QD-BEV : Quantization-aware View-guided Distillation for Multi-view 3D Object Detection

Anonymous submission

Abstract

Camera-only 3D detection based on BEV (bird-eye-view) has achieved significant improvements recently. However, due to its tremendous resource consumption, achieving large-scale deployment of state-of-the-art models on autonomous driving vehicles is difficult. In this work, we pioneer an efficient BEV model, **QD-BEV**, via a systematic progressive quantization-aware training (QAT) pipeline and a novel view-guided BEV distillation algorithm. Despite the wide application of quantization and distillation to lightening models in other tasks, as pointed out in our paper, directly applying these methods leads to intolerable performance degradation in BEV tasks. To solve this issue, QD-BEV enables an stable and effective QAT pipeline with novel distillation objective, converting the cumbersome BEVFormer model into efficient QD-BEV model. Systematic experiments demonstrate QD-BEV achieves comparable or even higher accuracy than prior art with significant efficiency improvements. On the nuScenes datasets, the 4-bit weight and 6-bit activation quantized QD-BEV-Tiny model achieves 37.2% NDS with only 15.8 MB model size, outperforming BevFormer-Tiny by 1.8% with an $8\times$ model compression. On the Small and Base variants, QD-BEV models also perform superbly and achieve 47.9% NDS (28.2 MB) and 49.2% NDS (32.9 MB), respectively.

Introduction and related work

Given its potential in enabling autopilot, camera-only 3D detection based on BEV (bird-eye-view) has became an important research direction for autonomous driving. Based on input sensors, previous work can be divided into LiDAR-based methods (Lang et al. 2019; Zhou and Tuzel 2018) and camera-only methods (Li et al. 2022; Wang et al. 2022; Huang et al. 2021; Huang and Huang 2022; Liu et al. 2022a,b). Compared to the LiDAR-based methods, camera-only methods have the merits of lower deployment cost, closer to human eyes, and easier access to visual information in the driving environment. However, even if using the camera-only methods, the computational costs of running state-of-the-art BEV models are still formidable, making it difficult to deploy the model onto vehicles. For example, BEVFormer-Base has a 540 ms inference latency (corresponds to 1.85 fps) on one NVIDIA V100 GPU, which is infeasible for real-time applications that generally require 30 fps. Therefore, it is particularly crucial to explore and devise lightweight models for camera-only 3D object detection based on BEV, such as quantization (Jacob et al. 2018; Gholami et al. 2021) and knowledge distillation (Hinton, Vinyals, and Dean 2015; Yin et al. 2020). Quantization can greatly save the model size and computational costs while improving the speed of model reasoning. However, directly applying quantization would lead to significant performance degradation. Compared to image classification and 2D object detection tasks,multi-camera 3D detection tasks are much more complicated and difficult due to the existence of multiple views and information from multiple dimensions(for example, the temporal information and spatial information used in BEVFormer (Li et al. 2022)).

In this work, we first conduct systematic experiments and analyses on quantizing BEV networks. Then we devise a quantization-aware view-guided distillation method (referred to as QD-BEV) that can decently solve the stability issue of standard QAT while improving the final performance of compact BEV models. Our proposed view-guided distillation (VGD) can better leverage information from both the image and the BEV domains, which can significantly outperform previous distillation methods which cannot jointly handle the different types of losses in BEV networks. As shown in Figure 1, we construct our QD-BEV pipeline leveraging the mapping relationship between the image feature and the BEV feature. Specifically, we first take the FP (floating-point) model as the teacher model and the low-bit quantized model as the student model, then we calculate the KL divergence on the image feature and the BEV feature, respectively.

Finally, we realize our unique View-Guided Distillation (VGD) by organically combining the image feature and the BEV feature through the camera's external parameters. Note that in our QD-BEV pipeline, neither additional training data nor larger powerful teacher networks are used to tune the accuracy, but QD-BEV models are still able to outperform previous baselines while having a significantly smaller model size and computational requirements. Our contributions are as follows:

- We pioneer the use of quantization to obtain QD-BEV, a family of efficient models for camera-only 3D object detection based on BEV.
- We conduct systematic experiments on quantizing BEV models, unveiling major issues hampering standard quantization-aware training methods on BEV.

- We propose view-guided distillation (VGD), which jointly leverages both image domain and BEV domain information. VGD boosts QAT performance by solving the stability issue of standard QAT.
- The resulted QD-BEV outperforms previous baselines while being significantly smaller. The W4A6 quantized QD-BEV-Tiny has 37.2% NDS with an only 15.8 MB model size, which outperforms the 8× larger BevFormer-Tiny model by 1.8%.

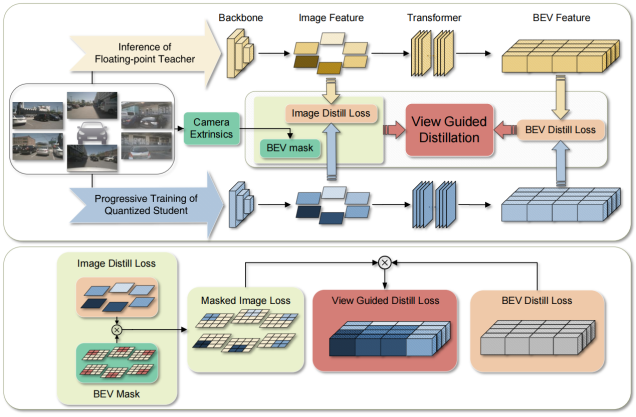


Figure 1: **Illustration of QD-BEV.** In our pipeline, multi-camera images are input into the floating-point teacher network and the quantized student network in order to compute the KL divergence in an element-wise manner. The KL divergence is used as distillation loss in the image feature and the BEV feature, respectively. Then we conduct view-guided distillation using the BEV mask obtained from the external parameters of the camera.

Method

In this section, we introduce our view-guided distillation and progressive quantization-aware training methods in detail.

Progressive quantization-aware training

In symmetric linear quantization, the quantizer maps weights and activations into integers with a scale factor S. Uniformly quantizing to k bit can be expressed as:

$$S = \frac{2|r_{\max}|}{2^k - 1},\ q = round\left(\frac{r}{S}\right), \tag{1}$$

where r is the floating-point number being quantized, $|r_{\max}|$ is the largest absolute value in r, and q is the quantized integer. In this work, we conduct systematic experiments to analyze the performance of quantization on BEV networks. For PTQ, we apply the above quantization directly to the pre-trained models during the inference stage. For QAT, we utilize the straight-through estimator (STE) (Bengio, Léonard, and Courville 2013) to define the forward and backward pass for the above quantization operations and train the model to better adapt to quantization. Both methods are useful and can be applied to different circumstances for deployment.

As discussed in , due to the complexity of multi-camera 3D object detection tasks and of the BEV networks, directly quantizing the model to ultra-low bit-width may incur sub-optimal and unstable training, even gradient explosion. To solve this problem, we introduce progressive QAT which reduces the quantization influence with a two-fold process. First, we propose to reach the target bit-width in a progressive way. Instead of directly quantizing the network to the final scheme of 4-bit weights and 6-bit activations (W4A6), we first quantize it to W32A6. The quantization perturbation from W32A32 to W32A6 is significantly smaller than the perturbation from W32A32 to W4A6, which can intrinsically address the issue of performance drop and gradient explosion. Furthermore, the W32A6 models has significantly higher learning capacity than the W4A6 models, which helps the progressive roadmap from W32A32 to W32A6 to W4A6 to learn smoothly and obtain better-optimized models.

Secondly, when converting the model from W32A6 to W4A6, we propose a stage-wise progressive quantization to gradually reducing the weight precision stage-by-stage. Specifically, we follow the design of BEVFormer (Li et al. 2022) to consider four stages: backbone, neck, encoder, and decoder. We first quantize the backbone to W4A6 while keeping the other stages W32A6, and perform QAT on this mixed-precision model. After the QAT converges, we move on to the next stage of QAT where the neck precision is also reduced. Iterative, we can obtain a fully quantized low-bit network at the end. This fully-quantized model serves as the starting point for our QD-BEV model, after which we further boost the performance using the novel view-guided distillation method.

View-guided distillation

The most distinct characteristics of the BEV networks are the joint process of extracting both image features and BEV features. Therefore, when distilling the BEV networks, it is crucial to jointly consider the two levels of information and combine them organically, rather than simply adding or stacking them together. In the following sections, we present details of our proposed view-guided distillation, the computation of which requires the image feature distillation in , the BEV feature distillation in , and joining the information in .

Image feature distillation Given a pair of aligned teacher and student model, We first compute element-wise distillation loss on image features. We extract the image neck output as the image features to be distilled. To improve the smoothness of the distillation loss, we follow previous attempts (Shu et al. 2021) to use a KL divergence-based distillation loss. Specifically, we consider the flattened image features of the student and the teacher model as logits, which we convert into probability distribution via a softmax function with temperature ϕ_τ, as defined in Eq. (2).

$$\phi_\tau(x_i) = \frac{e^{x_i/\tau}}{\sum_j e^{x_j/\tau}}. \tag{2}$$

Then we calculate the KL divergence of each camera's output separately to achieve the image feature distillation

loss, as in Eq. (3).

$$\mathcal{L}_{img} = \frac{\tau^2}{B \cdot W \cdot H \cdot C} \times \mathcal{D}_{KL}\left(\phi_\tau(F^T_{img}), \phi_\tau(F^S_{img})\right), \tag{3}$$

where B means batchsize, W means imgFeature width, H means imgFeature height, C means imgFeature Channel, and F^T_{img} and F^S_{img} denotes the imgFeature of teacher model and student model respectively.

BEV feature distillation We carried out the second step of distillation on BEV feature. As with image features, the BEV features of student and teacher are first converted into a probability distribution. Then we calculate the KL divergence for each point on the BEV seature, as shown in Eq. (4).

$$\mathcal{L}_{bev} = \frac{\tau^2}{B \cdot C} \times \mathcal{D}_{\mathrm{KL}}\left(\phi_\tau(F^T_{bev}), \phi_\tau(F^S_{bev})\right), \tag{4}$$

where B means batchsize, C means BEVFeature Channel, and F^T_{bev} and F^S_{bev} denotes the BEV Feature of teacher model and student model respectively. We get a loss with shape of $[H_{bev} \times W_{bev}, 1]$.

View-guided distillation objective In the first two sections, we obtained the loss of each camera on the Img feature and the corresponding loss of each point on the BEV feature. On the nuScenes data set, the camera external parameters are known, so we can get the distribution range of each camera corresponding to the BEV feature. Then we generate the mask of views which can be applied to the image feature, that is, M_{bev}, which is the same as defined in BEVFormer (Li et al. 2022). M_{bev} is a tensor with four dimensions: number of cameras, batch size, BEV Size$(H_{bev} \times W_{bev})$, and 3D Height, with binary values in each element. Flattening the last 2 dimensions gives the M_{bev} on the 2d plane. With M_{bev}, the $\mathcal{L}_{img}$ calculated for each camera can be extended to the corresponding loss for each point on the BEV feature, that is, $\widehat{\mathcal{L}}_{img}$.

$$\widehat{\mathcal{L}}_{img} = \mathcal{L}_{img} \odot M_{bev} \tag{5}$$

M_{bev} is calculated from the camera external parameters, and it is a tensor with the shape of $[6, H_{bev} \times W_{bev}]$, which indicates the range of BEVfeature affected by each camera. $\odot$ in crefeq:masked img loss denotes the hadamard product.

Finally, we use $\widehat{\mathcal{L}}_{img}$ to get the final BEV distill objective, View Guided Distillation, which is guided by the image distill information of each view in Eq. (6)

$$\mathcal{L}_{vgd} = \sum_{i=1}^{N \cdot H \cdot W} \widehat{\mathcal{L}}_{img} \odot \mathcal{L}_{bev} \tag{6}$$

Experiments

In this section, we first elaborate on the experimental settings, then we evaluate both PTQ and QAT methods on the BEV networks. Based on the analysis of these results, we propose QD-BEV to overcome shortcomings in PTQ and QAT, and we dedicatedly compare our results with previous works under different settings and constraints.

Table 1: PTQ results with different quantization bitwidth.

W-bit/A-bit	Model	NDS↑	NDS Drop	mAP↑
32/32	Tiny	0.354	-	0.252
	Small	0.479	-	0.370
	Base	0.517	-	0.416
8/8	Tiny	0.351	0.8%	0.248
	Small	0.477	0.4%	0.366
	Base	0.487	5.8%	0.384
6/6	Tiny	0.312	11.9%	0.203
	Small	0.430	10.2%	0.306
	Base	0.402	22.2%	0.262
4/6	Tiny	0.246	30.5%	0.146
	Small	0.369	23.0%	0.228
	Base	0.226	56.3%	0.076
4/4	Tiny	0.034	90.4%	0.001
	Small	0.034	92.9%	0.001
	Base	0.023	95.6%	0.000

Table 2: **QD-BEV results compared to baselines.**

Input Size	Model	Model Size(MB)	BOPS(Tera)	NDS↑	mAP↑
384×1056	BEVDet-R50(Huang et al. 2021)	203.3	94.11	0.381	0.304
450×800	BEVFormer-T(Li et al. 2022)	126.8	62.33	0.354	0.253
	BEVFormer-T-DFQ(Nagel et al. 2019)	31.7	3.90	0.340	0.236
	BEVFormer-T-HAWQv3(Yao et al. 2021)	15.9	1.46	0.348	0.234
	QD-BEV-T (Ours)	15.9	1.46	0.372	0.255
720×1280	BEVFormer-S(Li et al. 2022)	225.6	236.13	0.479	0.370
	BEVFormer-S-DFQ(Nagel et al. 2019)	56.4	14.76	0.467	0.356
	QD-BEV-S (Ours)	28.2	5.53	0.479	0.374
900×1600	DETR3D(Wang et al. 2022)	195.7	520.60	0.425	0.346
	FCOS3D(Wang et al. 2021)	200.3	1028.20	0.415	0.343
900×1600	BEVFormer-B(Li et al. 2022)	262.9	667.39	0.517	0.416
	BEVFormer-B-DFQ(Nagel et al. 2019)	65.7	41.71	0.486	0.384
	QD-BEV-B (Ours)	32.9	15.64	0.493	0.393

QD-BEV Results and Analysis

PTQ results We analyze the influence of different quantization bitwidth on post-training quantization. In Table 1, directly applying PTQ with less than 8-bit precision will lead to a significant accuracy drop, especially when quantized to W4A4 the results become pure noise with around 0 mAP. As can be observed from Table 1, performing QAT is necessary in order to preserve the accuracy while achieving ultra-low bit quantization.

Progressive QAT results Since PTQ cannot achieve ultra-low precision, we aim to apply QAT for 4-bit quantization.

From all our experiments, standard QAT methods which directly quantize the whole network to the target bitwidth will lead to unstable QAT processes, resulting in a gradient explosion and a rapid decline in accuracy.

Consequently, we propose to use progressive QAT to constrain the quantization perturbation along the training process. As an example, we plot the training curve of our progressive QAT in Figure 2, where we conduct W4A6 quanti-

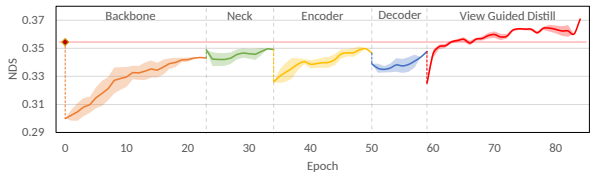
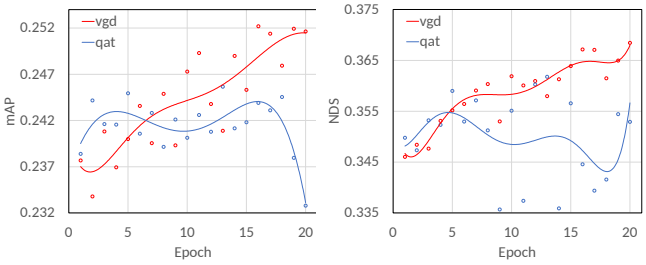


Figure 2: **Progressive QAT.** We observe that the precision loss may become intolerable when we quantize the model in one step. Thus, we propose progressive quantization-aware training which divides the quantization of the whole model into many phases. Progressive QAT can significantly improve the stability of QAT as well as the accuracy it achieves. The pink line in the figure represents the baseline NDS.



(a) mAP curve. (b) NDS curve.

Figure 3: Training curve of view-guided distillation versus QAT on W4A6 quantization of BEVFormer-Tiny.

zation on the BEVFormer-Tiny. We separate the progressive QAT into 4 stages and iteratively quantize a new module in each stage. As can be seen, there is an NDS drop at the beginning of each stage, corresponding to the quantization perturbation introduced by the quantized new module. We should note that progressive QAT is able to effectively recover the NDS drop, and the final results obtained purely with direct QAT is more than 5 points behind that of progressive QAT in our ablation study.

View-guided distillation results Although progressive QAT is significantly superior to standard QAT, it still suffers from fluctuation during training, as shown in Figure 3. In order to obtain better accuracy and stability, we apply view-guided distillation with the floating-point model as the teacher and the quantized model as the student. The effect of VGD on W4A6 quantization of BEVFormer-Tiny is shown in Figure 2. It can be seen from both figures that VGD can bring a very stable and significant improvement to the model. Benefiting from knowledge in both the image domain and the BEV domain, QD-BEV networks are able to fully recover the quantization degradation, and even outperform the floating-point baselines. As shown in Table 2, the NDS and mAP of the model outperform previous floating-point baselines as well as quantized networks. Note that since there are no existing results for compact BEV networks, we implement standard quantization methods DFQ (Nagel et al. 2019) and HAWQv3 (Yao et al. 2021) on BEVFormer as a comparison. We apply W8A8 quantization for DFQ (DFQ with lower bitwidth has intolerable accuracy degradation) and W4A6 for all QD-BEV models. As a comparison, QD-BEV can achieve 0.493 NDS with only 33 MB model size, which is similar to the size of BEVFormer-T-DFQ (0.340 NDS) and much smaller than BEVFormer-Tiny (126.8 MB, 0.354 NDS).

Conclusions

In this work, we pioneer the usage of quantization to obtain compact models for 3D object detection based on BEV. We systematically study both PTQ and QAT on BEV networks and showcase the major problems they are facing. Based on our analyses, in our solution QD-BEV, we propose to apply progressive QAT as well as the newly devised view-guided distillation. QD-BEV addresses the stability problem of previous QAT methods and can alleviate the accuracy degradation or even outperform the floating-point baselines. On the nuScenes datasets, the 4-bit weight and 6-bit activation quantized QD-BEV-Tiny model achieves 37.2% NDS with only 15.8 MB model size, outperforming BevFormer-Tiny by 1.8% with an 8× model compression.

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M. W.; and Keutzer, K. 2021. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

Huang, J.; Huang, G.; Zhu, Z.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Yu, Q.; and Dai, J. 2022. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. *arXiv preprint arXiv:2203.17270*.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022a. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, Q.; Wang, T.; Zhang, X.; and Sun, J. 2022b. PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images. *arXiv preprint arXiv:2206.01256*.

Nagel, M.; Baalen, M. v.; Blankevoort, T.; and Welling, M. 2019. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1325–1334.

Shu, C.; Liu, Y.; Gao, J.; Yan, Z.; and Shen, C. 2021. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5311–5320.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 913–922.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Yao, Z.; Dong, Z.; Zheng, Z.; Gholami, A.; Yu, J.; Tan, E.; Wang, L.; Huang, Q.; Wang, Y.; Mahoney, M.; et al. 2021. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, 11875–11886. PMLR.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.