BDC-Occ: Binarized Deep Convolution Unit For Binarized Occupancy Network

Zongkai Zhang, Peng Ling, Zidong Xu, Wenming Yang, *Senior Member, IEEE,* Qingmin Liao, *Senior Member, IEEE,* Jing-Hao Xue, *Senior Member, IEEE*

***Abstract*—Existing 3D occupancy networks demand significant hardware resources, hindering the deployment of resource-limited devices. Binarized Neural Networks (BNNs) offer a potential solution by substantially reducing computational and memory requirements. However, their performance decrease notably compared to full-precision networks. In addition, it is challenging to enhance the performance of the binarized model by increasing the number of binarized convolutional layers, which limits its practicability for 3D occupancy prediction. In this paper, we reconsider the components in binarized convolutional layers, and structures, for 3D occupancy prediction task. Two original insights into binarized convolution are presented, substantiated with theoretical proofs: (a) 1×1 binarized convolution introduces minimal binarization errors as the network deepens, and (b) binarized convolution is inferior to full-precision convolution in capturing cross-channel feature importance. Building on the above insights, we propose a novel binarized deep convolution (BDC) unit that significantly enhances performance, even when the number of binarized convolutional layers increases to meet the requirements of 3D occupancy networks. Specifically, in the BDC unit, additional binarized convolutional kernels are constrained to 1×1 to minimize the effects of binarization errors. Further, we propose a per-channel refinement branch to reweight the output via first-order approximation. Then, we partition the 3D occupancy networks into four distinct convolutional modules, employing BDC units to explore the effects of binarizing each of these modules. The proposed BDC unit minimizes binarization errors and improves perceptual capability, meeting the stringent requirements for accuracy and computational efficiency in 3D occupancy prediction. Extensive quantitative and qualitative experiments demonstrate that the proposed BDC unit achieves state-of-the-art performance in 3D occupancy prediction and 3D object detection tasks, while significantly reducing parameters and computational costs. This highlights the potential of the BDC unit as an efficient fundamental component in binarized 3D occupancy networks. Code for our paper will be released on "https://github.com/zzk785089755/BDC".**

***Index Terms*—3D occupancy prediction, binarized neural network, autonomous driving.**

I. Introduction

RECENT advancements in 3D occupancy prediction tasks have significantly impacted the fields of robotics [1]–[3] and autonomous driving [4]–[8], emphasizing the importance of accurate perception and prediction of voxel occupancy and semantic labels within 3D scenes. However, occupancy prediction requires predicting dense voxels, which leads to substantial computational expenses [9]–[13]. Moreover, the formidable performance of occupancy prediction models relies on increasing model size [14]. These factors collectively hinder the deployment of high-performance occupancy prediction networks on edge devices. For instance, Convolutional Neural Networks (CNN) [15]–[18] possess hardware-friendly and easily deployable characteristics. Moreover, CNN-based occupancy prediction networks [19] exhibit outstanding performance, making them the primary choice for deployment on edge devices. However, high-performance CNN-based occupancy networks [9], [14] often involve complex computations and numerous parameters. Therefore, it is necessary to introduce model compression techniques [20] to reduce the computational complexity and parameter count of CNN-based occupancy networks. Research on neural network compression and acceleration encompasses four fundamental methods: quantization [21], pruning [22], knowledge distillation [23], and lightweight network design [24]. Among these methods, Binarized Neural Networks (BNN), which fall under the quantization category, quantize the weights and activations of CNN to only 1 bit, leading to significant reductions in memory and computational costs. By quantizing both weights and activations to 1 bit, BNN [25] can achieve a memory compression ratio of 32× and a computational reduction of 64× when implemented on Central Processing Units (CPU). Furthermore, compared to full-precision models, BNN [25] only requires logical operations such as XNOR and bit counting, making them more easily deployable on edge devices. These characteristics of BNN are ideal for autonomous driving schemes, where the significantly reduced computational overhead allows more flexibility for downstream tasks. Nevertheless, current researches on BNN do not fully explore autonomous driving applications, such as 3D occupancy prediction. There is an urgent need to explore the capabilities of BNN in such task and provide a simple, strong, and extensible baseline for future researches and deployment.

Recent studies, such as BBCU [26], BiSRNet [27], JDB [28], BRVE [29] have demonstrated the capability of binarizing complex models with promising performance in tasks such as image super-resolution [30], denoising [31], ob-

This work was partly supported by the National Natural Science Foundation of China (No. 62171251) and the Special Foundations for the Development of Strategic Emerging Industries of Shenzhen (No. KJZD20231023094700001) *(Zongkai Zhang and Peng Ling contributed equally to this work.) (Corresponding author: Wenming Yang.)*

Zongkai Zhang, Peng Ling, Zidong Xu, Wenming Yang and Qingmin Liao are with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China. (email: zzk21@mails.tsinghua.edu.cn; lingp23@mails.tsinghua.edu.cn; xzd21@mails.tsinghua.edu.cn; yang.wenming@sz.tsinghua.edu.cn; liaoqm@tsinghua.edu.cn).

Jing-Hao Xue is with the Department of Statistical Science, University College London, London, WC1E 6BT, U.K. (email: jinghao.xue@ucl.ac.uk)

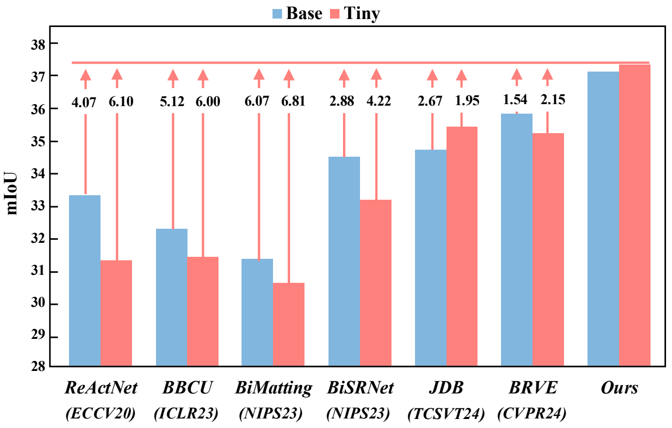


Fig. 1: **3D Occupancy Prediction Result.** Comparison between our BDC and state-of-the-art BNNs in the 3D occupancy prediction task [26]–[29], [32], [33]. Base means binarizing the BEV encoder and occupancy head, while Tiny means further binarizing the image neck based on Base.

ject detection, and video enhancement. We attempt to replace each full-precision convolutional unit in the 3D occupancy network with these binarized convolutional units. Such binarized models can achieve a respectable level of accuracy, but still exhibit a notable performance gap compared to the full-precision model. In full-precision models, it's common sense that increasing convolutional layers can lead to performance improvements. However, the binarized model did not exhibit a trend of performance improvement as the number of binarized convolutional layers increased. Instead, there is a tendency for performance to decline, making it challenging for binarized models to improve performance by increasing the number of convolutional layers [26]. This limitation significantly constrains the applicability of binarized occupancy networks, as they fail to enhance perception capabilities by increasing network depth, which is particularly detrimental for commonly used large-scale occupancy networks. Therefore, addressing the issues of decreasing accuracy with increasing binarized convolutional layers and limited perceptual capability is crucial for bridging the performance gap between binarized and full-precision 3D occupancy models.

To address these challenges, we propose a strong binarized deep convolution (BDC) unit, along with variants tailored for different components of 3D occupancy networks, marking the first exploration of binarized methods for 3D occupancy prediction. Our novel theoretical insights stem from two intrinsic properties of binarized convolution: (a) 1×1 binarized convolution introduces minimal binarization errors as the network deepens, and (b) binarized convolution is inferior to full-precision convolution in capturing cross-channel feature importance. Drawing on these insights, we restrict additional binarized convolutional kernels to 1×1 to reduce the impact of binarization errors as the 3D occupancy network depth increases. Secondly, we introduce a per-channel refinement branch that leverages newly added convolutional layers to narrow the gap with the output of full-precision convolution through first-order approximation. Integrating the two proposed techniques, we develop the Binarized Deep Convolution (BDC) unit, which remarkably enhances binarized model performance, despite the deepening of the binarized convolutional layers in 3D occupancy network. The 3D occupancy network is further decomposed into four fundamental modules, with customized binarization applied to each module using the BDC unit.

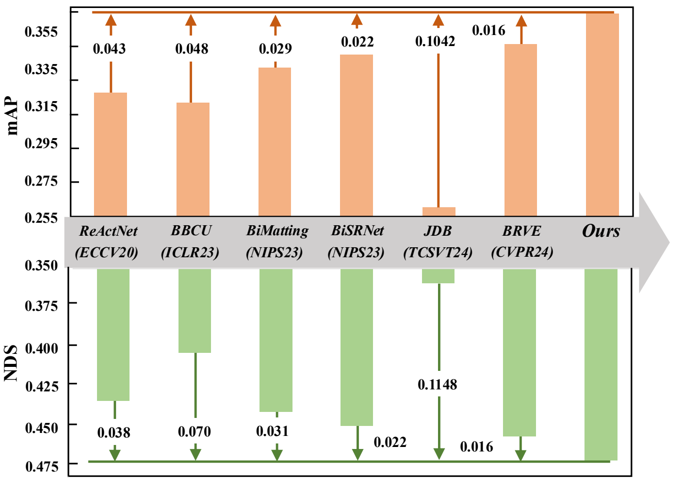


Fig. 2: **3D Object Detection Result.** Comparison between our BDC and state-of-the-art BNNs in the 3D object detection task.

The innovations and contributions of this paper are summarized as follows:

(i) Building on the original insights reinforced by theoretical proofs, we propose the **B**inarized **D**eep **C**onvolution (**BDC**) unit and further introduce a novel BNN-based 3D occupancy network. To our knowledge, this is the first paper to study the binarized 3D occupancy network.

(ii) In the BDC unit, additional binarized convolutional kernels are constrained to 1×1 to minimize the effects of binarization errors as the network depth increases. Subsequently, we propose a per-channel refinement branch to reweight the output via first-order approximation, thereby mitigating the limitations of binarized convolutional layers in assigning importance to features across channels. The 3D occupancy network is further decomposed into four fundamental modules, allowing for a customized design using the BDC unit.

(iii) The proposed BDC unit reduces binarization errors and enhances perceptual capability while considerably increasing computational efficiency, thus meeting the demanding requirements for accuracy and computational efficiency in 3D occupancy prediction. Extensive experiments on the Occ3D-nuScenes dataset demonstrate that our method achieves state-of-the-art (SOTA) mIOU, closely approaching that of full-precision models while utilizing only **52.26%** of the operations and **59.97%** of the parameters, and achieving a **21.06%** improvement in FPS.

II. Related Work

A. 3D Occupancy Prediction

The 3D occupancy prediction task comprises two sub-tasks: predicting the geometric occupancy status for each voxel in 3D space and assigning corresponding semantic labels.

We can categorize mainstream 3D occupancy networks into two architectures: CNN architecture based on the LSS [9], [34]–[39] method and Transformer architecture based on the BEVFormer [10], [40]–[45] method. Due to the deployment advantages of CNN models, this paper focuses on CNN-based 3D occupancy networks. MonoScene [9] is a pioneering work that utilizes a CNN framework to extract 2D features, which it then transforms into 3D representations. BEVDet-Occ [19] utilizes the LSS method to convert image features into BEV (Bird's Eye View) features and employs BEV pooling techniques to accelerate model inference. FlashOcc [36] replaces 3D convolutions in BEVDet-Occ with 2D convolutions and occupancy logits derived from 3D convolutions with channel-to-height transformations of BEV-level features obtained through 2D convolutions. SGN [37] adopts a dense-sparse-dense design and proposes hybrid guidance and efficient voxel aggregation to enhance intra-class feature separation and accelerate the convergence of semantic diffusion. InverseMatrixVT3D [38] introduces a new method based on projection matrices to construct local 3D feature volumes and global BEV features. Despite achieving impressive results, these CNN-based methods rely on powerful hardware with substantial computational and memory resources, which are impractical for edge devices. How to develop 3D occupancy prediction networks for resource-constrained devices remains underexplored. Our goal is to address this research gap.

B. Binarized Neural Network

BNN [25]–[29], [32], [33], [46]–[50] represents the most extreme form of model quantization, quantizing weights and activations to just 1 bit. Due to its significant effectiveness in memory and computational compression, BNN [25] finds wide application in both high-level vision and low-level vision. For instance, Xia et al. [26] designed a binarized convolutional unit, BBCU, for tasks such as image super-resolution, denoising, and reducing artifacts from JPEG compression. Cai et al. [27] devised a binarized convolutional unit, BiSR-Conv, capable of adjusting the density and distribution of representations for hyperspectral image (HSI) recovery. Beyond low-level static image restoration, recent studies also explore BNNs for raw video enhancement and 2D detection. Zhang et al. [29] propose a binarized framework that couples an easy-to-binarize spatio-temporal shift operator with a distribution-aware binary convolution, effectively aggregating temporal cues in RAW streams while narrowing the gap between binary and full-precision convolutions. For binary object detection, Xie et al. [28] design a joint-guided distillation scheme with dynamic channel-wise diversity enhancement, alleviating the representation degradation commonly seen in BNN detectors. However, the potential of BNN in 3D occupancy tasks remains unexplored. Hence, this paper explores binarized 3D occupancy networks, aiming to maintain high performance while minimizing computational and parameter overhead.

III. Method

A. Preliminary: Base 3D Occupancy Network

The full-precision models to be binarized should be lightweight and easy to deploy on edge devices. However, prior 3D occupancy network models based on CNNs [15] or Transformers [51], [52] have high computational complexity or large model sizes. Some of these works utilize complex operations such as deformable attention, which are challenging to binarize and deploy on edge devices. Therefore, we redesign a simple, lightweight, and deployable baseline model without using complex computational operations.

BEVDet-Occ [19] and FlashOcc [36] demonstrate outstanding performance in 3D occupancy prediction tasks using only lightweight CNN architectures. Inspired by these works, we adopt the network structure shown in Figure 3 as the full-precision baseline model. It consists of an image encoder $\mathcal{E}_{2D}$, a view transformer module $\mathcal{T}$, a BEV encoder $\mathcal{E}_{BEV}$, and an occupancy head $\mathcal{H}$. The occupancy prediction network is composed of these modules concatenated sequentially. Assuming the input images are $\mathbf{I} \in \mathbb{R}^{N_{view} \times 3 \times H \times W}$, the occupancy prediction output $\mathbf{O} \in \mathbb{R}^{X \times Y \times Z}$ can be formulated as

$$\mathbf{O} = \mathcal{H}(\mathcal{E}_{BEV}(\mathcal{T}(\mathcal{E}_{2D}(\mathbf{I})))) \tag{1}$$

where H and W represent the height and width of the input images, and X, Y, and Z denote the length, width, and height of the 3D space, respectively, N_{view} represents the number of multi-view cameras.

Specifically, multi-view images are first fed into the image encoder $\mathcal{E}_{2D}$ to obtain 2D features $\mathbf{f}_{2D}$ of shape $\mathbb{R}^{N_{\text{view}} \times C_{2D} \times H_{2D} \times W_{2D}}$ and depth prediction $\mathbf{f}_{\text{depth}}$ of shape $\mathbb{R}^{N_{\text{view}} \times N_{\text{depth}} \times H_{2D} \times W_{2D}}$, where C_{2D}, H_{2D}, W_{2D} denote the number of channels, height, and width of 2D features, respectively.

Subsequently, the image features $\mathbf{f}_{2D}$ and depth prediction $\mathbf{f}_{depth}$ are passed through the visual transformation module $\mathcal{T}$, which transforms them into primary BEV features $\mathbf{f}_{T} \in \mathbb{R}^{C_{BEV} \times H_{BEV} \times W_{BEV}}$ using camera intrinsic and extrinsic projection matrices. Here, C_{BEV} represents the number of channels of BEV features, while H_{BEV} and W_{BEV} represent the length and width of the BEV space, respectively. Since the voxel distribution obtained from the depth map through projection matrices is sparse, the representation capability of primary BEV features may be insufficient. To this end, $\mathbf{f}_{T}$ is passed through the BEV encoder $\mathcal{E}_{BEV}$ to obtain fine BEV features $\mathbf{f}_{BEV} \in \mathbb{R}^{C_{BEV} \times H_{BEV} \times W_{BEV}}$ for further refinement.

Finally, the semantic prediction output logits $\mathbf{O}_{logits} \in \mathbb{R}^{N_{class} \times X \times Y \times Z}$ come from the BEV features $\mathbf{f}_{BEV}$ processed through the occupancy prediction head $\mathcal{H}$, where N_{class} is the number of semantic classes in the dataset. By taking the index corresponding to the maximum value of the logits, we can obtain the final 3D occupancy prediction output $\mathbf{O}$.

B. Preliminary: Base Binarized Convolution Unit

To adjust the density and enable effective binarization of convolutional layers, Cai et al. [27] proposed the BiSR-Conv unit and the network, BiSRNet. We empoly BiSR-Conv unit as the initial version of BDC unit (**BDC-V0**) to binarize FlashOcc [36], with its structure shown in Figure 5 (a).

Specifically, we employ channel-wise feature redistribution in BDC-V0 to address the problem of significant differences in feature distribution:

$$\mathbf{X}_r = \kappa \cdot \mathbf{X}_f + b \tag{2}$$

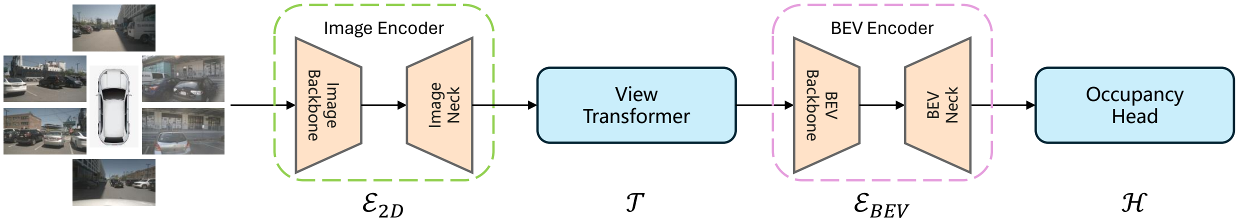


Fig. 3: CNN-based 3D Occupancy Network, consisting of an image encoder $\mathcal{E}_{2D}$, a view transformer $\mathcal{T}$, a BEV encoder $\mathcal{E}_{BEV}$, and an occupancy head $\mathcal{H}$. In our binarized variants, the BEV encoder and occupancy head are binarized (**-B**), and further binarizing the image neck yields the **tiny** version (**-T**). The image backbone in $\mathcal{E}_{2D}$ remains full-precision.

Here, $\mathbf{X}_r \in \mathbb{R}^{C\times H\times W}$ represents the activations after channel-wise feature redistribution. Both full-precision image features and Bird's Eye View (BEV) features, represented as $\mathbf{X}_f \in \mathbb{R}^{C\times H\times W}$, serve as input for the full-precision activations. κ represents the learnable density of redistribution, while b represents the learnable bias of redistribution.

Next, $\mathbf{X}_r$ is passed through the Sign function to binarize it, yielding 1-bit binarized activations $\mathbf{X}_b \in \mathbb{R}^{C\times H\times W}$, as follows:

$$x_b = \text{Sign}(x_r) = \begin{cases} +1, & \text{if } x_r > 0 \\ -1, & \text{if } x_r \le 0 \end{cases} \tag{3}$$

where $x_r \in \mathbf{X}_r$, $x_b \in \mathbf{X}_b$.

Since the Sign function is not differentiable, approximation functions are required to ensure successful backpropagation. Common approximation functions include piecewise linear function Clip$(\cdot)$, piecewise quadratic function Quad$(\cdot)$, and hyperbolic tangent function Tanh$(\cdot)$. We use the hyperbolic tangent function as the approximation function, defined as:

$$x_b = \text{Tanh}(\alpha x_r) = \frac{e^{\alpha x_r} - e^{-\alpha x_r}}{e^{\alpha x_r} + e^{-\alpha x_r}} \tag{4}$$

The Tanh function ensures gradients exist even when weights and activations exceed 1, allowing parameter updates downstream during backpropagation.

In the binarized convolutional layer, the 32-bit precision weights $\mathbf{W}_f$ are binarized into 1-bit binarized weights $\mathbf{W}_b$ according to the following formula:

$$w_b = \mathbb{E}_{w_f \in \mathbf{W}_f}(|w_f|) \cdot \text{Sign}(w_f) \tag{5}$$

Here, $\mathbb{E}_{w_f \in \mathbf{W}_f}(|w_f|)$ represents the average absolute value of the full-precision weights, which serves as a scaling factor to reduce the discrepancy between the binarized weights $\mathbf{W}_b$ and the full-precision weights $\mathbf{W}_f$. Multiplying this value by Sign$(w_f) = \pm 1$ yields element-wise binarized weights w_b.

Subsequently, the binarized activation $\mathbf{X}_b$ is convolved with the binarized weights $\mathbf{W}_b$. Binarized convolution can be accomplished purely through logical operations. The formulation of binarized convolution [48] is illustrated as follows:

$$\mathbf{Y}_b = \text{Biconv}(\mathbf{X}_b, \mathbf{W}_b) = \text{BitCount}(\text{XNOR}(\mathbf{X}_b, \mathbf{W}_b)) \tag{6}$$

Here, $\mathbf{Y}_b$ is the output of binarized convolution, Biconv denotes the binarized convolution layer, and BitCount and XNOR represent the bit count and logical XOR operations, respectively. In BDC-V0, the convolutional kernel size is 3×3.

For the activation function, we utilize RPReLU, whose expression is defined as follows:

$$\text{RPReLU}(y_i) = \begin{cases} y_i - \gamma_i + \zeta_i, & \text{if } y_i > \gamma_i \\ \beta_i \cdot (y_i - \gamma_i) + \zeta_i, & \text{if } y_i \le \gamma_i \end{cases} \tag{7}$$

Here, $y_i \in \mathbb{R}$ represents the i-th element value of $\mathbf{Y}_b$, and β_i, γ_i, and ζ_i are learnable parameters for the i-th channel.

C. Binarized Deep Convolution Unit Design

As discussed in Section I and Section II, increasing the depth of binarized convolutional units often leads to accuracy degradation, which limits their applicability in 3D occupancy prediction. To overcome this limitation, we propose the Binarized Deep Convolution (BDC) unit, designed to maintain or improve performance as the depth of binarized convolutional layers increases.

Theorem 1. *In the process of backpropagation, we denote the expected value of the element-wise absolute gradient error of the parameters* **w** *in the l-th binarized convolutional layer as $\mathbb{E}[\Delta\frac{\partial L}{\partial w_{mn}^{(l)}}]$. The specific expression is as follows.*

$$\mathbb{E}\left[\Delta\frac{\partial L}{\partial w_{mn}^{(l)}}\right] \le 0.5354 \cdot \Bigg(\sum_{i,j}\sum_{m'=-(k//2)}^{k//2}\sum_{n'=-(k//2)}^{k//2} \mathbb{E}\left[\left|\frac{\partial\sigma(y^{(l)}_{(i+m')(j+n')})}{\partial y^{(l)}_{ij}} \cdot w^{(l+1)}_{m'n'} \cdot \frac{\partial L}{\partial y^{(l+1)}_{ij}}\right|\right]\Bigg) \tag{8}$$

where k is the binarized convolution kernel size, $\frac{\partial\sigma(y^{(l)}_{(i+m')(j+n')})}{\partial y^{(l)}_{ij}}$ is the derivative of the activation function $\sigma(\cdot)$, $w^{(l+1)}_{m'n'}$ represents the weights of the binarized convolutional kernel in the next layer, and $\frac{\partial L}{\partial y^{(l+1)}_{ij}}$ is the element-wise gradient in the next layer.

Proof. *We assume the element of the input of a binarized convolutional layer as x_{ij}, with a binarization error denoted as ϵ_{ij}, the full-precision input before binarization as $\hat{x}_{ij}$, and the output of the binarized convolutional layer as y_{ij}. Thus, we have:*

$$x_{ij} = \hat{x}_{ij} + \epsilon_{ij} \tag{9}$$

Since the full-precision input $\hat{x}_{ij}$ at the current layer is the output from the batch normalization layer in the previous layer, we can assume that the full-precision input $\hat{x}_{ij}$ follows *a Gaussian distribution $\mathcal{N}(0,1)$. We can then derive the distribution of ϵ_{ij} as follows:*

$$\begin{aligned} |\epsilon_{ij}| &= |\hat{x}_{ij} - x_{ij}| \\ &= |\hat{x}_{ij} - \text{Sign}(\hat{x}_{ij})| = \begin{cases} |\hat{x}_{ij} - 1|, & \text{if } \hat{x}_{ij} > 0 \\ |\hat{x}_{ij} + 1|, & \text{if } \hat{x}_{ij} \leq 0 \end{cases} \end{aligned} \tag{10}$$

Assuming the convolution kernel size k is odd, the kernel weights w_{mn} and biases b_{mn} are defined for indices $m, n \in \{-\lfloor k/2 \rfloor, \ldots, \lfloor k/2 \rfloor\}$. The forward propagation for a $k \times k$ convolutional layer is expressed as:

$$y_{ij} = \sum_{m,n} (x_{(i+m)(j+n)} \cdot w_{mn} + b_{mn}) \tag{11}$$

Assuming that during backpropagation, the gradient at the current layer l is given by $h_{ij}^{(l)} = \frac{\partial L}{\partial y_{ij}^{(l)}}$, we can use the chain rule to derive the gradient for a $k \times k$ convolutional layer as follows:

$$\begin{aligned} \frac{\partial L}{\partial w_{mn}^{(l)}} &= \sum_{i,j} x_{(i+m)(j+n)}^{(l)} h_{ij}^{(l)} \\ &= \sum_{i,j} (\hat{x}_{(i+m)(j+n)}^{(l)} + \epsilon_{(i+m)(j+n)}^{(l)}) \cdot h_{ij}^{(l)} \end{aligned} \tag{12}$$

Given that the output of the current layer $y_{ij}^{(l)}$ becomes the input of the next layer after passing through the activation function $\sigma(\cdot)$. Based on Equation (11), we can derive:

$$y_{ij}^{(l+1)} = \sum_{m',n'} \sigma(y_{(i+m')(j+n')}^{(l)}) \cdot w_{m'n'}^{(l+1)} + b_{m'n'}^{(l+1)} \tag{13}$$

We can obtain the gradient relationship between $h_{ij}^{(l)}$ and $h_{ij}^{(l+1)}$:

$$\frac{\partial L}{\partial y_{ij}^{(l)}} = \sum_{m',n'} g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)} \tag{14}$$

where $g_{ijm'n'}^{(l)}$ denotes $\frac{\partial \sigma(y_{(i+m')(j+n')}^{(l)})}{\partial y_{ij}^{(l)}}$. By substituting Equation (14) into Equation (12), we can obtain:

$$\begin{aligned} \frac{\partial L}{\partial w_{mn}^{(l)}} = &\sum_{i,j,m',n'} (\hat{x}_{(i+m)(j+n)}^{(l)} + \epsilon_{(i+m)(j+n)}^{(l)}) \\ &\cdot g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)} \end{aligned} \tag{15}$$

We can derive the additional gradient error $\Delta \frac{\partial L}{\partial w_{mn}^{(l)}}$ induced by the binarization error ϵ as follows:

$$\begin{aligned} \Delta \frac{\partial L}{\partial w_{mn}^{(l)}} &:= |\frac{\partial L}{\partial w_{mn}^{(l)}} - \frac{\partial L}{\partial w_{mn}^{(l)}}|_{\epsilon=0}| \\ &= |\sum_{i,j,m',n'} \epsilon_{(i+m)(j+n)}^{(l)} \cdot g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)}| \\ &\leq \sum_{i,j,m',n'} |\epsilon_{(i+m)(j+n)}^{(l)} \cdot g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)}| \end{aligned} \tag{16}$$

By utilizing Equation (10), we can calculate the expected value of the absolute binarization error, denoted as $\mathbb{E}[|\epsilon_{ij}|]$:

$$\begin{aligned} \mathbb{E}[|\epsilon_{ij}|] &= \int_0^{\infty} |\hat{x}_{ij} - 1| \frac{1}{\sqrt{2\pi}} e^{-\frac{\hat{x}_{ij}^2}{2}} \, d\hat{x}_{ij} \\ &+ \int_{-\infty}^{0} |\hat{x}_{ij} + 1| \frac{1}{\sqrt{2\pi}} e^{-\frac{\hat{x}_{ij}^2}{2}} \, d\hat{x}_{ij} \\ &= \int_0^1 \frac{2 - 2\hat{x}_{ij}}{\sqrt{2\pi}} e^{-\frac{\hat{x}_{ij}^2}{2}} \, d\hat{x}_{ij} - \int_1^{\infty} \frac{2 - 2\hat{x}_{ij}}{\sqrt{2\pi}} e^{-\frac{\hat{x}_{ij}^2}{2}} \, d\hat{x}_{ij} \end{aligned} \tag{17}$$

The Gaussian error function, often abbreviated as "$erf(x)$" is defined as follows:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt \tag{18}$$

Based on the definition of the Gaussian error function and the use of the substitution rule, equation (17) can be written as follows:

$$\begin{aligned} \mathbb{E}[|\epsilon_{ij}|] = &\frac{2}{\sqrt{2\pi}} \{[\frac{\sqrt{\pi}}{\sqrt{2}}(erf(\frac{1}{\sqrt{2}}) - erf(\frac{0}{\sqrt{2}})) - e^{-\frac{0}{2}} + e^{-\frac{1}{2}}] \\ &- [(\frac{\sqrt{\pi}}{\sqrt{2}}(erf(\frac{\infty}{\sqrt{2}}) - erf(\frac{1}{\sqrt{2}})) - e^{-\frac{1}{2}} + e^{-\frac{\infty}{2}}]\} \\ &\overset{\substack{\text{erf(0) = 0} \\ \text{erf}(\infty)\text{ = 1}}}{=\!=\!=\!=} 2[erf(\frac{1}{\sqrt{2}}) - \frac{1}{2} - \frac{1}{\sqrt{2\pi}} + \frac{2}{\sqrt{2\pi e}}] \\ &\approx 0.5354 \end{aligned} \tag{19}$$

Therefore, based on Equations (16), the expected value of the additional gradient error $\mathbb{E}[\Delta \frac{\partial L}{\partial w_{mn}^{(l)}}]$ can be expressed as follows:

$$\begin{aligned} &\mathbb{E}[\Delta \frac{\partial L}{\partial w_{mn}^{(l)}}] \leq \\ &\qquad \sum_{i,j,m',n'} \mathbb{E}[|\epsilon_{(i+m)(j+n)}^{(l)} \cdot g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)}|] \end{aligned} \tag{20}$$

Based on Equation (10), since the binarization error $\epsilon_{ij}^{(l)}$ depends solely on the input magnitude $|x_{ij}^{(l)}|$ and is independent of any other variables, and given that the sign and the magnitude of $x_{ij}^{(l)}$ are independent [53]–[55], $\epsilon_{ij}^{(l)}$ and the other random variables in Equation (20) can be regarded as independent. Therefore, it follows that:

$$\begin{aligned} \mathbb{E}[\Delta \frac{\partial L}{\partial w_{mn}^{(l)}}] \leq &\sum_{i,j,m',n'} \mathbb{E}[|\epsilon_{(i+m)(j+n)}^{(l)}|] \\ &\cdot \mathbb{E}[|g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)}|] \\ \approx &\, 0.5354 \cdot (\sum_{i,j,m',n'} \mathbb{E}[|g_{ijm'n'}^{(l)} \cdot w_{m'n'}^{(l+1)} \cdot h_{ij}^{(l+1)}|]) \end{aligned} \tag{21}$$

From the above equations, it is evident that as the size k of the convolutional kernel in the subsequent layer increases, the element-wise gradient error introduced during the binarization process also increases. Consequently, in binarized convolutional units, the smaller the size of the convolutional kernel k, the smaller the binarization error introduced into the binarized model.

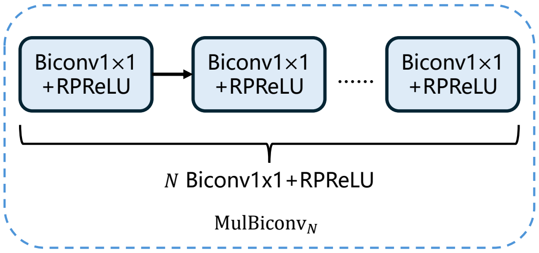


Fig. 4: The struct of MulBiconv$_N$. It consists of N RPReLU activations and 1×1 binarized convolutional layers

Based on Theorem 1, using a 3×3 convolutional kernel for binarized convolution leads to more binarization errors than a 1×1 kernel. Additionally, the model necessitates the presence of the first 3×3 binarized convolutional layer to maintain its capability for extracting local features. Therefore, building upon the binarized convolution unit BDC-V0, we introduce a 1×1 binarized convolutional layer after the 3×3 binarized convolution and before the residual connection, proposing **BDC-V1** unit as shown in Figure 5(b). By deepening the binarized convolution unit, BDC-V1 enhances its feature extraction capability while effectively balancing the trade-off introduced by binarization errors.

We further investigate whether increasing the model's parameter count by introducing additional 1×1 binary convolutional layers can further enhance model performance. To this end, we incorporated multiple 1×1 binary convolutional layers into BDC-V1 unit, resulting in a new unit named **BDC-V2**. The structure of BDC-V2 is illustrated in Figure 5(c). The newly added multi-layer binarized convolution is defined as MulBiconv$_N$, which consists of N RPReLU activations and 1×1 binarized convolutional layers, as illustrated in Figure 4. This can be formulated as:

$$\text{MulBiconv}_N(\cdot) = \text{Repeat}_N(\text{Biconv}1 \times 1(\text{RPReLU}(\cdot))) \tag{22}$$

where Repeat$_N(f)$ denotes repeating N times operation f.

However, BDC-V2 does not lead to performance improvement. On the contrary, in our experiments, we observe a decreasing trend in network performance as the number of 1×1 binarized convolutional layers increases. This occurs because the accumulated binarization errors increase with the addition of more binarized convolutional layers in the unit. The negative impact of these binarization errors on the performance of binary models outweighs the benefits of the increased parameter count, ultimately leading to a decline in overall model performance.

D. Per-Channel Refinement Branch

The straightforward addition of 1×1 binarized convolutions in BDC-V2, compared to BDC-V1, fails to deliver positive improvements for the 3D occupancy prediction task. To further enhance the perceptual capability of binarized models, we analyzed the theoretical limitations of the binarized unit and proposed per-channel refinement branch.

Theorem 2. *Compared to full-precision convolutional layers, binarized convolutional layers exhibit disadvantages in capturing the scale variations across multiple channels of the feature maps. The specific expression is as follows.*

$$\sup_{X,\phi_{c_1},\phi_{c_2}} |S_{\hat{y}^{c_1}} - S_{\hat{y}^{c_2}}| < \sup_{X,\phi_{c_1},\phi_{c_2}} |S_{y^{c_1}} - S_{y^{c_2}}| \tag{23}$$

Let $X \in \mathbb{R}^{C \times H \times W}$ represent the input feature maps, and let ϕ_c denote the full-precision convolution kernel of the c-th channel, which satisfies $avg(|\phi_c|) < max(|\phi_c|)$. The term $S_.$ refers to the scale of the feature map, defined as the normalized ℓ_1-norm. Furthermore, y and $\hat{y}$ represent the output feature map for a specific channel obtained from ϕ_c and its binarized version, respectively.

Proof. *We define the input feature maps as $\mathbf{X} = [x^1, x^2, \ldots, x^C], \mathbf{X} \in \mathcal{R}^{C \times H \times W}$, and the output feature maps of the full-precision convolution as $\mathbf{Y} = [y^1, y^2, \ldots, y^C], \mathbf{Y} \in \mathcal{R}^{C \times H \times W}$, where we assume the number of channels remains unchanged. For the scale S_{y^c} of the c-th channel in the output feature map, we have:*

$$\begin{aligned} y^{c,i,j} &= \sum_{q=1}^{C} \sum_{m',n'} (x^{q,i+m',j+n'} \phi_c^{q,m\prime,n\prime} + b_c^{q,m\prime,n\prime}) \\ S_{y^c} &= avg_{i,j}(|y^{c,i,j}|) = \frac{1}{HW} \sum_i \sum_j |y^{c,i,j}| \end{aligned} \tag{24}$$

where ϕ_c and b_c are the weight and bias of the c-th kernel, respectively. Consider $S_{y^{c_1}}$ and $S_{y^{c_2}}$, and if $S_{y^{c_2}} < S_{y^{c_1}}$, we have:

$$\begin{aligned} S_{y^{c_1}} - S_{y^{c_2}} &= \frac{1}{HW} \sum_i \sum_j |y^{c_1,i,j}| - \frac{1}{HW} \sum_i \sum_j |y^{c_2,i,j}| \\ &\leq \frac{1}{HW} \sum_i \sum_j |y^{c_1,i,j} - y^{c_2,i,j}| \end{aligned} \tag{25}$$

Let the bias b be 0, for full-precision convolution:

$$\begin{aligned} |y^{c_1,i,j} - y^{c_2,i,j}| &= |\sum_{q=1}^{C} \sum_{m',n'} (x^{q,i+m',j+n'} \phi_{c_1}^{q,m\prime,n\prime} \\ &\quad - x^{q,i+m',j+n'} \phi_{c_2}^{q,m\prime,n\prime})| \\ &\leq Ck^2 \cdot max(|x^{q,i+m',j+n'}|) \\ &\quad \cdot max(|\phi_{c_1}^{q,m\prime,n\prime} - \phi_{c_2}^{q,m\prime,n\prime}|) \\ &\leq Ck^2 \cdot max(|x^{q,i+m',j+n'}|) \\ &\quad \cdot (max(|\phi_{c_1}^{q,m\prime,n\prime}|) + max(|\phi_{c_2}^{q,m\prime,n\prime}|)) \end{aligned} \tag{26}$$

For binary convolution, we have:

$$\begin{aligned} |\hat{y}^{c_1,i,j} - \hat{y}^{c,i,j}| &= |\sum_{q=1}^{C} \sum_{m',n'} (x^{q,i+m',j+n'} (avg(\phi_{c_1}) w_{c_1}^{q,m\prime,n\prime} \\ &\quad - avg(\phi_{c_2}) w_{c_2}^{q,m\prime,n\prime})| \\ &\leq Ck^2 \cdot max(|x^{q,i+m',j+n'}|) \\ &\quad \cdot max(|avg(\phi_{c_1}) w_{c_1}^{q,m\prime,n\prime} \\ &\qquad - avg(\phi_{c_2}) w_{c_2}^{q,m\prime,n\prime}|) \\ &\leq Ck^2 \cdot max(|x^{q,i+m',j+n'}|) \\ &\quad \cdot (|avg(\phi_{c_1})| + |avg(\phi_{c_2})|) \end{aligned} \tag{27}$$

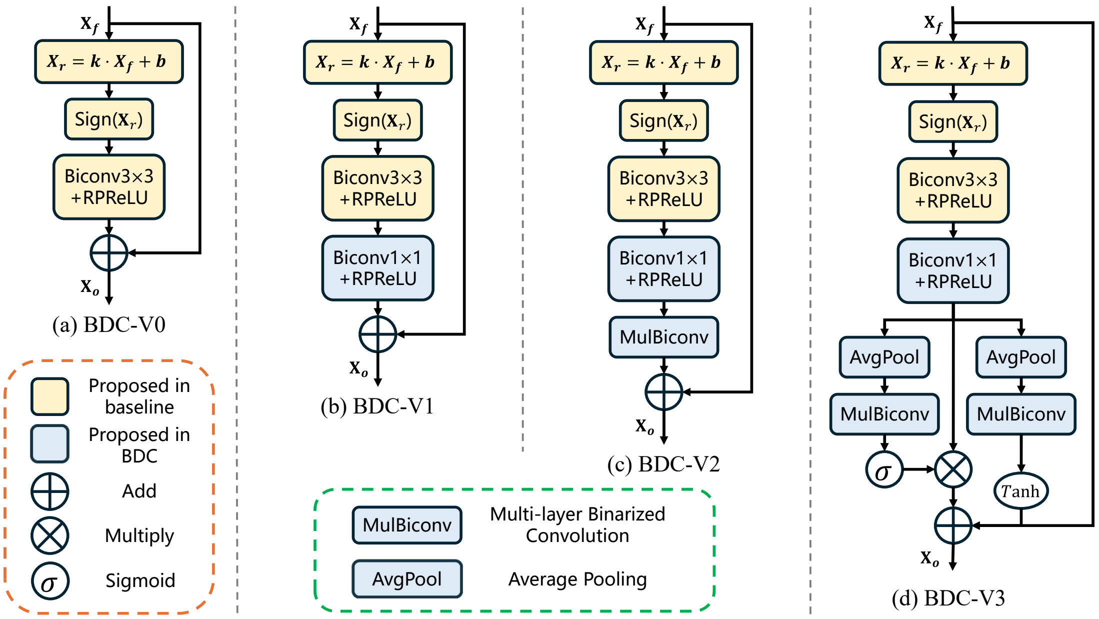


Fig. 5: The illustration of the **design process** of our BDC. We employ the BiSR-Conv [27] unit as the initial version of the binarized convolution unit (**BDC-V0**). Building upon this, we demonstrate and prove that 1×1 convolution introduces minimal binarization error, which forms the basis of **BDC-V1**. **BDC-V2**, an ineffective intermediate product in the design process, is intended to show that simply increasing the number of 1×1 convolution layers in BDC-V1 does not further improve performance. **BDC-V3**, the adopted version of the proposed **BDC** unit, leverages per-channel refinement to emulate the cross-channel feature extraction capability of full-precision convolution at the first-order level. Moreover, it enables the binarized model to focus more on channels less affected by binarization errors, thereby enhancing its perceptual capability.

Here, $w_i^{q,m\prime,n\prime} = sign(\phi_i^{q,m\prime,n\prime})$, thus it can be proven that the supremum of $|y^{c_1,i,j} - y^{c_2,i,j}|$ is greater than $|\hat{y}^{c_1,i,j} - \hat{y}^{c_2,i,j}|$. According to (25)*, the supremum of $S_{Y^{c_1}} - S_{Y^{c_2}}$ is greater than or equal to $S_{\hat{y}^{c_1}} - S_{\hat{y}^{c_2}}$. It indicates that binary convolution reduces the scale differences between different feature channels, which implies a decline in attention across feature channels.*

In BNNs, all weights in each convolutional kernel share a unified scaling factor, with only the polarity varying. The cross-channel amplitude-frequency perception capability of full-precision convolution kernels degrades to a mere frequency response in binarized convolution. Based on Theorem 2, this characteristic of binary convolution hinders its ability to effectively integrate the attention of the input feature map across channels, leading to a suboptimal representation of inter-channel importance in the output feature maps. However, constructing robust inter-channel importance is essential for classification tasks [56] and is equally critical for 3D occupancy prediction tasks, which focus on the classification of 3D samples.

Based on the above considerations, we propose the per-channel refinement branch, which forms the foundation of **BDC-V3** unit. The structure of the per-channel refinement branch is illustrated in Figure 5(d). First, the output of the first 1×1 binarized convolution, $\mathbf{X}_1$, served as the input for the per-channel refinement branch. The first-order and zero-order coefficients, designed to recover channel-wise scaling properties, are obtained through a dual-path structure comprising global average pooling (AvgPool), multi-layer binarized convolution (MulBiconv), and activation functions of Sigmoid and Tanh for each respective path. The branch output $\mathbf{Y}_1$ is formally expressed as

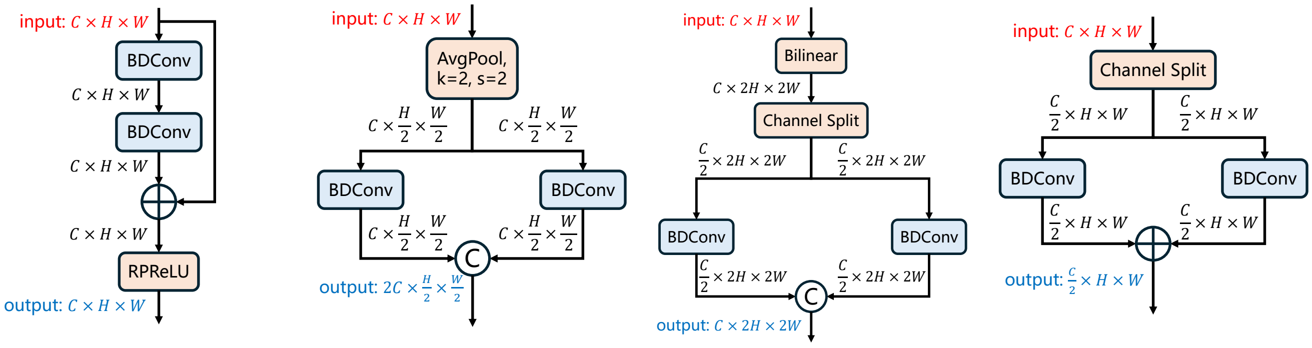
$$\begin{aligned} \mathbf{Y}_1 = \mathrm{Sigmoid}(\mathrm{MulBiconv}_N^A(\mathrm{AvgPool}(\mathbf{X}_1))) \odot \mathbf{X}_1 \\ + \mathrm{Tanh}(\mathrm{MulBiconv}_N^B(\mathrm{AvgPool}(\mathbf{X}_1))) \end{aligned} \tag{28}$$

where $\odot$ denotes element-wise multiplication. Through the proposed per-channel refinement branch, the newly introduced binarized convolutional layers reconstruct and enhance the cross-channel importance of the feature maps, enabling BDC-V3 to emulate the cross-channel feature extraction capability of full-precision convolution at first-order level. Additionally, from the perspective of Theorem 1, modeling the channel importance of feature maps through a first-order approximation enables the binarized model to focus more on channels less affected by binarization errors, thereby enhancing its perceptual capability.

We empirically set $N = 2$, where we observed in experiments that the performance of the binarized network closely approaches the upper bound offered by the full-precision baseline model. We selected the BDC-V3 unit with $N = 2$ as the adopted version of the binarized convolutional unit, referred to as the proposed **BDC** unit.

E. Binarized Convolution Module

Cai et al. [27] demonstrated the necessity of maintaining consistency in input and output dimensions for binarized convolutional layers to ensure the propagation of full-precision


input: $C \times H \times W$
BDConv
$C \times H \times W$
BDConv
$C \times H \times W$
$C \times H \times W$
RPReLU
output: $C \times H \times W$
input: $C \times H \times W$
AvgPool, k=2, s=2
$C \times \frac{H}{2} \times \frac{W}{2}$
$C \times \frac{H}{2} \times \frac{W}{2}$
BDConv
BDConv
$C \times \frac{H}{2} \times \frac{W}{2}$
$C \times \frac{H}{2} \times \frac{W}{2}$
C
output: $2C \times \frac{H}{2} \times \frac{W}{2}$
input: $C \times H \times W$
Bilinear
$C \times 2H \times 2W$
Channel Split
$\frac{C}{2} \times 2H \times 2W$
$\frac{C}{2} \times 2H \times 2W$
BDConv
BDConv
$\frac{C}{2} \times 2H \times 2W$
$\frac{C}{2} \times 2H \times 2W$
C
output: $C \times 2H \times 2W$
input: $C \times H \times W$
Channel Split
$\frac{C}{2} \times H \times W$
$\frac{C}{2} \times H \times W$
BDConv
BDConv
$\frac{C}{2} \times H \times W$
$\frac{C}{2} \times H \times W$
output: $\frac{C}{2} \times H \times W$

(a) Basic (b) DownSample (c) UpSample (d) Channel Reduction

Fig. 6: The illustration of the binarized convolution module based on the BDC unit. A BDConv block represents a BDC unit. In (a), the binarized basic convolutional module preserves both the spatial size and the number of channels in the input feature map; In (b), the binarized downsample convolution module reduces the spatial size of the input feature map by half while doubling the number of channels; In (c), the binarized upsample convolution module doubles the spatial size of the input feature map while preserving the number of channels; In (d), the binarized channel reduction convolution module maintains the spatial size of the input feature map while halving the number of channels.

residual information. Consequently, specialized design considerations are necessary for each binarized convolution module. We decompose the base full-precision CNN-based 3D occupancy network (e.g., FlashOcc) into four types of convolution modules:

(1) Basic convolution module: Input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, output $\mathbf{Y} \in \mathbb{R}^{C\times H\times W}$;

(2) Down-sampling convolution module: Input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, output $\mathbf{Y} \in \mathbb{R}^{2C\times \frac{H}{2}\times \frac{W}{2}}$;

(3) Up-sampling convolution module: Input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, output $\mathbf{Y} \in \mathbb{R}^{C\times 2H\times 2W}$;

(4) Channel reduction convolution module: Input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, output $\mathbf{Y} \in \mathbb{R}^{\frac{C}{2}\times H\times W}$;

For these four convolution modules, we leverage the methodologies from previous works [26], [27], [32] to binarize them, as illustrated in Figure 6. In the constructed binarized convolution modules, the proposed BDC unit serves as the fundamental convolutional unit.

Figure 6(a) illustrates the binarized basic convolutional module, preserving both the size and the number of channels in the input feature map. Figure 6(b) depicts the binarized downsample convolution module, reducing the size of the input feature map by half and doubling the number of channels. Figure 6(c) showcases the binarized upsample convolution module, doubling the size of the input feature map while preserving the number of channels. Finally, Figure 6(d) presents the binarized channel reduction convolution module, maintaining the size of the input feature map while halving the number of channels.

IV. Experiment

A. Experimental Settings

Datasets. We use the Occ3D-nuScenes dataset [60], which comprises 28,130 samples for training and 6,019 samples for validation.

Evaluation Metrics. We evaluate the Occ3D-nuScenes' validation set using the mean Intersection over Union (mIoU) metric. Following [25], [26], [47], [48], [61], [62], we compute the operations per second of BNN (OPs^b) as OPs^b = OPs$^f/64$ to measure the computational complexity, where OPs^f represents FLOPS. To calculate the parameters of BNN, use the formula Params^b = Params$^f/32$, where the superscript b and f refer to the binarized and full-precision models, respectively. To compute the total operations and parameters, we sum OPs as OPs^b + OPs^f and Params as Params^b + Params^f.

Binarization Details. For 3D occupancy prediction tasks, we adopt FlashOcc [36] as the baseline network and binarize it using our proposed BDC unit, assembled in the four different binarized modules described in Section III.E. To ensure performance, we refrain from binarizing the image backbone in the image encoder. This component contains pre-trained weights from image classification tasks, effectively facilitating model convergence and incorporating prior semantic information from images. Fully binarizing it substantially degrades performance, whereas binarizing downstream BEV or occupancy components captures most of the efficiency benefit with far smaller accuracy cost. As illustrated in Figure 3, we binarize the BEV encoder and occupancy head as the **base** version (**-B**) for binarized networks. We further binarize the image neck in the image encoder to obtain the **tiny** version (**-T**) based on the base version.

Implementation Details. We utilized ResNet50 [15] as the image backbone, with an input size of 256×704. Default learning rate 1×10^{-4}, AdamW [63] optimizer, and weight decay of 1×10^{-2} were utilized. The training lasted approximately 29 hours, utilizing 24 epochs on two NVIDIA RTX 3090 GPUs, with a batch size of 2 per GPU. All evaluations are conducted on a single GPU of the same model. Data augmentation strategies for the Occ3D-nuScenes dataset remained consistent with those of FlashOcc [36]. Previous works, such as FlashOcc

TABLE I: **Occupancy Prediction Performance (mIoU↑) on the Occ3D-nuScenes Dataset.** Based on the full-precision baseline model FlashOcc [36], we compare the performance between binarization using our BDC unit and other baseline methods [26]–[29], [32], [33], [57], [58]. The best and second-best performance among BNNs are highlighted in red and blue, respectively. Additionally, we apply the BDC unit to binarize the full-precision model RenderOcc [59], as shown in the last two rows. The BEVDet-Occ [19] is other full-precision model for comparison.

Methods	Params(M)	OPs(G)	others 🟦	barrier 🟧	bicycle 🟪	bus 🟨	car 🟦	const. veh. 🟦	motorcycle 🟨	pedestrian 🟥	traffic cone 🟨	trailer 🟫	truck 🟪	drive. suf. 🟪	other flat ⬜	sidewalk 🟪	terrain 🟩	manmade ⬜	vegetation 🟩	mIoU
CNN-based (32 bit)																				
BEVDet-Occ	29.02	241.76	8.22	44.21	10.34	42.08	49.63	23.37	17.41	21.49	19.70	31.33	37.09	80.13	37.37	50.41	54.29	45.56	39.59	36.01
FlashOcc	44.74	248.57	9.08	46.32	17.71	42.70	50.64	23.72	20.13	22.34	24.09	30.26	37.39	81.68	40.13	52.34	56.46	47.69	40.60	37.84
BNN-based (1 bit)																				
ReActNet-T	26.80	129.74	7.55	38.87	16.64	35.78	44.27	20.34	15.53	16.16	18.70	24.42	33.59	73.64	29.05	39.80	41.27	39.31	34.00	31.29
ReActNet-B	28.17	133.89	8.62	40.92	15.94	37.45	47.23	18.57	17.47	18.91	21.52	23.14	33.13	77.20	34.58	45.48	48.31	42.95	35.06	33.32
PokeBNN-T	26.81	129.84	6.64	42.26	**21.80**	36.29	47.78	22.08	21.33	20.90	21.69	26.09	34.92	78.82	37.75	46.79	49.50	44.40	38.64	35.16
PokeBNN-B	28.18	133.97	7.50	32.19	16.41	36.47	46.25	21.06	16.95	21.43	21.23	26.97	32.97	72.72	27.45	44.51	48.16	41.45	37.28	32.41
AdaBin-T	26.78	129.78	8.21	40.59	17.12	37.02	46.92	21.18	18.67	19.40	19.79	24.56	34.47	76.62	19.77	44.75	48.22	43.87	37.57	32.87
AdaBin-B	28.15	133.79	6.00	38.55	7.89	36.65	45.21	18.10	13.92	17.86	13.77	24.21	32.77	77.75	35.58	45.52	50.54	42.49	36.43	31.95
BBCU-T	26.79	129.69	6.24	38.16	14.33	31.95	43.18	20.57	16.50	17.39	13.45	22.26	32.51	75.69	32.97	42.46	48.50	41.68	35.75	31.39
BBCU-B	28.16	133.84	7.61	41.14	13.64	35.54	46.55	20.86	17.44	19.87	17.58	24.24	33.94	76.19	34.05	44.61	48.08	42.67	35.28	32.27
BiMatting-T	26.82	129.95	5.96	38.17	15.27	35.85	44.11	19.35	14.38	18.98	15.84	23.22	31.16	73.97	30.51	35.42	40.90	41.65	35.05	30.58
BiMatting-B	28.17	134.05	6.80	38.65	17.99	33.02	43.80	19.91	18.29	18.67	19.82	21.83	32.09	72.99	32.44	41.23	43.64	36.24	35.07	31.32
BiSRNet-T	26.79	129.70	8.38	41.06	16.76	33.94	46.11	18.96	19.10	17.90	16.94	23.70	35.14	76.86	35.68	46.77	50.39	41.41	34.78	33.17
BiSRNet-B	28.16	133.85	9.27	41.94	19.53	37.33	47.48	20.83	19.17	20.08	20.21	25.36	33.99	77.42	35.78	47.35	50.58	43.24	37.20	34.51
JDB-T	26.79	129.71	8.64	42.28	17.73	39.14	47.60	21.16	17.66	20.96	20.19	28.61	34.48	79.58	37.35	49.68	53.58	45.22	38.62	35.44
JDB-B	28.20	134.34	8.65	41.56	17.90	38.31	48.05	21.56	20.44	20.92	20.96	27.11	35.17	78.05	36.03	45.81	49.47	42.22	38.04	34.72
BRVE-T	32.56	129.83	9.01	42.04	16.66	37.27	47.34	20.63	18.57	20.32	19.69	**28.66**	35.14	79.42	37.07	49.48	53.65	45.23	38.94	35.24
BRVE-B	30.46	133.94	8.88	43.58	19.07	39.63	48.34	21.74	19.21	21.36	21.03	27.52	35.13	79.62	37.52	49.44	53.64	45.26	38.53	35.85
BDC-T (Ours)	26.83	129.90	**10.16**	**44.38**	18.53	**41.40**	**49.87**	**23.12**	**20.94**	**22.33**	**23.29**	**29.93**	**36.19**	**81.14**	**39.37**	**51.43**	**55.25**	**47.37**	**40.87**	**37.39**
BDC-B (Ours)	28.22	134.50	**9.57**	**44.80**	**20.45**	**40.21**	**49.96**	**23.72**	**21.48**	**22.58**	**24.47**	27.40	**36.48**	**80.22**	**38.34**	**50.12**	**54.74**	**47.19**	**40.04**	**37.16**
CNN-based (32 bit)																				
RenderOcc	-	-	11.23	46.09	23.56	41.36	49.75	25.75	21.93	23.23	25.25	32.51	37.06	81.35	40.83	52.19	55.81	45.66	40.19	38.46
BNN-based (1 bit)																				
BDC-RenderOcc	-	-	11.02	44.25	22.98	40.58	49.92	22.86	22.46	23.71	24.62	31.40	36.63	81.63	40.59	52.58	56.12	46.04	40.09	38.09

and BEVDet-Occ [19], have demonstrated the effectiveness of camera visibility masks during training. Therefore, we also employ camera visibility masks to enhance performance. Following the settings of FlashOcc, we employ the pre-trained model from BEVDet [19] for 3D object detection task.

B. Main Results

Table I presents the evaluation results on the validation set of Occ3D-nuScenes. To validate the effectiveness of our proposed BDC unit, we replace it with other state-of-the-art binarization methods, including ReActNet [32], PokeBNN [57], AdaBin [58], BBCU [26], BiMatting [46], BiSRNet [27], BRVE [29], and JDB [28]. We also compare the binarized 3D occupancy networks with the full-precision counterparts, including BEVDet-Occ [19], RenderOcc [59] and FlashOcc [36], where FlashOcc serves as the baseline network for all binarized methods and represents the theoretical upper limit.

Table I presents performance metrics (mIoU), parameter counts, and the number of operations for different methods. Compared to other binarized methods, our BDC-T and BDC-B achieve the best or second-best results across almost all binarized models. Specifically, BDC significantly improves performance without increasing parameter count or computational complexity. Compared to the previous SOTA method, BRVE-B, our BDC-T demonstrates superior performance in mIoU, exceeding it by 1.54 mIoU (**+4.3%**) while saving **3.02%** of operations and **11.92%** of parameters. Moreover, BDC-T achieves competitive results compared to the full-precision model FlashOcc, using only **52.26%** of operations and **59.97%** of parameters, with a minimal performance loss of -0.45 mIoU (-1.19%) due to binarization errors. Both BBCU and BiSRNet exhibit performance degradation issues when binarizing additional modules. Compared to BDC-B, BDC-T performs slightly better when binarizing image neck modules. It demonstrates the robustness of BDC to the binarized modules. In Table III, we compare the wall-clock time computational efficiency, showing that our model achieves a **21.06%** improvement in FPS.

To validate the generalizability of the proposed BDC, we

TABLE II: **3D Object Detection performance (mAP↑, NDS↑) on the nuScenes `val` set.** Best performance among BNNs are in **bold**. We employ BEVDet [19] as the full-precision baseline network and binarize the image neck, BEV encoder, and detection head using binarization methods, including BiSRNet [27], ReActNet [32], BBCU [26], BiMatting [46], JDB [28], BRVE [29], and our proposed BDC unit.

Methods	Params(M)	OPs(G)	**mAP↑**	**NDS↑**	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CNN-based (32 bit)									
BEVDet	44.25	148.77	0.3836	0.4995	0.5815	0.2790	0.4750	0.3807	0.2067
BNN-based (1 bit)									
ReActNet-T	26.53	101.30	0.3222	0.4358	0.6609	0.3057	0.6298	0.4468	0.2100
BBCU-T	26.51	101.24	0.3166	0.4046	0.6697	0.3137	0.7822	0.5461	0.2255
BiMatting-T	26.55	101.41	0.3356	0.4428	0.6358	0.2968	0.6527	0.4485	0.2159
BiSRNet-T	26.52	101.25	0.3431	0.4519	0.6633	0.2940	0.5777	0.4550	0.2061
JDB-T	26.52	101.25	0.2606	0.3594	0.7220	0.3430	0.6634	0.7412	0.2392
BRVE-T	32.28	101.33	0.3485	0.4585	0.6573	0.2895	0.5621	**0.4434**	0.2054
BDC-T	26.56	101.36	**0.3648**	**0.4742**	**0.6291**	**0.2822**	**0.5250**	0.4460	**0.1994**

TABLE III: **Computational efficiency.** The FPS and runtime (ms) for 32-bit and 1-bit versions of FlashOcc [36] and BDC-T. The comparison of operations and parameters is shown in Table I.

Methods	32 bit	1 bit	total time	FPS
FlashOcc	160.77	0	160.77	6.22
BDC-T	130.93	1.88	132.81	7.53

also conduct experiments on 3D object detection tasks using the nuScenes [64] dataset. Table II presents performance metrics for the 3D object detection task in nuScenes, where our approach, BDC, consistently demonstrates superior performance across all metrics.

For a comparison of direct quantization on full-precision model, as illustrated in Figure 7, we use FlashOcc without temporal information as the baseline and employ BDC-T for binarization on this baseline. We plot the performance of FlashOcc at different bit levels and compare it with the performance of BDC-T. We can observe that the performance of our BDC-T is comparable to that of the full-precision model and is superior to the performance of FlashOcc at both 16-bit and 8-bit levels.

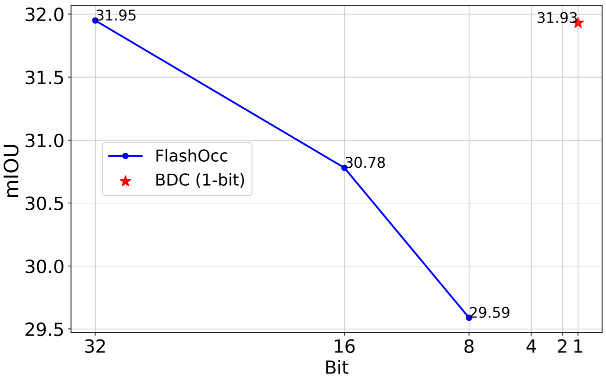


Fig. 7: FlashOcc mIoU-bit curve and our BDC performance. The blue curve represents the direct quantization of FlashOcc. Compared to direct quantization, BDC achieves optimal compression rates and superior 3D occupancy prediction performance.

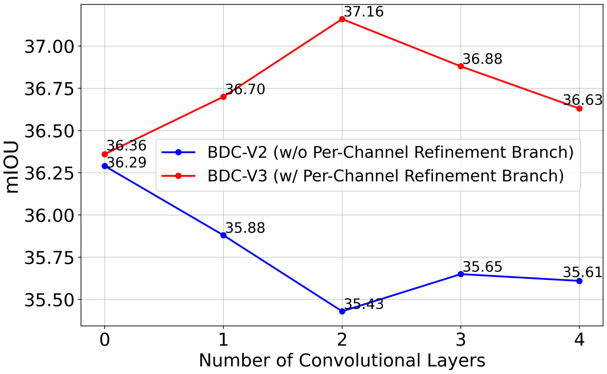


Fig. 8: Ablation study of multi-layer binarized convolution (MulBiconv). Increasing the number of 1×1 binarized convolution layers in BDC-V2 degrades performance, as indicated by the blue curve. In BDC-V3, the optimal number of 1×1 binarized convolution layers is two, as demonstrated by the red curve.

C. Ablation Study

In all ablation studies, the binarization settings are configured as the **base** version (**-B**) for all methods, as described in Table I. Furthermore, we emphasize that BDC-V3 corresponds to our final design (i.e., the proposed BDC unit), while the other versions should be regarded as intermediate variants explored during the design process.

Multi-layer Binarized Convolution (MulBiconv) Ablation. To explore the impact of the number of binarized convolutional layers in MulBiconv on the model's performance, we binarize FlashOcc using both BDC-V2 and BDC-V3 (i.e., BDC) unit while varying the number of binarized convolutional layers in MulBiconv ($N = 0, 1, 2, 3, 4$).

The results are illustrated in Figure 8. When $N = 0$, the structure of BDC-V2 is identical to that of BDC-V1. BDC-V3 contains no learnable parameters with the per-channel refinement branch. As N increases, we observe a gradual decline followed by fluctuations in the performance of BDC-V2. This also confirms our statement in Section III.C. that the negative impact of straightforwardly adding 1×1 binarization layers outweighs the benefits of the increased parameter count. In contrast, BDC-V3 initially shows performance improvement, followed by decreases as N increases. When MulBiconv selects $N = 2$, BDC-V3 achieves the best performance,

TABLE IV: **Break-down ablation.** Figure 5 illustrates the structure of various versions of the BDC.

Methods	mIoU	OPs (G)	Params (M)
BDC-V0	34.51	133.85	28.16
BDC-V1	36.29	133.93	28.17
BDC-V2	35.43	134.10	28.19
BDC-V3	**37.16**	134.50	28.22

TABLE V: **Kernel size ablation.** $A \to B$ represents the concatenation structure of $A \times A$ binarized convolution followed by $B \times B$ binarized convolution.

Kernel	mIoU	OPs (G)	Params (M)
$3 \to 1$	**36.29**	133.93	28.17
$3 \to 3$	33.01	133.93	28.17
$1 \to 1$	35.32	133.93	28.17
$3 \to 3 \to 1$	33.37	134.02	28.18

TABLE VI: **The proportion of 32-bit OPs and Params versus 1-bit OPs and Params in each module of BDC-T.** (%) denotes the proportion of 32-bit and 1-bit operations within each module.

	Model	Bit	Image Backbone	Image Neck	View Transformer	BEV Backbone	BEV Neck	Occupancy Head	Total
OPs(G)	FlashOcc	32-bit	88.785	1.377	0.165	17.724	102.989	34.755	248.572
	BDC-T	32-bit	88.785	0	0.165	0	29.491 (28.64%)	11.141 (32.06%)	129.582 (52.13%)
		1-bit	0	0.034	0	0.046	0.474 (71.36%)	0.031 (67.94%)	0.585 (47.87%)
Params(M)	FlashOcc	32-bit	23.508	4.155	0.039	12.394	6.556	0.869	44.744
	BDC-T	32-bit	23.508	0	0.039	0	2.949 (44.98%)	0.279 (32.11%)	26.775 (59.84%)
		1-bit	0	0.022	0	0.020	0.012 (55.02%)	0.001 (67.89%)	0.055 (40.16%)

reaching 37.16 mIoU. The optimal trade-off occurs when the performance gain from increasing model parameters outweighs the performance degradation caused by binarization errors.

Break-down Ablation. We binarize FlashOcc using four variants of BDC, where BDC-v0 is equivalent to the binarized method BiSRNet. Additionally, BDC-V2 and BDC-V3 utilize the multi-layer binarized convolution (MulBiconv), and we set $N = 2$.

The results are presented in Table IV, from which we can draw the following conclusions: (1) Compared to BDC-V0, BDC-V1 achieves a significant gain of **1.78 mIoU (+5.16%)** by adding only one 1×1 binarized convolution layer. Extra binarized convolution layers result in negligible changes to full model parameters and computational complexity. (2) By adding MulBiconv to each binarized convolution unit in BDC-V1 (i.e., BDC-V2), we observe a substantial decrease in performance, along with slight increases in parameters and computational complexity. (3) Compared to BDC-V2, BDC-V3 (i.e., BDC) exhibits a significant performance improvement of **1.73 mIoU**. Additionally, BDC-V3 gains an extra **0.87 mIoU** over BDC-V1. Placing additional binarized convolutional layers within the per-channel refinement branch effectively enhances model performance.

Kernel Size Ablation. To validate whether 3×3 binarized convolutions incur more binarization errors than 1×1 ones, potentially leading to performance degradation, we apply BDC-V1 and BDC-V2 ($N = 1$) to FlashOcc. We present the results in Table V. For BDC-V1, replacing the 1×1 binarized convolution with consecutive 3×3 binarized convolutions led to a decrease in performance from 36.29 mIoU to 33.01 mIoU.

TABLE VII: Performance metrics for binarizing different modules

Module	**mIoU**
Image Neck	37.91 (+0.07 ↑)
BEV Backbone	31.62 (-6.22 ↓)
BEV Neck	37.59 (-0.25 ↓)
Occupancy Head	31.46 (-6.38 ↓)
BDC-T	37.39 (-0.45 ↓)
FlashOcc	37.84

Additionally, we validate the necessity of using a 3×3 binarized convolution as the first convolution layer. If replaced with a 1×1 binarized convolution, the receptive field of the binarized convolution unit becomes limited, preventing the establishment of connections with neighboring pixel features, resulting in a decrease in performance from 36.29 mIoU to 35.32 mIoU. Experiments conducted on BDC-V2 ($N = 1$) also support the conclusion that consecutive 3×3 binarized convolutions lead to binarization errors and affect binarized model performance.

Module Ablation. We binarized different modules in the 3D occupancy network. According to Table VII, we can find: (1) The binarization of the BEV backbone and the occupancy head significantly impacts performance. (2) During joint training, the binarization errors of the entire network can be considered and optimized as a whole. In Table VI, we investigate the changes in computation (OPs) and parameters (Params) across different modules of the 3D occupancy network before and

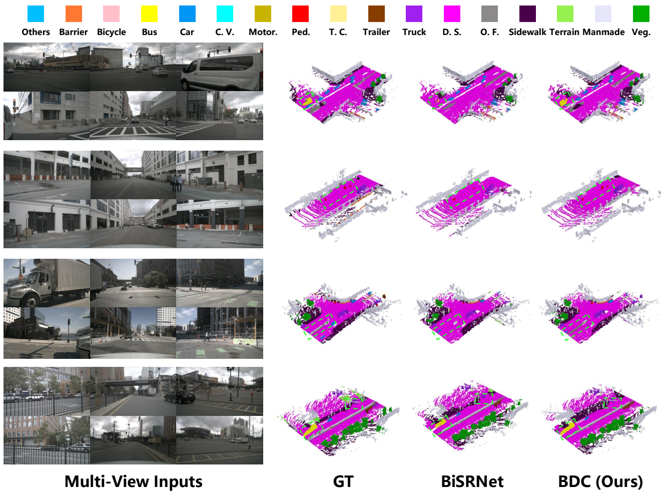


Fig. 9: Visualization rensults on Occ3D-nuScenes validation set

after binarization. The image encoder consists of the image backbone and image neck, while the BEV encoder includes the BEV backbone and BEV neck. (x%) indicates that x% of the full-precision operations/parameters have been binarized.

We do not binarize the view transformer because its 32-bit full-precision parameters and computation are already sufficient. Additionally, the view transformer relies on full-precision computation to precisely map 2D image features to 3D BEV features.

D. Visualization

We also present some qualitative results on the Occ3D-nuScenes' validation set. As illustrated in Figure 9, BDC exhibits comprehensive predictions about the bus in the first and last rows. In the second row, BDC successfully identifies all pedestrians, whereas BiSRNet overlooks some pedestrians in the scene. Moreover, in the third row, BDC provides accurate predictions about curbs, whereas BiSRNet misclassifies them as drivable surfaces, potentially posing safety concerns. Additionally, in the fourth row, BDC accurately reconstructs traffic lights in the scene, showcasing its robust capability in scene perception.

V. Conclusion

This paper introduces a binarized deep convolution (BDC) unit for 3D occupancy networks, addressing the performance degradation caused by increasing the number of binarized convolutional layers. Our original theoretical analysis shows that 1×1 binarized convolution introduces minimal binarization errors, and binarized convolution is less effective than full-precision convolution in capturing cross-channel feature importance. Consequently, we restrict additional binarized convolution kernels to 1×1 in the BDC unit. Furthermore, we propose a per-channel refinement branch to overcome the limitations of binarized convolutional layers in assigning feature importance across channels. Extensive experiments validate that our method surpasses existing SOTA binarized convolution networks and closely approaches the performance of full-precision models while using only **52.26%** of the operations and **59.97%** of the parameters and achieving a **21.06%** improvement in FPS.

Limitations. Our study has several limitations. In our main setting, the image backbone remains full-precision to ensure stability and semantic priors, which constrains the overall compression upper bound. Moreover, the present work primarily focuses on convolutional pipelines tailored to 3D occupancy, leaving Transformer-based architectures for future exploration. In addition, the theoretical analysis relies on simplifying assumptions to approximate practical training dynamics, which may not fully capture all interactions. Given the absence of native 1-bit tensor primitives on commodity GPUs, the reported OPs and Params improvements and wall-clock speedups are best viewed as a first estimate of potential gains, not definitive measurements attainable with hardware-level 1-bit arithmetic. Device-level kernel optimization and hardware co-design remain promising directions for future work.

References

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[2] J. Ye, Z. Zhang, Y. Jiang, Q. Liao, W. Yang, and Z. Lu, "Occgaussian: 3d gaussian splatting for occluded human rendering," *arXiv preprint arXiv:2404.08449*, 2024.

[3] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan *et al.*, "Vastgaussian: Vast 3d gaussians for large scene reconstruction," *arXiv preprint arXiv:2402.17427*, 2024.

[4] Y. Shi, K. Jiang, J. Li, J. Wen, Z. Qian, M. Yang, K. Wang, and D. Yang, "Grid-centric traffic scenario perception for autonomous driving: A comprehensive review," *arXiv preprint arXiv:2303.01212*, 2023.

[5] X. Yan, H. Zhang, Y. Cai, J. Guo, W. Qiu, B. Gao, K. Zhou, Y. Zhao, H. Jin, J. Gao *et al.*, "Forging vision foundation models for autonomous driving: Challenges, methodologies, and opportunities," *arXiv preprint arXiv:2401.08045*, 2024.

[6] H. Zhang, X. Yan, D. Bai, J. Gao, P. Wang, B. Liu, S. Cui, and Z. Li, "Radocc: Learning cross-modality occupancy knowledge through rendering assisted distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7060–7068.

[7] Y. Wang, R. Gao, K. Chen, K. Zhou, Y. Cai, L. Hong, Z. Li, L. Jiang, D.-Y. Yeung, Q. Xu *et al.*, "Detdiffusion: Synergizing generative and perceptive models for enhanced data generation and perception," *arXiv preprint arXiv:2403.13304*, 2024.

[8] Z. Lu, B. Cao, and Q. Hu, "Lidar-camera continuous fusion in voxelized grid for semantic scene completion," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[9] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3991–4001.

[10] Y. Wang, Y. Chen, X. Liao, L. Fan, and Z. Zhang, "Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation," *arXiv preprint arXiv:2306.10013*, 2023.

[11] J. Liu, S. Zhang, C. Kong, W. Zhang, Y. Wu, Y. Ding, B. Xu, R. Ming, D. Wei, and X. Liu, "Occtransformer: Improving bevformer for 3d camera-only occupancy prediction," *arXiv preprint arXiv:2402.18140*, 2024.

[12] Z. Hong and C. P. Yue, "Real-time 3d visual perception by cross-dimensional refined learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[13] W. Ouyang, Z. Xu, B. Shen, J. Wang, and Y. Xu, "Linkocc: 3d semantic occupancy prediction with temporal association," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[14] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "Fb-occ: 3d occupancy prediction based on forward-backward view transformation," *arXiv preprint arXiv:2307.01492*, 2023.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[19] J. Huang and G. Huang, "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection," *arXiv preprint arXiv:2203.17054*, 2022.

[20] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, 2020.

[21] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*. Chapman and Hall/CRC, 2022, pp. 291–326.

[22] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[23] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[24] Y. Zhou, S. Chen, Y. Wang, and W. Huan, "Review of research on lightweight convolutional neural networks," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2020, pp. 1713–1720.

[25] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[26] B. Xia, Y. Zhang, Y. Wang, Y. Tian, W. Yang, R. Timofte, and L. Van Gool, "Basic binary convolution unit for binarized image restoration network," in *The Eleventh International Conference on Learning Representations*, 2023.

[27] Y. Cai, Y. Zheng, J. Lin, X. Yuan, Y. Zhang, and H. Wang, "Binarized spectral compressive imaging," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[28] Y. Xie, X. Hou, Y. Guo, X. Wang, and J. Zheng, "Joint-guided distillation binary neural network via dynamic channel-wise diversity enhancement for object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 1, pp. 448–460, 2024.

[29] G. Zhang, Y. Zhang, X. Yuan, and Y. Fu, "Binarized low-light raw video enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 753–25 762.

[30] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106–3121, 2019.

[31] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, pp. 251–275, 2020.

[32] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng, "Reactnet: Towards precise binary neural network with generalized activation functions," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 143–159.

[33] H. Qin, L. Ke, X. Ma, M. Danelljan, Y.-W. Tai, C.-K. Tang, X. Liu, and F. Yu, "Bimatting: Efficient video matting via binarization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[34] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[35] W. Gan, N. Mo, H. Xu, and N. Yokoya, "A simple attempt for 3d occupancy estimation in autonomous driving," *arXiv preprint arXiv:2303.10076*, 2023.

[36] Z. Yu, C. Shu, J. Deng, K. Lu, Z. Liu, J. Yu, D. Yang, H. Li, and Y. Chen, "Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin," *arXiv preprint arXiv:2311.12058*, 2023.

[37] J. Mei, Y. Yang, M. Wang, J. Zhu, X. Zhao, J. Ra, L. Li, and Y. Liu, "Camera-based 3d semantic scene completion with sparse guidance network," *arXiv preprint arXiv:2312.05752*, 2023.

[38] Z. Ming, J. S. Berrio, M. Shan, and S. Worrall, "Inversematrixvt3d: An efficient projection matrix-based approach for 3d occupancy prediction," *arXiv preprint arXiv:2401.12422*, 2024.

[39] J. Hou, X. Li, W. Guan, G. Zhang, D. Feng, Y. Du, X. Xue, and J. Pu, "Fastocc: Accelerating 3d occupancy prediction by fusing the 2d bird's-eye view and perspective view," *arXiv preprint arXiv:2403.02710*, 2024.

[40] 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 *European conference on computer vision*. Springer, 2022, pp. 1–18.

[41] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar, "Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9087–9098.

[42] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9223–9232.

[43] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 729–21 740.

[44] H. Jiang, T. Cheng, N. Gao, H. Zhang, W. Liu, and X. Wang, "Symphonize 3d semantic scene completion with contextual instance queries," *arXiv preprint arXiv:2306.15670*, 2023.

[45] H. Liu, H. Wang, Y. Chen, Z. Yang, J. Zeng, L. Chen, and L. Wang, "Fully sparse 3d panoptic occupancy prediction," *arXiv preprint arXiv:2312.17118*, 2023.

[46] Z. Li, Y. Zhang, J. Lin, H. Qin, J. Gu, X. Yuan, L. Kong, and X. Yang, "Binarized 3d whole-body human mesh recovery," *arXiv preprint arXiv:2311.14323*, 2023.

[47] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 722–737.

[48] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[49] T. Chen, Z. Zhang, X. Ouyang, Z. Liu, Z. Shen, and Z. Wang, "" bnn-bn=?": Training binary neural networks without batch normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4619–4629.

[50] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, and J. Song, "Forward and backward information retention for accurate binary neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2250–2259.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[52] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[53] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2384, 1998.

[54] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University Press, 2008.

[55] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted gaussian signals," Research Laboratory of Electronics, Massachusetts Institute of Technology, Tech. Rep. Technical Report No. 216, 1952.

[56] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[57] Y. Zhang, Z. Zhang, and L. Lew, "Pokebnn: A binary pursuit of lightweight accuracy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 475–12 485.

[58] Z. Tu, X. Chen, P. Ren, and Y. Wang, "Adabin: Improving binary neural networks with adaptive binary sets," in *European conference on computer vision*. Springer, 2022, pp. 379–395.

[59] M. Pan, J. Liu, R. Zhang, P. Huang, X. Li, H. Xie, B. Wang, L. Liu, and S. Zhang, "Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 404–12 411.

[60] X. Tian, T. Jiang, L. Yun, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *arXiv preprint arXiv:2304.14365*, 2023.

[61] Y. Cai, Y. Zheng, J. Lin, X. Yuan, Y. Zhang, and H. Wang, "Binarized spectral compressive imaging," *Advances in Neural Information Processing Systems*, vol. 36, pp. 38 335–38 346, 2023.

[62] G. Zhang, Y. Zhang, X. Yuan, and Y. Fu, "Binarized low-light raw video enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 753–25 762.

[63] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *Learning,Learning*, Nov 2017.

[64] 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.

Biography Section

Zongkai Zhang received the B.S. degree in Informatics Engineering from Southeast University, Nanjing, China, in 2021. He is currently working toward the M.S. degree in Electronics and Communication Engineering with the Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. He is now focusing on 3D visual perception in autonomous driving.

Peng Ling received the B.S. degree in Intelligent Science and Technology from South China University of Technology, Guangzhou, China, in 2023. He is currently pursuing the Ph.D. degree in Information and Communication Engineering at the Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. His research interests include autonomous driving, 3D vision, and embodied intelligence.

Zidong Xu received the B.S. degree in Electronic and Information Engineering from Sun Yat-sen University, Guangzhou, China, in 2021. He is currently working toward the M.S. degree in Electronics and Communication Engineering with the Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. He is now focusing on 3D visual perception in autonomous driving.

Wenming Yang (IEEE Senior Member) received his Ph.D. degree in information and communication engineering from Zhejiang University in 2006. He is an associate professor in Shenzhen International Graduate School / Department of Electronic Engineering, Tsinghua University. His research interests include image processing, computer vision, pattern recognition, deep learning and their applications.

Qingmin Liao (IEEE Senior Member) received the B.S. degree in radio technology from the University of Electronic Science and Technology of China, China, in 1984, and the M.S. and Ph.D. degrees in signal processing and telecommunications from the University of Rennes 1, France, in 1990 and 1994, respectively. He is currently a Professor with the Shenzhen International Graduate School / Department of Electronic Engineering, Tsinghua University. His research interests include image / video processing, analysis, biometrics, and their applications.

Jing-Hao Xue (IEEE Senior Member) received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998, and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor of Statistical Pattern Recognition in the Department of Statistical Science, University College London. His research interests include statistical pattern recognition, machine learning, and computer vision. He received the Best Associate Editor Award of 2021 from the IEEE Transactions on Circuits and Systems for Video Technology, and the Outstanding Associate Editor Award of 2022 from the IEEE Transactions on Neural Networks and Learning Systems.