# Bird's-Eye-View-Based LiDAR Point Cloud Coding For Machines

Xiang Gao, Zhiwei Zhu, Lu Yu*
Zhejiang University, Hangzhou, China
{xiangg, zhuzhiwei21, yul}@zju.edu.cn

2023 IEEE International Conference on Visual Communications and Image Processing (VCIP) | 979-8-3503-5985-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/VCIP59821.2023.10402658

***Abstract*—Recently, there has been a growing interest in image/video coding tailored for intelligent analysis tasks, such as Image/Video Coding for Machines (ICM/VCM). These approaches have shown remarkable results when compared to human perception-based coding methods. However, point cloud coding methods for machine intelligence have not been extensively studied. Inspired by current LiDAR point cloud intelligent analysis methods which convert point cloud into bird's-eye-view (BEV) perspective, we propose an end-to-end learnt point cloud coding framework for 3D machine intelligent tasks with BEV representation, named PC4M. Specifically, the PC4M system consists of a LiDAR encoder, a learnt BEV feature codec and a BEV region proposal network with a task-specific head. To achieve better rate-distortion performance for analysis tasks, we propose an efficient Res-NeXt fusion block with powerful multi-scale modeling ability to compress sparse BEV features, and design a long-distance adaptive attention module by using VanAtten block. Experimental results demonstrate that our method outperforms the state-of-the-art MPEG standard Geometry-based Point Cloud Coding (G-PCC) on the object detection and BEV map segmentation by 83.12% and 85.32% of BD-rate gain on nuScenes, respectively. To the best of our knowledge, this is the first end-to-end learnt task-oriented point cloud codec.**

***Index Terms*—Point cloud coding for machine, Point cloud compression, Bird's-eye-view representation, Multi-task learning.**

## I. Introduction

Over the recent years, point clouds have become a significant data format for providing high-accuracy depth information of objects for autonomous driving systems. However, huge volume of data generated by LiDAR remains a challenge for data communication. Efficient compression of LiDAR point clouds to meet storage and bandwidth requirements can improve the efficiency of autonomous driving systems.

Traditional point cloud coding (PCC) algorithms have rapidly developed under the efforts of the MPEG expert group [1]. They are mainly divided into two typical PCC architectures: Geometry-based PCC methods (G-PCC) which compresses point cloud by utilizing the octree structure and Video-based PCC methods (V-PCC) which compresses the projected plane from point cloud using a video codec. Furthermore, increasing studies have begun to focus on learnt PCC methods. Due to the sparsity and irregularity of point clouds, these methods change the original 3D representation of point clouds, such as regular voxels [2], [3], octree [4]–[6], or depth image [7], [8], etc. They all aim to maintain the signal fidelity of reconstructed point clouds as the compression target without considering joint optimization for downstream tasks which make these decoded point clouds bear obvious perception accuracy loss in the tasks with low bitrate.

The rapid development of deep learning has led to breakthroughs in autonomous driving. The intelligent perception system of a single vehicle converts LiDAR point cloud into bird's-eye-view (BEV) representation, demonstrating real-time and high-precision sensing ability, giving rise to various perception tasks under BEV, such as BEV object detection and tracking [9]–[12], and BEV map segmentation [12]–[14]. Besides, in order to compensate for the limitation of single-vehicle perception range, collaborative perception methods [15]–[17] improve the perception quality by transmitting the intermediate BEV feature of point clouds between vehicles.

Recently, increasing research has been devoted to task-oriented image and video compression. They compress image/video for intelligent analysis tasks with joint optimization. Existing Image/Video Coding for Machine (ICM/VCM) methods either compress image/video in their original domain [18], [19] for specific task or compress intermediate feature representation of image/video in feature domain [20]–[22] for task analysis. Currently, compressing intermediate feature methods become the mainsteam branch of ICM/VCM schemes which take task loss and entropy loss as optimized target and outperform the state-of-the-art Versatile Video Coding (VVC) standard on intelligent analysis tasks.

Motivated by ICM/VCM, we present a novel point cloud coding framework optimized for machine analysis tasks based on BEV representation, as shown in Fig. 1. We propose an intelligent feature codec with BEV feature as compressed object, and design a Res-NeXt fusion block and VanAtten block specially for sparse BEV feature whose coding efficiency is obviously better than that of ordinary image codec.

This paper has the following major contributions:

- We establish a novel research direction and propose an end-to-end learnt point cloud codec for machine intelligence based on bird's-eye-view representation.
- We have confirmed that considering machine intelligence accuracy in optimizing coding schemes can significantly improve coding efficiency with task accuracy.
- We propose an intelligent feature codec by utilizing spatially sparse convolutional kernels with large receptive fields, which effectively adapt to the spatial sparsity characteristic of BEV features, resulting in a significant improvement in coding efficiency.

The authors are with the College of Information Science and Electronic Engineering, Zhejiang University, and also with the Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking (IPCAN), Hangzhou 310027, China. *Corresponding author: Lu Yu.

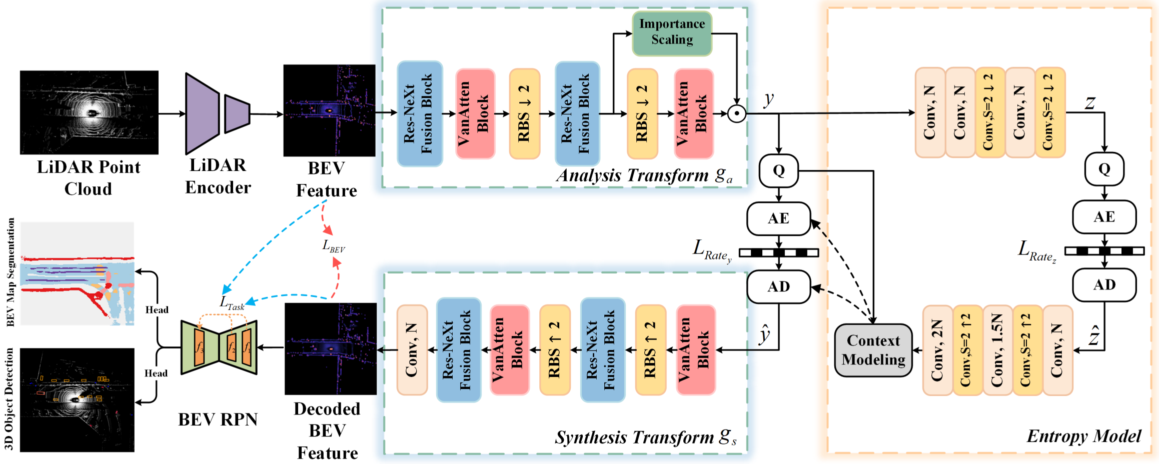

Fig. 1: The proposed LiDAR Point Cloud Coding For Machines (**PC4M**) architecture. Q refers to quantization. AE and AD are arithmetic encoding and decoding, respectively. ↑ 2 and ↓ 2 represent the up/down-sample operation. ⊙ represents the dot multiplication operation.

## II. Related Work

ICM/VCM methods convert image/video into an unified intermediate feature representation and utilze learnt image codec to compress the feature for intelligent analysis tasks. Therefore, we rethink current BEV-based point cloud analysis and learnt image compression (LIC) methods.

**BEV-based point cloud analysis:** The BEV representation has gradually become a natural and simple candidate view as a unified representation in the field of autonomous driving [23]. Various LiDAR object detection methods firstly convert point cloud into regular grids such as 3D voxels [24] or 3D pillars [9]. And then, sparse 3D convolution/standard 2D convolution is used respectively to aggregate local voxel/pillar features and fuse height channel information, finally flatten 3D features into the 2D BEV space. BEV representation can easily identify problems such as object blocking and cross-traffic and serve as an unified feature space to complete multiple downstream tasks [12], such as object detection and map segmentation.

**Learnt image compression:** LIC utilizing auto-encoder structures has shown remarkable progress and performance. Ballé [25] proposed an end-to-end CNN-based compression model, followed by a VAE architecture and the incorporation of hyperpriors for improved compression [26], and [27] utilized a local context model to enhanced the entropy model. Building upon these frameworks, various approaches have emerged to enhance the backbone of image compression. Cheng [28] introduced residual networks and attention modules, while [29] employed multi-scale residual blocks and a content-adaptive bit allocation strategy using an importance scaling map. Some studies utilized Vision Transformers (ViT) [30] to extract global context information. In contrast, TCM [31] incorporated the local modeling ability of CNNs with the non-local modeling ability of transformers, achieving impressive coding performance.

LiDAR point cloud analysis methods transform the point cloud into image-like 2D BEV features. It becomes possible to leverage the existing learnt image compression techniques. In our work, we organically combine both above approaches to realize the PC4M framework.

## III. Methodology

### A. *Architecture*

The overall PC4M framework we proposed is shown in Fig. 1, which comprises a LiDAR Encoder, an end-to-end BEV codec, and a BEV Region Proposal Network (RPN) with specific Head for downstream tasks. The LiDAR point cloud $\mathcal{P} \in \mathbb{R}^{N \times F_{in}}$ is fed into the LiDAR Encoder, where $N$ denotes the size and $F_{in}$ represents the coordinates and corresponding attributes of the point cloud. It voxelizes the point cloud and gradually extracts the BEV features $\mathcal{F}_{BEV} \in \mathbb{R}^{C \times H \times W}$ by using 3D sparse convolution and height flattening. The BEV codec are responsible for compressing and reconstructing the BEV representation. The BEV RPN further extracts more detailed features from the decoded BEV features. The task-specific heads generate final prediction results for each specific task. In this paper, we focus on two downstream tasks: 3D object detection and BEV map segmentation, to evaluate the point cloud compression efficiency for machine analysis.

Our end-to-end BEV codec consists of three sub-models: Analysis Transform $g_a$, Entropy Model and Synthesis Transform $g_s$. The $g_a$ learns the compact latent representation $\boldsymbol{y}$ of the input BEV feature, the Entropy Model learns the side information $\boldsymbol{z}$ to capture spatial dependencies among the elements of $\boldsymbol{y}$, and estimates the probability distribution parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ for arithmetic coding and decoding. The $g_s$ outputs the reconstructed BEV features through the decoded potential representation from bitstreams.

Inspired by the feature fusion method of TCM [31] and MSRB [29], we employ two stages of an advanced Res-NeXt fusion block in the $g_a$ and $g_s$ network to enhance the

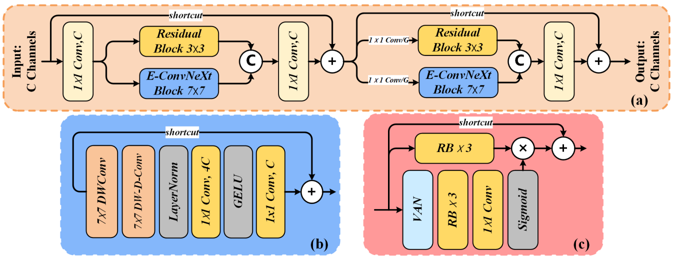

Fig. 2: (a) Res-NeXt fusion block. (b) The enhanced ConvNeXt block. (c) VanAtten block.

compression performance. The detailed structure of the Res-NeXt fusion block is illustrated in Fig. 2 (a). Specifically, the input BEV feature $\mathcal{F}_{BEV}$ is firstly input into a $1\times1$ convolutional layer which outputs two channel branches. One is passed through the residual block to generate local features $\mathcal{F}_{resnet}$, the other is fed into the enhanced ConvNeXt block to extract larger-scale features and inter-channel relationships, resulting in $\mathcal{F}_{convnext}$. Subsequently, these two features are concatenated into a new feature $\mathcal{F}'_{BEV}$. It is then passed through a $1\times1$ convolutional layer to reorganize the channels and added with a shortcut output. Afterwards, we employ two $1\times1$ convolutional layers with generalized divisive normalization (GDN) [25] operators to split $\mathcal{F}'_{BEV}$ into two branches again, and repeat the previous operation to finally output an representation that fully integrates multi-scale features.

ConvNeXt [32] are constructed entirely from standard ConvNet modules and outperforms Transformers in terms of both accuracy and scalability. In the Res-NeXt fusion block, we enhanced the ConvNeXt block by incorporating $7\times7$ depth-wise convolution with dilation 3 (DW-D-Conv) to expand the receptive field, as depicted in Fig. 2 (b). Additionally, we design a VanAtten block by introducing the VAN [33], illustrated in Fig. 2 (c). VAN decomposes large kernel convolution into a depth-wise convolution (DW-Conv), depth-wise dilatation convolution (DW-D-Conv) and point-wise convolution ($1\times1$ Conv), reducing the computational cost and obtaining spatial and channel adaptability. The detail of VAN is presented in Fig. 3, we adopt $5\times5$ DW-Conv and DW-D-Conv with dilation 3. The designed VanAtten block can capture local information by residual blocks (RB) and enable long range correlations by VAN, making the network more capable of preserving downstream task perception accuracy with lower bitrate.

The input BEV features are gradually downsampled by residual block with stride (RBS) of 2, we utilize two layers of spatial downsampling in $g_a$. At the end of $g_a$, we utilize the importance scaling method [29] to achieve content-adaptive bit allocation, which makes the model focus more on the crucial regions for machine analysis. The $g_s$ is symmetrical to the $g_a$, and two upsampling layers are used to gradually restore the original BEV features.

### *B. Rate-distortion optimization combined with task analysis*

Rate-distortion optimization (RDO) applied in learnt image and point cloud compression typically considers both entropy rate and reconstructed image distortion. Given a BEV feature $\boldsymbol{x}$ and a set of analysis transform $g_a$ and synthesis transform $g_s$, the BEV feature codec can be formulated by:

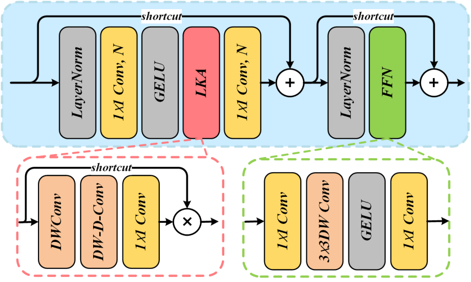

Fig. 3: The detail of VAN module.

$$\begin{aligned}\boldsymbol{y} &= g_a(\boldsymbol{x};\boldsymbol{\phi})\\ \hat{\boldsymbol{y}} &= Q(\boldsymbol{y}-\boldsymbol{\mu})+\boldsymbol{\mu}\\ \hat{\boldsymbol{x}} &= g_s(\hat{\boldsymbol{y}};\boldsymbol{\theta})\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ here represent the raw BEV feature and decompressed BEV feature, respectively. $\boldsymbol{y}$ is the latent representation of BEV feature, $\hat{\boldsymbol{y}}$ is the quantized latent.

We aim to strike a balance between bitrate and accuracy of machine analysis. To further preserve the semantic information in a deeper feature level, the original feature $\boldsymbol{x}$ and its reconstructed one $\hat{\boldsymbol{x}}$ are both fedding into BEV RPN network, obtaining multi-layers features $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ before specific Head. And then, we calculate task feature loss $\mathcal{L}_{task}$ utilizing Mean Square Error (MSE) loss between those multi-layers features of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. By the way, we also calculate $\mathcal{L}_{BEV}$ which denotes MSE distortion between original and reconstructed BEV features. Our particular loss function is defined as:

$$\begin{aligned}\mathcal{L} =& \underbrace{\lambda\cdot\mathcal{R}(\hat{\boldsymbol{y}})+\lambda\cdot\mathcal{R}(\hat{\boldsymbol{z}})}_{\mathcal{L}_{rate}}+\underbrace{\alpha\cdot\mathcal{D}(\boldsymbol{x},\hat{\boldsymbol{x}})}_{\mathcal{L}_{BEV}}+\underbrace{\sum_{i=1}^{3}\omega_i\cdot\mathcal{D}(\boldsymbol{f_i},\hat{\boldsymbol{f}}_{\boldsymbol{i}})}_{\mathcal{L}_{task}}\\ =&\lambda\cdot\mathbb{E}\left[-\log_2\left(p_{\hat{\boldsymbol{y}}\mid\hat{\boldsymbol{z}}}(\hat{\boldsymbol{y}}\mid\hat{\boldsymbol{z}})\right)\right]+\lambda\cdot\mathbb{E}\left[-\log_2\left(p_{\hat{\boldsymbol{z}}\mid\boldsymbol{\psi}}(\hat{\boldsymbol{z}}\mid\boldsymbol{\psi})\right)\right]\\ &+\alpha\cdot\mathcal{D}(\boldsymbol{x},\hat{\boldsymbol{x}})+\sum_{i=1}^{3}\omega_i\cdot\mathcal{D}(\boldsymbol{f_i},\hat{\boldsymbol{f}}_{\boldsymbol{i}}),\end{aligned} \tag{2}$$

where $\lambda$ controls the rate-distortion tradeoff. $\alpha$ and $\omega$ are both hyperparameters which control the importance of different distortions, and the number of layers is 3.

## IV. Experiment

### *A. Dataset and Settings*

To evaluate the compression ability and quality with different tasks, we choose nuScenes as training and testing dataset. The nuScenes [34] dataset is a large-scale autonomous driving benchmark supporting for object detection and BEV map segmentation including 1,000 driving scenes in total, which are split into 700, 150 and 150 scenes for training, validation and testing, respectively.

**Model details** We initialize LiDAR Encoder, BEV RPN and detection/segmentation heads using pretrained Bevfusion-L [12] models. We voxelize the LiDAR point cloud with 0.075m for detection, and 0.1m for segmentation on nuScenes.

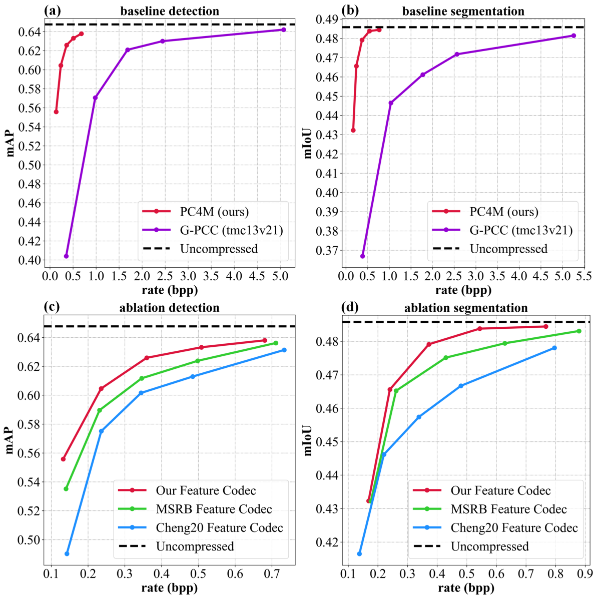

Fig. 4: Baseline curves and ablation on feature codec.

**Training details** We train the BEV feature codec with fixed parameters of other models in the PC4M framework. Optimization is carried out using AdamW [35] with a batch size of 8. The learning rate is maintained at a fixed value of $1 \times 10^{-4}$ during the training process, and is reduced to $1 \times 10^{-5}$ for the last 5 epochs. $\lambda$ in Eq. 2 belongs to the set $\{0.005, 0.01, 0.02, 0.04, 0.08\}$, $\alpha$ is experimentally set to 0.5, $\omega_1$, $\omega_2$, $\omega_3$ are set to 0.25, 0.25 and 0.5, respectively. All the methods are trained on 8 RTX 3090 GPUs.

**Evaluate setting** For 3D Object Detection, we use the mean average precision (mAP) [36] across 10 foreground classes on nuScenes as our detection perception metrics. For BEV Map Segmentation, we report mean Intersection-over-Union (mIoU) [37] of 6 background classes as our segmentation perception metric.

### *B. Baseline Experimental Results*

Fig. 4 (a) and (b) report our rate-performance curve result, compared with G-PCC (tmc13v21) [38] anchor on 3D object detection and BEV map segmentation. Specific results are illustrated in Table I, the PC4M method achieves 83.12% BD-rate gain on detection. Meanwhile, after using channel split in Res-NeXt fusion block, we can save a lot of parameters and FLOPs, while we can still get a comparable BD-rate. Baseline results prove that the efficiency of PC4M optimized with machine analysis is significantly higher than that of traditional LiDAR point cloud codec.

### *C. Ablation Study*

**Feature codec ablation** We implement ablation study on the BEV feature codec, compared with the learnt image compression methods applied into our PC4M system including Cheng20 [28], MSRB [29], as shown in Fig. 4 (c) and (d). Our BEV feature codec outperforms Cheng20 by 59.53% and 50.75% detection and segmentation BD-rate gains, respectively. MSRB using multi-scale residual blocks achieves better performance than Cheng20, ours on average saves 27.54% and 16.00% bitrate for the same task performance level on detection and segmentation than MSRB, respectively. As we can see, our feature codec specially designed for sparse BEV feature can achieve much better coding performance.

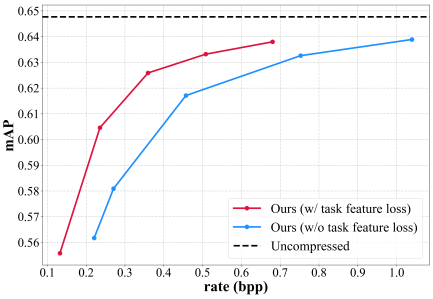

Fig. 5: Ablation study on **task feature loss**.

**Loss function ablation** We implement ablation study on 3D object detection task using our designed loss function, Fig. 5 illustrates the result that validate the contribution of the task feature loss. The **w/o task feature loss** represents that we remove the task feature loss in Eq. 2. The inclusion of the task loss in the compression process leads to a 57.93% BD-rate gain compared to the absence of the task loss. This result indicates that the distortion of multi-layer features can significantly contribute to preserving downstream task information.

TABLE I: Bjøntegaard Delta rate (BD-rate) with respect to detection performance against G-PCC anchor as in Figure 4.

| Methods | BD-rate | Parameters(/M) | FLOPs(/G) |
|---|---|---|---|
| Cheng20 [28] | -74.66 | 34.03 | 221.52 |
| MSRB [29] | -79.18 | 42.43 | 402.97 |
| Ours (w/ channel split) | -82.09 | 44.94 | 450.25 |
| Ours (w/o channel split) | **-83.12** | 62.43 | 852.82 |

## V. Conclusions

In this paper, we proposed the first end-to-end learnt point cloud coding for machines, named as PC4M. We transform the point cloud into BEV representation, utilizing a learnt feature coding method to compress the BEV feature. We proposed a Res-NeXt fusion block and introduced a VanAtten block to the feature codec for achieving better compression performance. Finally, we introduced a special loss function that combines machine intelligence tasks to jointly optimize the BEV feature codec. Our PC4M method totally outperforms traditional G-PCC method and our feature codec also shows outstanding performance than learning-based approaches. In the future, we will continue to explore the potential of point cloud coding schemes for machine intelligence.

## VI. Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant 62071427, Key Research and Development Program of Zhejiang Province, China, under Grant 2021C01119 and National Natural Science Foundation of China under Grant No.U21B2004.

# References

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[2] J. Wang, H. Zhu, H. Liu, and Z. Ma, "Lossy point cloud geometry compression via end-to-end learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4909–4923, 2021.

[3] J. Wang, D. Ding, Z. Li, and Z. Ma, "Multiscale point cloud geometry compression," in *2021 Data Compression Conference (DCC)*. IEEE, 2021, pp. 73–82.

[4] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for lidar compression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1313–1323.

[5] Z. Que, G. Lu, and D. Xu, "Voxelcontext-net: An octree based framework for point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6042–6051.

[6] C. Fu, G. Li, R. Song, W. Gao, and S. Liu, "Octattention: Octree-based large-scale contexts model for point cloud compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 625–633.

[7] S. Wang, J. Jiao, P. Cai, and L. Wang, "R-pcc: a baseline for range image-based point cloud compression," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 055–10 061.

[8] L. Zhao, K.-K. Ma, Z. Liu, Q. Yin, and J. Chen, "Real-time scene-aware lidar point cloud compression using semantic prior representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5623–5637, 2022.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[10] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[11] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1090–1099.

[12] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[13] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 760–13 769.

[14] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 2022, pp. 1–18.

[15] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 605–621.

[16] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, pp. 107–124.

[17] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers," in *6th Annual Conference on Robot Learning*, 2022.

[18] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: an end-to-end learned approach," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1590–1594.

[19] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, H. R. Tavakoli, and E. Rahtu, "Learned image coding for machines: A content-adaptive approach," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[20] S. Ma, X. Zhang, S. Wang, X. Zhang, C. Jia, and S. Wang, "Joint feature and texture coding: Toward smart video representation via front-end intelligence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3095–3105, 2018.

[21] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3349–3353.

[22] R. Feng, X. Jin, Z. Guo, R. Feng, Y. Gao, T. He, Z. Zhang, S. Sun, and Z. Chen, "Image coding for machines with omnipotent feature learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer, 2022, pp. 510–528.

[23] H. Li, C. Sima, J. Dai, W. Wang, L. Lu, H. Wang, E. Xie, Z. Li, H. Deng, H. Tian *et al.*, "Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe," *arXiv preprint arXiv:2209.05324*, 2022.

[24] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[25] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2016.

[26] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[27] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in neural information processing systems*, vol. 31, 2018.

[28] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7939–7948.

[29] H. Fu, F. Liang, J. Liang, B. Li, G. Zhang, and J. Han, "Asymmetric learned image compression with multi-scale residual block, importance scaling, and post-quantization filtering," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[31] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-cnn architectures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 388–14 397.

[32] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[33] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Computational Visual Media*, vol. 9, no. 4, pp. 733–752, 2023.

[34] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017.

[36] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[37] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[38] "MPEG-PCC-TMC13," http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc13, accessed: 2023.