each layer uses 3 x 3 convolution kernel and is followed by a 2 x 2 max pooling except for the 1st layer. The sizes of the last three fully connected layers are 2048, 512, and 512, respectively.

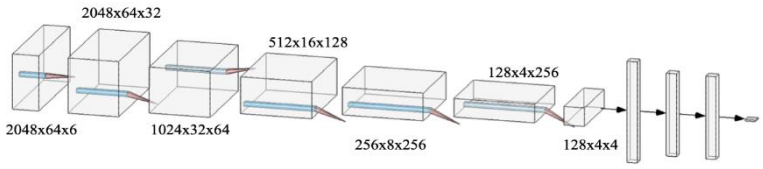

Fig. 4. Discriminator: VGG based convolutional network

C. Loss Function and Training

The training, shown in Fig. 5, is based on conditional *GAN (cGAN)* [15] architecture. The discriminator, *D*, learns to classify fake (predicted semantic map) and real (ground truth map). Both generator and discriminator observe the same 2D range imagery input.

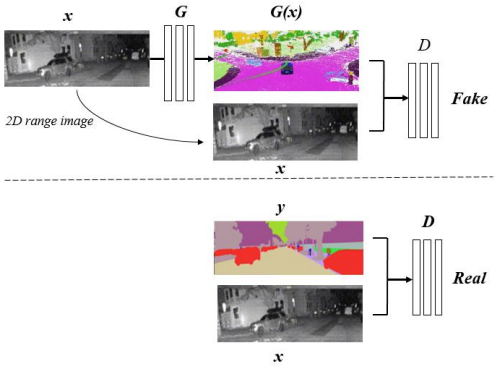

Fig.5. Conditional GAN training: map 2D range imagery to segmentation map.

There are three lost terms, the first term is the general cross-entropy term for segmentation network (generator), $S(\cdot)$, to predict each location (pixel-wise) of the output map with independent class label. It is a weighted cross-entropy loss as is expressed as.

$$L_{wce} = -\sum_{c \in C} \frac{1}{\sqrt{f_c}} Y_c log(S_c) \quad (2)$$

where *Y* and *S* are the one-hot vector maps for ground truth and predicted label, respectively. Due to the imbalance data nature of the street scene, pedestrians and cyclists are less seen compared to other cars, the way to mitigate the network biases toward to the classes with higher frequency of occurrence is to add a weighted factor *f*. The second

term is the *Lovász -Softmax* loss [16]. The loss is used to improve the intersection-of-union (IoU) or *Jaccard index*. The convex *Lovász* extension of submodular losses relaxes the IoU hypercube constraint where each vertex is a plausible combination of the class labels. Therefore, IoU score can be defined anywhere inside the hypercube. This term is expressed as

$$L_{ls} = \frac{1}{|C|} \sum_{c \in C} \Delta J_C(m(c)) \quad (3)$$
$$m_i(c) = 1 - x_i(c) \; if \quad c = y_i(c)$$
$$= x_i(c), \; otherwise$$

where $x_i(c) \in [0,1]$ is the pixel-wise predicted probability, $y_i(c)$ is the predicted label. The loss will penalize the wrong prediction.

The third term is the adversarial loss which can be expressed as

$$L_{adv}(G, D) = E_{x,P_{gt}}[logD(x, y)] + E_{x,P_p}[log\big(1 - D\big(x, G(x, z)\big)\big], \quad (4)$$

where D is the discriminator which produces *Real* and *Fake* binary outputs, and G generates the predicted label, x is the 2D range image, z is the optional random noise input, P_{gt} is the distribution of ground truth label, y, and P_p is the distribution of the predicted label.

D tries to maximize the *Jensen-Shannon* divergence [1] between P_{gt} and P_p. On the contrary, G tries to minimize the same distribution divergence in order to make P_p indistinguishable from P_{gt}. The final objective is a mix-max optimization of the loss summation of cross entropy, *Lovász -Softmax* and adversarial terms as shown in Equation (5)

$$G^* = arg \min_G \max_D \; L_{adv} + L_{ls} + L_{ce} \quad (5)$$

4 Experiments

Semantic KITTI data set [17] was used for algorithm evaluation. The dataset contains 28 classes including classes of non-moving and moving objects. The scanned sequences of 0-10 except 8 were used for training, and sequence 8 was used for validation. Sequences 11-21 was used for testing, however, the annotations for the testing sequence are not available to the general public. In order to evaluate the performance, the labeled data were submitted to Semantic KITTI official server for test results. The evaluation metric is based on *Jaccard Index* or mean Intersection-over-Union (IoU) metric as shown in the Equation (6).

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c} \quad (6)$$

where *TP*, *FP*, and *FN* correspond to the number of true positive, false positive, and false negative predictions for class *c*, and *C* is the number of classes.

A. Quantitative Results

Table 1. Quantitative results comparison on SemanticKITTI testing set (Sequence 11-21).

	Approach	car	bicycle	motor-cycle	truck	other-vehicle	person	bicyclist	motor-cyclist	road	parking	sidewalk	Other-ground	building	fence	vege-tation	trunk	terrain	Pole	Traffic-sign	mean-IOU
Point-wise	Pointnet	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.4
	Pointnet++	53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1
	RandLA-Net	**94.2**	26.0	25.8	40.1	**38.9**	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	**66.8**	49.2	47.1	53.9
	LatticeNet	88.6	12.0	20.8	**43.3**	24.8	34.2	39.9	**60.9**	88.8	64.6	73.8	25.5	86.9	55.2	76.4	**67.9**	54.7	41.5	42.7	52.2
Projection-based	SqueezeSegV2	81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7
	SqueezeSegV2-CRF	82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6
	RangeNet++	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	**91.8**	**65.0**	**75.2**	**27.8**	87.4	58.6	80.5	55.1	64.6	47.9	55.9	52.2
	SqueezeSegV3	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	**89.0**	59.4	**82.0**	58.7	65.4	49.6	58.9	55.9
	SalsaNext (train by ourself)	91.1	43.7	34.3	37.5	29.9	**59.1**	53.6	30.4	90.9	60.4	73.2	23.6	87.6	56.6	79.5	58.6	63.9	51.3	62.0	57.2
	SalsaNext +Discriminator	91.1	46.6	32.0	37.1	31.8	57.5	55.0	26.2	91.0	61.8	74.7	25.7	87.9	57.6	80.1	61.1	66.2	**54.7**	**62.6**	57.9
	MamboNet	92.0	**47.4**	**39.0**	25.6	34.6	59.0	**57.6**	27.8	**91.8**	64.9	75.0	21.3	88.8	**60.5**	81.3	64.7	64.7	54.6	61.2	**58.5**

In Table 1, our method not only out performs most of the 3D point-wise methods [18,19,20,21], but also is superior to other projection based methods, especially in small object segmentation, such as person, bicyclist, and motor-cyclist categories. We compare our method with two other networks, the first one is the SalsaNext baseline [9], and the second one is the SalsaNext augmented with a discriminator. The discriminator is a VGG-based convolutional network. In the beginning, we trained the SalsaNext baseline using their open source Github repository, and the test result of mIOU is 57.2, which is a little lower than the published result (59.5) [9]. The discrepancy can be due to the limited batch size [15] in our single board training configuration. In Table I, SalsaNext with discriminator outperforms baseline in 15 out of 19 categories, and the mIOU of 57.9 is slightly improved. Our method, MamboNet, achieves over one percent mOUT improvement of 58.5. .

B. Qualitative Results

In Fig. 6, four blocks of segmented map results are shown for visual examination. Each block has three maps, the top is the SalsaNext baseline, the middle one is our

method with adversarial discriminator, and the bottom one is the ground true for comparison.

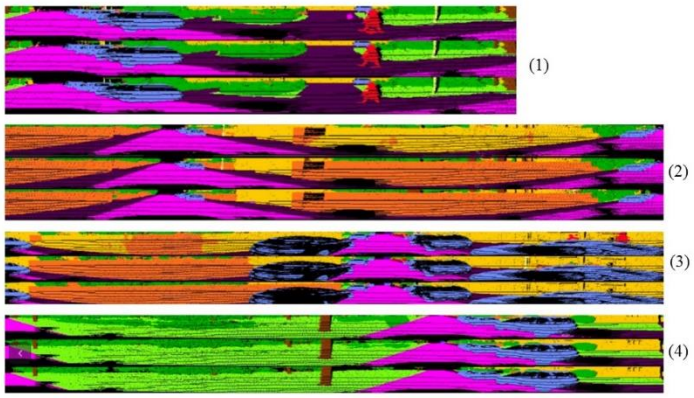

Fig. 6. Qualitative evaluation with four examples: the top strip of each example is the result without adversary training, the middle strip is with adversary training, and the bottom strip is the ground-truth label.

In the top strip of the first example, there is a small mis-classified pink circle inside the dark purple region (road). The middle strip of the same example, the circle disappears due to the discriminator power of enforcing regional consistency. The same rectification can be observed in the second and third examples, all middle strips correctly identify the fence region (brown), while the top strip mis-classify part of the fence to be the building regions (yellow). Finally in the fourth example, the top strip also misclassifies portion of light green (terrain) to be dark green (vegetation), however, the middle strip correctly identifies the terrain area.

5 Conclusion

We augmented an encoder-decoder segmentation network with an adversarial network to improve the semantic segmentation performance. Adversarial network can implicitly enforce the regional contextual continuity. Unlike conventional CRF and KNN post processing techniques, the adversarial is learnt only during the offline training and is not active during the test. Therefore, the online computation is greatly reduced and yet the comparable results are still attainable.

1 References

1. I, J, Goodfellow, J. P-Abadie, M. Mirza, B. Xu, D. W-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, in NIPS, 2014.
2. O. Ronneberger, P.Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in Medical Image Computing and Computer-Assisted Intervention, 2015..
3. A. Kendall, V. Badrinarayanan and R. Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding., Proceedings of the British Machine Vision Conference (BMVC), pages 57.1-57.12. BMVA Press, 2017.
4. B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," ICRA, 2018.
5. B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud, in ICRA, 2019.
6. F. Yu, V, Koltun "Multi-Scale Context Aggregation by Dilated Convolutions", ICLR, 2016
7. C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation, arXiv:2004.01803, 2020.
8. A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, RangeNet++: Fast and Accurate LiDAR Semantic Segmentation, in IROS, 2019.
9. Cortinhal, T., Tzelepis, G., Aksoy, E.E., SalsaNext: Fast, uncertainty-aware se- mantic segmentationof LiDAR point clouds for autonomous driving, arXiv:2003.03653, 2020.
10. P. Luc, C. Couprie, S. Chintala, J. Verbeek, Semantic Segmentation using Adversarial Networks, 2016, Workshop on Adversarial Training, in NIPS 2016.
11. N. Souly, C. Spampinato, M. Shah, Semi Supervised Semantic Segmentation Using Generative Adversarial Network, ICCV, 2017.
12. A. Bochkovskiy, C.-Y. Wang, H. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004:10934, 2020.
13. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D, Erhan, V. Vanhoucke, A, Robinvich, Going deeper with convolutions, in CVPR 2015.
14. W. Shi, J. Caballero, F. Huszár , J. Totz. A. Aitken, R. Bishop, Z. Wang, Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, in CVPR 2016.
15. P. Isola, J-Y. Zhu, T. Zhou, A. A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, in CVPR, 2017.
16. M. Berman, A. Rannen Triki, and M. B. Blaschko, The Lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, in CVPR, 2018.
17. J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences, Proceedings of International Conference on Computer Vision, Seoul, Korea, pp. 1-17, 2019.
18. Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, Randlanet: Efficient semantic segmentation of largescale point clouds, in CVPR, 2020
19. C. R. Qi, H. Su, K. Mo, and L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in CVPR, 2017.
20. C. R. Qi, L. Yi, H. Su, and L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in NIPS, 2017.
    R. Alexandru Rosu, P. Schu¨tt, J. Quenzel, and S. Behnke, LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices, arXiv:1912.05905, 2019.