2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Real-Time BEVFormer: Fast Transformer-based BEV Perception Network on Edge Device

Juyoung Yang*,1, Seoha Baek*,1, Eunbin Seo*,1, Wonseok Jeon*,1, Doyeon Kim*,1, Jongsun Kim[1], Heeyeon Nah†,1

***Abstract*— The development of camera-based real-time 3D perception network for edge devices is essential for embodied systems such as autonomous vehicles and robots. However, existing methods often demand substantial computational resources and tend to overlook performance on resource-constrained devices. In this paper, we propose RT-BEVFormer, a simple yet effective multi-task 3D perception framework designed for efficiency. Based on BEVFormer, RT-BEVFormer enhances the feature extraction capability of the backbone and redesigns the spatial cross-attention module in the encoder, guided by two key observations: 1) the computational load and total number of parameters are dominated by the backbone, and 2) the sampling process within the deformable attention module is a primary bottleneck. Specifically, we leverage powerful foundation models to distill their rich and comprehensive knowledge, thereby crafting a highly efficient student backbone. This allows RT-BEVFormer to achieve significant performance gains without incurring additional latency. Furthermore, we introduce an efficient static sampling method. This approach replaces the dynamic and deployment unfriendly nature of standard spatial cross-attention, allowing the model to focus on salient image features with minimal overhead. On the widely-used edge device, NVIDIA Jetson Orin, RT-BEVFormer outperforms the previous state-of-the-art model in both accuracy and inference speed. Extensive experiments on the nuScenes dataset show that each component of our framework is effective in both inference speed and overall accuracy. Finally, as RT-BEVFormer is implemented without any model-specific custom plugin, it ensures superior flexibility and ease of deployment.**

I. INTRODUCTION

Bird's-Eye-View (BEV) perception constructs a unified view of the surrounding environment by integrating visual information from diverse sensors. As BEV perception networks can perform multiple tasks like 3D object detection [1], BEV segmentation [2], or planning [3], they have been actively studied and applied in embodied systems such as autonomous vehicles and robots. Given the need for rapid adaptation to dynamic environments in these applications, achieving real-time inference on lightweight and edge-oriented devices is essential. In particular, camera-based BEV perception has become an active research area due to its ability to enable environmental perception without relying on expensive sensors such as LiDAR [1], [4], [5].

The process of forming a unified BEV representation from multi-view images can be broadly categorized into two main paradigms: (1) 3D projection, which integrates multi-view features into a predefined 3D coordinate volume, and (2) 2D unprojection, which projects 2D points into 3D space using explicit depth estimation. The former typically leverages transformer-based attention mechanisms to construct unified representation, while the latter utilizes feature aggregation techniques such as BEV pooling. Each method comes with its own trade-offs: the former, while achieving superior performance, tends to be computationally impractical on edge devices; the latter is relatively less accurate but is more computationally efficient. However, both approaches have barely been explored from the perspective of deployment on edge devices. We posit that the following challenges have limited research into deploying BEV perception networks.

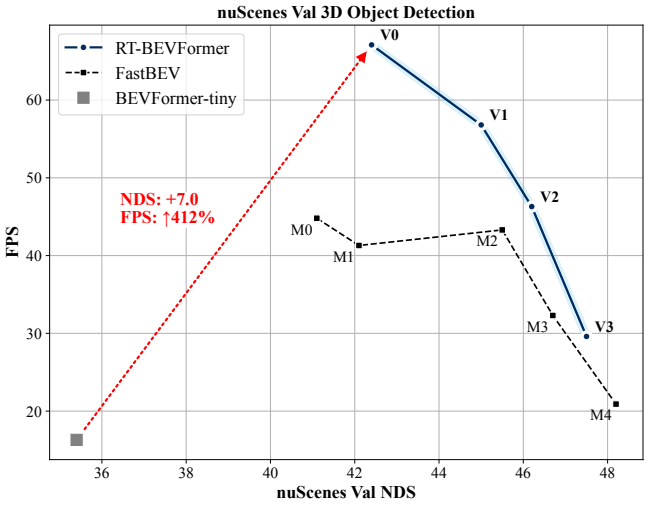


Fig. 1: Comparison with BEVFormer-tiny [1] and previous real-time 3D BEV perception network FastBEV [6] on NVIDIA Orin. Frames Per Second (FPS) of BEVFormer-tiny is measured using custom implementation[§].

First, in transformer-based approaches, attention mechanisms cause the primary inference bottleneck. Analysis of existing networks shows that, while transformer modules account for only a fraction of total parameters and Multiply-Accumulate operations (MACs), they dominate overall latency. This tendency is especially exaggerated on edge devices, where memory bandwidth is relatively limited. Second, the reliance of 2D unprojection-based approaches on device-specific computation kernels (e.g., CUDA) creates a strong hardware dependency, hindering their deployment on diverse hardware platforms. Even for NVIDIA devices,

* These authors contributed equally and are listed randomly.
† Corresponding author.
[1] Semiconductor Development Group, Hyundai Motor Company, 117, Bundangnaegok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

[§] https://github.com/DerryHub/BEVFormer_tensorrt

exporting PyTorch models to deployment-friendly formats such as ONNX and TensorRT typically requires hundreds of lines of code. To our knowledge, FastBEV [6] is the only work explicitly targeting the on-vehicle deployment of BEV perception. FastBEV proposes a fast BEV transformation using look-up tables instead of BEV pooling and adopts a CNN-based architecture, resulting in a network suitable for edge deployment. However, is real-time inference on edge devices truly impractical for transformer-based BEV perception networks?

In this work, we introduce RT-BEVFormer, an efficient real-time BEV perception network built upon BEVFormer [1], a seminal work in the 3D projection paradigm. Our framework achieves real-time performance on edge devices while improving the accuracy of the network. To begin with, by adopting a 3D projection approach free from BEV pooling, our BEV feature construction avoids device-specific operations. Next, we have identified two key improvement directions that significantly boost performance while reducing latency. First, we enhance the backbone to improve accuracy without increasing inference latency. The backbone accounts for over half of total network parameters and over 90% of MACs; as such, backbone dominates overall performance. Building upon this, we propose constructing a more powerful backbone using RADIO [7], recent foundation model research. Second, we target and improve the main bottlenecks of BEVFormer on edge devices. Spatial cross-attention in the encoder involves sampling values from multi-camera features, while accounting for the dynamic characteristics of the cameras. However, such dynamic properties introduce additional overheads and complicate deployment in current deep learning systems [8]. We address this by proposing an efficient static sampling method, resolving both the deployment and latency issues arising from sampling.

RT-BEVFormer shows a superior real-time performance for its detection accuracy compared to the previous state-of-the-art, FastBEV. As illustrated in Fig. 1, it is verified that across various scales, RT-BEVFormer achieves substantially higher FPS while maintaining comparable detection performance when compared to FastBEV on the NVIDIA Orin. Our results demonstrate that, despite its transformer architecture, RT-BEVFormer achieves a superior performance-efficiency trade-off compared to the CNN-based network. Furthermore, RT-BEVFormer achieves significant results in BEV segmentation, highlighting its broad potential for various downstream applications. Finally, by minimizing reliance on specialized functions throughout the forward process, our model is more readily deployable across diverse devices. Deployment on Orin or Xavier can be accomplished by simply using `torch.onnx.export` and `trtexec`.

II. RELATED WORKS

A. Camera-based BEV Perception Networks

The BEV space obtained from multiple cameras contains integrated information about the surrounding environment, enabling various perception tasks such as detection [1], [9] and segmentation [2], [10]. Lift-Splat-Shoot (LSS) [11] introduces a foundational BEV perception approach that predicts a depth distribution for each image pixel, lifts 2D image features into a 3D frustum, and splats these features onto the BEV plane to construct BEV feature maps. Building on this method, BEVDet [12] develops a pipeline for 3D object detection in BEV space, which is further extended by BEVDet4D [9] through the incorporation of temporal information.

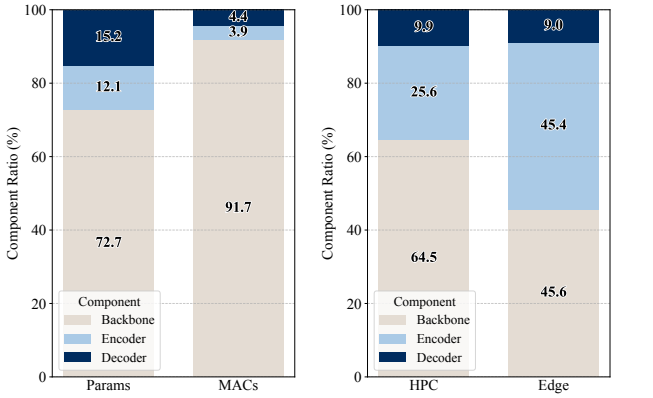
Transformer-based approaches have also emerged rapidly in this field. BEVFormer [1] is one of the pioneering studies in applying transformers for BEV perception. This method fuses multi-view image features and previous BEV frames using cross-attention and temporal-attention mechanisms based on learnable BEV queries. This enables spatio-temporally consistent BEV feature representations and achieves state-of-the-art 3D detection performance.

Since the performance of LSS-based models heavily depends on depth estimation, subsequent research has focused on improving this aspect. BEVDepth [13] enhances BEV features by leveraging depth supervision provided by LiDAR-based depth labels within the LSS framework. Recently, GaussianLSS [14] has been proposed as an extension of the LSS concept that explicitly models depth uncertainty using Gaussian splatting techniques, resulting in more accurate and robust BEV representations.

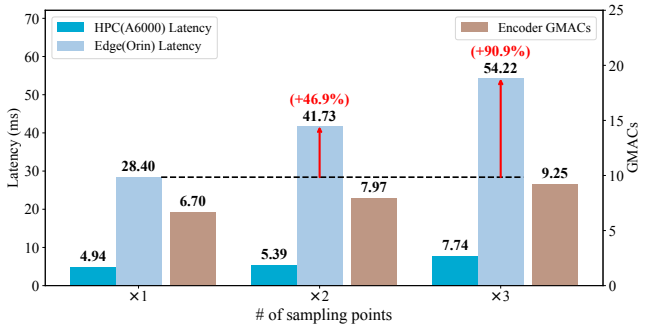
B. Efficient Camera-based BEV Perception Networks

Driven by the development of autonomous driving and robotics, the need for computationally efficient perception models has become critical. To address this efficiency challenge, several approaches have been proposed. MatrixVT [15] presents an efficient structure that uses convolution and matrix multiplication to transform multi-camera images into BEV space, and introduces modules such as Feature Transporting Matrix, Prime Extraction, and Ring & Ray Decomposition to reduce computational cost and memory usage. PolarBEV [16] divides the BEV space into polar grids based on angle and radius, applying polar rasterization and embedding decomposition to improve inference speed and performance. BEVFusion [10] increases computational efficiency for large-scale inputs by accelerating the camera-to-BEV transformation with a custom CUDA kernel, enabling fast inference in multi-sensor environments.

More recently, a new direction in edge-oriented research has emerged, extending beyond conventional efficiency-focused methods. FastBEV [6] demonstrates fast inference and deployability on various edge devices by using a lightweight structure and an efficient view transformation, without complex transformer operations. While such study is still scarce, developing models that achieve both efficiency and practicality in diverse environments is becoming an important challenge for real-world deployment of BEV perception techniques. In this work, we enhance the backbone with foundation models and propose efficient sampling for fast transformer inference on edge devices.



(a) Model Complexity Analysis (b) Latency Profiling



(c) Impact of Sampling Points on Encoder

Fig. 2: **Complexity analysis and profiling of BEVFormer on HPC and edge device.** Component analysis and profiling of BEVFormer reveals the computational bottleneck in dynamic encoder sampling and the gap between theoretical complexity and actual latency on edge devices. Latency is measured using TRT-FP32-Orin setting. (a) Parameters and MACs distribution among backbone, encoder, and decoder. (b) Latency analysis by model component on HPC and edge hardware. (c) Encoder latency increases sharply with more sampling points, especially on edge devices.

III. METHOD

A. Analysis of BEVFormer

Fig. 2 shows the profiling results from BEVFormer on a High Performance Computing (HPC)-oriented NVIDIA GPU A6000 and an edge-oriented SoC NVIDIA Jetson Orin. To deploy BEVFormer, we perform the following process. BEVFormer consists of a backbone that extracts image features, an encoder that generates BEV features, and task-specific decoders; the temporal attention and spatial attention inside the encoder are where the main attention mechanisms reside. In particular, the spatial cross-attention receives the 3D-to-image calibration information of each camera and projects the evenly distributed points in 3D BEV space onto the image space. Because the sampling locations for these points slightly differ for each camera and sample, new queries and reference points are generated dynamically to calculate attention values. However, this dynamic nature makes the deployment of the model challenging (Fig. 4-(a), Dynamic Sampling). As a workaround, we analyze an approach (Fig. 4-(b), Static Masking) in which the calculation is done for the entire BEV space, then a mask is generated for each camera so that only the corresponding values can be selectively gathered, thus facilitating deployment.

In Fig. 2-(a) and (b), we can see the backbone accounts for more than 70% of the total parameters and more than 90% of all MACs, but it only consumes 64.5% of the total inference time on HPC. Notably, its proportion is even lower on the edge device, accounting for only 45.6%. By contrast, the encoder, which consists of several attention layers, occupies only a small fraction of the total parameters (12.1%) and MACs (3.9%), but its share of total latency on HPC (25.6%) is significant, and surges to as high as 45.4% on the edge device. We hypothesize that the grid sampling process within deformable attention is responsible for this phenomenon.

To verify that the sampling process significantly impacts overall latency relative to computational load, Fig. 2-(c) examines how the encoder's computation time and MACs vary as the number of sampling points increases. Using the smallest sampling point setting as a baseline, doubling and tripling the number of sampling points results in increased encoder MACs by 19.0% and 38.1%, respectively. However, latency rises by 46.9% and 90.9% correspondingly, a much steeper rate, especially on edge devices compared to the server-side. This indicates that as the number of sampling points grows, the associated latency increases far more rapidly than the computational requirements, especially on edge devices. As noted in [17], this is attributed to irregular memory access patterns and high cache miss rates induced by the grid sampling process. Compared to HPC-oriented systems, edge devices inevitably experience this as the dominant source of latency because of their lower memory bandwidth. Therefore, efficient sampling process in deformable attention is not an optional consideration for fast and efficient inference on edge devices.

In the following section, based on two key observations: 1) the backbone accounts for the majority of the parameters and most of the computations, and 2) the sampling process constitutes a significant bottleneck, we propose a network design that achieves faster speed and higher performance.

B. Overall Structure

This section details the proposed RT-BEVFormer framework. We adopt BEVFormer as our baseline, featuring an encoder enhanced with an efficient static sampling mechanism. RT-BEVFormer consists of three main components as shown in Fig. 3: a backbone, a BEV encoder, and a task-specific decoder (i.e., detection or segmentation decoder). The backbone takes multi-camera images as input and extracts features from images. We leverage a foundation model as a teacher to impart its powerful feature extraction capabilities to our backbone via knowledge distillation (KD) [18]. Subsequently, the BEV encoder utilizes a set of BEV queries to aggregate information from the multi-view features, effectively forming a unified BEV representation. The BEV encoder's architecture consists of three key modules: temporal self-attention,

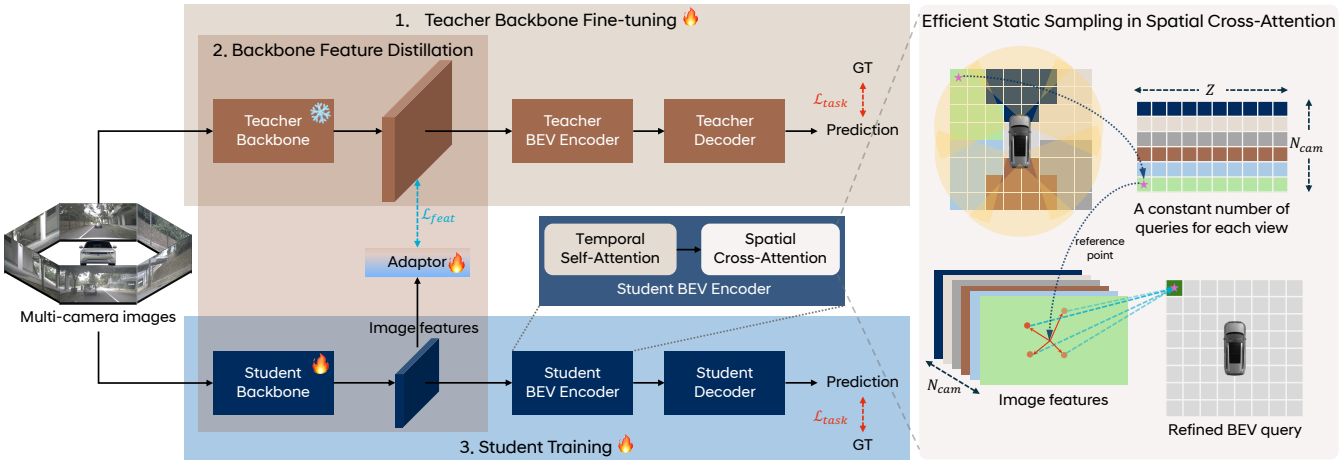


Fig. 3: **An overview of the proposed RT-BEVFormer**. Left: Three steps are followed to build a strong backbone: 1) The teacher backbone is trained for the target task, 2) Using the trained teacher backbone, a more efficient student backbone is trained with a loss at the image feature level, while the teacher backbone is frozen, 3) The lightweight student backbone, which has distilled knowledge from the teacher backbone, is further trained for the target task. Right: By sampling image features at predefined regions from N_{cam} cameras, latency is reduced and a static set of queries is generated, making deployment easier.

spatial cross-attention, and a feed-forward network (FFN). The spatial cross-attention module uses a deformable cross-attention mechanism to enrich the BEV queries with spatial context from the multi-view image features. In this process, we introduce an efficient static sampling strategy that uses a constant number of queries per camera to ensure efficient inference. The resulting features are then passed to a task-specific decoder head to generate the final predictions.

The entire framework, as illustrated in Fig. 3, consists of a three stage process: 1) teacher backbone fine-tuning. 2) backbone feature distillation, and 3) student model training.

C. Backbone Capability Boosting

As indicated by the analysis in Section III-A, the backbone constitutes a significant portion of the overall model in terms of parameter count and inference time. Being a major component of the model's capacity, it profoundly influences the final task performance. Indeed, previous studies [4], [13], [19], [20] have demonstrated performance gains by leveraging extra supervision or prior knowledge of pre-trained backbones with extra data. Thus, the feature extraction capability of the backbone plays a pivotal role in enhancing the model's overall task performance.

Recently, vision foundation models trained on internet-scale data have been proposed [21], [22], [23], [24]. These models acquire generalized feature extraction capabilities from massive datasets, enabling them to exhibit superior performance on various downstream tasks. In particular, RADIO [7], an agglomerative vision foundation model trained via multi-teacher distillation from foundation models, achieves a superior ability to extract high-quality features compared to its predecessors. However, the large size of foundation models makes them impractical for deployment on edge devices. We address this challenge by imparting the advanced feature extraction capabilities of a vision foundation model to a compact student backbone via KD. By utilizing this distilled student backbone, we develop a high-performance BEV perception network that is efficient enough for edge deployment.

We select C-RADIOv3-B, a recent model from the RADIO family, as our teacher model. First, we fine-tune the teacher on the BEV perception task, following the standard training procedure [1]. Subsequently, the feature extraction capability of the fine-tuned teacher backbone is distilled into the student backbone through simple feature distillation.

Let $x \in \mathbb{R}^{C\times H\times W}$ be the input image, where C, H, and W represent its channel, height, and width respectively. The teacher backbone f_t extracts a teacher feature map $z_t = f_t(x) \in \mathbb{R}^{d_t\times H_t\times W_t}$, while the student backbone f_s extracts a student feature map $z_s = f_s(x) \in \mathbb{R}^{d_s\times H_s\times W_s}$. To align the spatial dimensions, the teacher feature map z_t is resized to match the student's spatial dimensions ($H_s \times W_s$) via bilinear interpolation (Eq. (1)). The student feature map z_s is then processed by an adaptor g to match the teacher's channel dimension d_t (Eq. (2)). This adaptor, similar to that in RADIO, is composed of a Linear - Layer Normalization - GELU - Linear architecture.

$$\tilde{z}_t = \text{resize}(z_t) \in \mathbb{R}^{d_t\times H_s\times W_s} \tag{1}$$

$$\tilde{z}_s = g(z_s) \in \mathbb{R}^{d_t\times H_s\times W_s} \tag{2}$$

The objective of KD is to align the student's feature map $\tilde{z}_s$ with the teacher's feature map $\tilde{z}_t$. To achieve this, the student backbone is trained to minimize the mean squared error between the two features. The loss function $\mathcal{L}_{feat}$ for feature distillation is defined as:

$$\mathcal{L}_{feat} = ||\tilde{z}_t - \tilde{z}_s||_2^2 \tag{3}$$

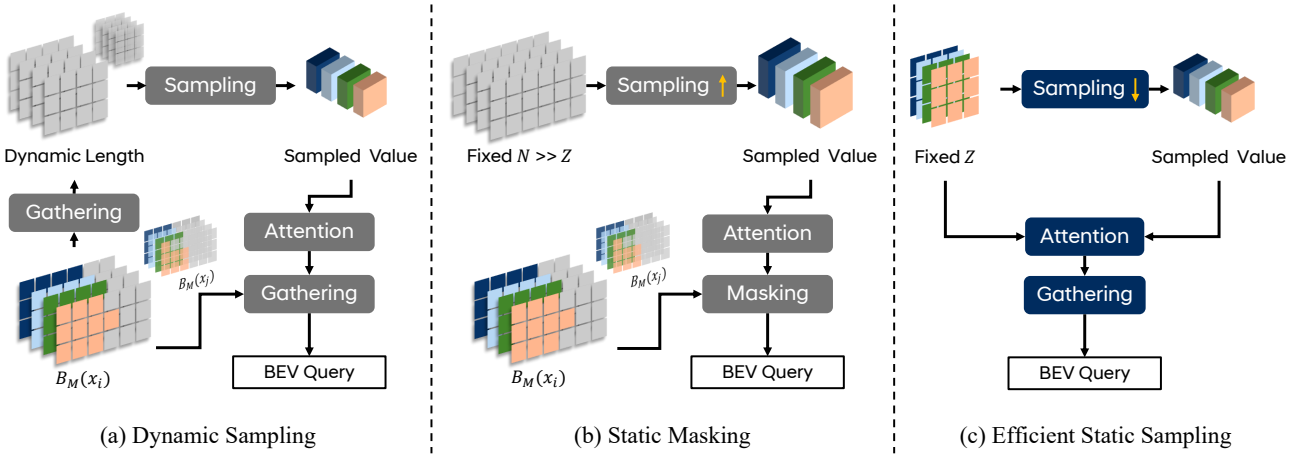


Fig. 4: **Detailed description of three types of spatial cross-attention.** (a) Dynamic Sampling represents original BEVFormer [1] case which defines the size of BEV mask with the maximum length from valid camera reference points. (b) Static Masking fixes the size of BEV Mask $B_M(x)$ to $H_{BEV} \times W_{BEV}$ to avoid dynamic nature and enable easy deployment, but it increases the amount of sampling points. (c) Efficient Static Sampling greatly reduces the number of expensive sampling operations and simplifies the whole process. (a) and (b) both require calculating B_M for different input image x_i and x_j.

D. Efficient Static Sampling

As discussed in Section III-A, one of the major factors hindering efficient inference in recent transformer-based models is grid sampling within deformable attention. Particularly in BEVFormer, the sampling process within the encoder's spatial cross-attention constitutes a significant bottleneck. It consumes considerable time during inference and causes the BEV query tensor to change dynamically as shown in Fig. 4-(a), which inevitably results in inefficient inference. This raises the question: is it really necessary to specify and compute the camera-derived point for every single sample? Given that the camera positions and corresponding parameters are fixed upon the construction of the perception system, it can be reasonably assumed that it is unnecessary to compute all points at every time step. To investigate, we conduct an experiment where the parameters obtained from the first sample was fixed during training, and compare the results with scenarios where different parameters were dynamically applied at every time step. The comparison shows that there is no significant difference in performance between the dynamic (37.5 NDS) and fixed (37.3 NDS) approaches. In other words, simply using the precomputed BEV mask is as effective as reflecting the mask dynamically.

By fixing this number to a certain value, we can reduce the size of the BEV embedding space used as queries in deformable attention. Through statistical analysis, we find that the number of points per camera view in an $H_{BEV} \times W_{BEV}$ BEV space is typically distributed between $(H_{BEV} \times W_{BEV})/6$ and $(H_{BEV} \times W_{BEV})/4$, with most cameras clustered near $1/6$ and the front view approaching $1/4$ of the space. Based on this, we devise a method to reduce the number of sampling operations and sample the points statically. First, select a BEV mask B_M randomly from a sample to serve as the reference. From this sample mask, determine θ_{start} and θ_{end} denoting the camera's viewing angles. Then, count the number of points between θ_{start} and θ_{end} in the BEV space. If this number is less than the targeted pixel number Z, increase θ_{start} and θ_{end} by a fixed interval δ; if it is greater, decrease them by δ. Repeat this process until the desired value Z is reached for all cameras, resulting in a point mask P_M containing a fixed number Z of points for each camera. Thus, by reducing the total number of sampling and simplifying the overall spatial cross-attention process, we can achieve more efficient inference (Fig. 4-(c)).

One potential concern is that assigning a fixed number of points without considering the viewing angles of each camera could cause problems. However, our experiments in Section IV-E show that fixing the points has little impact on accuracy while dramatically improving latency. This is likely because the regions near the ego vehicle—which are always within view—are the most important, while the parts outside the camera's view tend to be those farther from the ego vehicle and of relatively lower importance. By applying efficient static sampling, we significantly improve latency compared to BEVFormer by optimizing several elements that are redundantly sampled and have a large impact on latency, all without affecting performance.

IV. EXPERIMENTS

A. Implementation Details

Dataset and Metrics. We evaluate RT-BEVFormer using the nuScenes dataset [25]. Evaluation of 3D object detection task is based on the official metrics, including nuScenes Detection Score (NDS) and mean Average Precision (mAP).

Name	Backbone	Image Size	BEV Size	Static Points	# of Dec	NDS	mAP	Orin(ms)	Xavier(ms)
V0	ResNet18	480×800	50×50	500	3	42.4	28.8	14.9	40.8
V1	ResNet34	480×800	50×50	500	3	45.0	32.3	17.6	50.6
V2	ResNet50	480×800	50×50	500	3	46.2	33.7	21.6	61.6
V3	ResNet50	480×800	75×75	1125	6	47.5	35.1	33.8	95.7

TABLE I: Detailed design specifications for the various variants of our model. Latency is measured using TRT-INT8 on NVIDIA Orin (170 TOPS without DLA) and Xavier (21 TOPS without DLA).

Methods	NDS↑	mAP(%)↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
BEVFormer-Tiny [1]	35.4	25.2	0.900	0.294	0.655	0.657	0.216
BEVDet-R50 [12]	38.9	31.8	0.718	0.272	0.553	0.897	0.258
PETR [4]	40.3	33.9	0.748	0.273	0.539	0.907	0.203
MatrixVT [15]	41.5	33.7	0.653	0.271	0.473	0.903	0.231
PETRv2 [5]	45.6	35.0	0.726	0.277	0.505	0.503	0.181
FastBEV-R50 [6]	47.3	33.4	0.665	0.285	0.393	0.388	0.210
BEVDepth† [13]	47.5	35.1	-	-	-	-	-
Ours (V3)	47.5	35.1	0.744	0.282	0.416	0.374	0.192

TABLE II: Comparisons with other 3D Object Detection methods. All methods use ResNet50 as a backbone and are trained with CBGS except BEVFormer-tiny. †: Trained with additional depth information.

Experimental Settings. RT-BEVFormer adopts most of the training configuration as BEVFormer [1]. We train the models for 18 epochs on 16 A6000 GPUs with an initial learning rate of 5e-4, employing CBGS [26] and EMA [27] strategies. These strategies are omitted for ablation studies to isolate the effects of other components. To ensure its optimal performance, the V3 model is further trained up to 26 epochs with SWA [28], omitting both CBGS and EMA. Performance evaluation is carried out under the same GPU environment as described above, while inference speed is measured on the edge device NVIDIA Jetson Orin using CUDA 12.6, TensorRT 10.3.0, and INT8 precision except for Fig. 2.

B. Deployment on the Edge Device

Fig. 1 presents a performance of RT-BEVFormer variants against the BEVFormer on an edge device. Notably, our V0 model, utilizing ResNet18 backbone, achieves a 7.0 point improvement in NDS. It validates the efficacy of enhancing the backbone's expressiveness via KD, as proposed in Section III-C. Furthermore, the inference throughput is boosted by 412%, confirming that the efficient static sampling method introduced in Section III-D successfully alleviates the key computational bottleneck.

Furthermore, a scale-wise comparison reveals that RT-BEVFormer is significantly faster than FastBEV while delivering better detection performance. For instance, RT-BEVFormer-V0 shows 42.4 NDS and 67 FPS, surpassing FastBEV-M0 (41.1 NDS and 45 FPS), and this trend of superior speed with better accuracy holds across all model scales. These indicate that the backbone enhancement and efficient encoder of RT-BEVFormer not only lead to improvements in detection accuracy but also contribute to reduced latency. Moreover, RT-BEVFormer does not rely on custom functions such as CUDA kernels, enabling rapid and straightforward edge deployment, which is a challenge for other BEV perception models. To the best of our knowledge, this makes RT-BEVFormer the first algorithm among transformer-based 3D projection object detectors to achieve seamless deployment.

C. Model Variants

The variants of our model are presented in Table I. For RT-BEVFormer-V0 through V2, the BEV size is fixed at 50×50 while the backbone architecture is varied. Correspondingly, detection performance increases by 42.4 NDS and 46.2 NDS, respectively, while latency on Orin also increases by 14.9 ms and 21.6 ms. The V3 builds upon V2 by increasing both the BEV size and the number of static points.

D. Comparison with other 3D Object Detection Methods

To compare the detection performance with other 3D object detection methods, all models are implemented using a ResNet-50 backbone. Except for BEVFormer-tiny, all methods are trained using CBGS. The results are presented in Table II. As shown, RT-BEVFormer achieves 47.5 NDS and 35.1 mAP, demonstrating superior detection performance compared to BEVDet-R50 [12], PETR [4], MatrixVT [15], and PETRv2 [5]. This trend is also observed when compared to BEVDepth [13], which incorporates depth information during training; RT-BEVFormer achieves competitive detection performance without explicit depth supervision. This result verifies that RT-BEVFormer shows enhanced accuracy while achieving real-time inference speed on edge device.

E. Ablation Study

In this section, we conduct several ablation experiments to show the effectiveness of our proposed approaches.

Methods	NDS	mAP
ImageNet [29]	38.8	27.4
FCOS3D [19]	41.4	29.5
BEVDepth [13]	41.7	31.5
NuImage [25]	42.9	31.9
Ours	**44.9**	**32.8**

TABLE III: Ablation study of the backbone. All methods use pretrained ResNet50 with each setting.

Methods	Baseline		Efficient Static Sampling				
	(a)	(b)	400	450	500	550	600
NDS	37.5	37.2	36.0	37.2	37.5	37.8	37.6
Latency(ms)	-	37.3	7.8	8.2	8.5	9.0	9.4

TABLE IV: Impact of the number of static points. The first and second rows represent the NDS and the encoder-only latency, respectively. (a): Fig. 4-(a) Dynamic Sampling. (b): Fig. 4-(b) Static Masking. The BEV Space is set to 50×50 and ResNet34 is used as the backbone.

Backbone Capability Boosting. To assess the effect of different backbone pre-training strategies on 3D object detection performance, we conduct a thorough ablation study comparing five ResNet50 initialization variants and present the results in Table III. Using ImageNet [29] pre-trained ResNet50 as our baseline, we observe that this widely adopted approach offers robust and generic image representations, but lacks explicit geometric cues required for BEV-based detection tasks. Incorporating depth-aware pre-trained ResNet50 via BEVDepth [13] substantially improves performance over the baseline, indicating the value of spatial-geometric feature learning. The model initialized with FCOS3D [19] pre-trained weights, commonly used in existing works such as BEVFormer [1] and PETR [4], also shows considerable advantages. Similarly, the model initialized with pre-trained weights obtained from NuImage [25] achieves high performance due to the advantage of a much larger data pool compared to the nuScenes. The most effective strategy is our proposed method, which leverages KD from a pre-trained C-RADIOv3-B backbone. As demonstrated in Table III, our method significantly surpasses previous approaches by transferring both high-level semantic and detailed geometric information from a powerful teacher to the student backbone.

Effectiveness of Static Sampling. To verify the effectiveness of the static sampling and the influence of the number of static points for covering the surrounding view of the ego vehicle, we perform experiments with different settings. As the number of points in the BEV space usually report 1/6-1/4 of overall space, we choose several numbers from this range to validate. As shown in Table IV, the performance decreases when the value is set to the smallest, but as the value increases, the performance improves. We can observe that the performance stabilizes and shows similar results around 500 points. This is likely because a value that is too small does not sufficiently cover the space. However, if the static points sufficiently covers an area similar to that covered by the original BEVFormer, we observe that the performance is comparable to or better than the original method.

We choose 500 static sampling points as a trade-off between accuracy and latency. When applying this static sampling with 500 static points, the encoder's latency is dramatically reduced from 37.3 ms to 8.5 ms compared to the baseline (Fig. 4-(b)), a $\times$4.39 speedup. From this result, we can see that by choosing an appropriate static point value, it is possible to configure the BEV space using deformable attention while maintaining accuracy and achieving faster inference and easier deployment.

Method	Drivable	Ped. Cross.	Walkway	Divider
LSS [11]	0.754	0.388	0.463	0.365
CVT [2]	0.743	0.368	0.399	0.294
GaussianLSS [14]	0.763	**0.463**	0.502	0.387
BEVFormer-base [1]	0.801	-	-	0.257
Ours	**0.803**	0.440	**0.506**	**0.390**

TABLE V: Comparison of BEV semantic segmentation performance on the nuScenes validation set.

F. Segmentation

To demonstrate that our proposed encoder architecture is not limited to a specific task but can be applied to various BEV downstream tasks by learning BEV feature representations, we further evaluate it on BEV segmentation. In this experiment, we reuse the encoder with ResNet34 backbone from the detection model and design a decoder composed of CNN and pixel shuffle [30] layers to generate segmentation maps. The nuScenes dataset is employed for training, and the loss function is defined as the weighted sum of sigmoid focal loss and mIoU loss with equal weights.

Table V presents a quantitative comparison of BEV segmentation performance between our model and existing approaches. The performance is measured on the four semantic layers, following GaussianLSS [14] which also uses the same settings for range and resolution. Our model consistently achieves strong and competitive results across all major classes. Specifically, it records the highest IoU scores on drivable area, walkway, and divider, while the pedestrian crossing remains competitive against recent works such as GaussianLSS. Moreover, despite using a lightweight backbone, our model surpasses BEVFormer with its heavier ResNet101-DCN backbone and demonstrates superior performance compared to other efficiency-oriented methods. These results highlight that the proposed architecture and methodology are not confined to detection but also generalize effectively to segmentation, providing a favorable balance between accuracy and computational efficiency.

V. CONCLUSIONS

We introduce RT-BEVFormer, a novel BEV perception framework that achieves real-time inference speed on

the widely used edge device NVIDIA Jetson Orin. RT-BEVFormer leverages its perception ability by employing a powerful backbone which empowered by recent prominent foundation model, to incorporate high-quality features obtained from massive and diverse datasets. Furthermore, our efficient static sampling strategy eliminates the excessive computational cost associated with redundant dynamic sampling, while maintaining the network's overall accuracy. Consequently, RT-BEVFormer achieves remarkable performance in both accuracy and latency. Additionally, deployment of our network on the edge device is streamlined, requiring minimal additional code compared to previous works, which greatly lowers the technical barrier for practitioners who may not be deeply familiar with the code. Given that RT-BEVFormer builds upon the architecture of BEVFormer, we hope and expect that RT-BEVFormer can become a new baseline for transformer-based BEV perception networks on edge devices, and can be further advanced by integrating such techniques [31], [32] in future work.

References

[1] 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*, 2022, pp. 1–18.

[2] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 760–13 769.

[3] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 17 853–17 862.

[4] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *European conference on computer vision*. Springer, 2022, pp. 531–548.

[5] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, and X. Zhang, "Petrv2: A unified framework for 3d perception from multi-camera images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3262–3272.

[6] B. Huang, Y. Li, E. Xie, F. Liang, L. Wang, M. Shen, F. Liu, T. Wang, P. Luo, and J. Shao, "Fast-bev: Towards real-time on-vehicle bird's-eye view perception," *arXiv preprint arXiv:2301.07870*, 2023.

[7] G. Heinrich, M. Ranzinger, H. Yin, Y. Lu, J. Kautz, A. Tao, B. Catanzaro, and P. Molchanov, "Radiov2. 5: Improved baselines for agglomerative vision foundation models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 487–22 497.

[8] K. Zhu, W. Zhao, Z. Zheng, T. Guo, P. Zhao, J. Bai, J. Yang, X. Liu, L. Diao, and W. Lin, "Disc: A dynamic shape compiler for machine learning workloads," in *Proceedings of the 1st Workshop on Machine Learning and Systems*, 2021, pp. 89–95.

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

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

[11] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European conference on computer vision*. Springer, 2020, pp. 194–210.

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

[13] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 2, 2023, pp. 1477–1485.

[14] S.-W. Lu, Y.-H. Tsai, and Y.-T. Chen, "Toward real-world bev perception: Depth uncertainty estimation via gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 17 124–17 133.

[15] H. Zhou, Z. Ge, Z. Li, and X. Zhang, "Matrixvt: Efficient multi-camera to bev transformation for 3d perception," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8548–8557.

[16] H. Yang, X. Bai, X. Zhu, and Y. Ma, "One training for multiple deployments: Polar-based adaptive bev perception for autonomous driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5602–5609.

[17] Y. Xu, D. Lyu, Z. Li, Y. Chen, Z. Wang, G. Wang, Z. Wang, H. Li, and G. He, "Defa: Efficient deformable attention acceleration via pruning-assisted grid-sampling and multi-scale parallel processing," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024, pp. 1–6.

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

[19] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 913–922.

[20] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?" in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3142–3152.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[22] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[23] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[24] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 10 371–10 381.

[25] 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.

[26] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[27] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[28] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[30] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[31] X. Ye, B. Yaman, S. Cheng, F. Tao, A. Mallik, and L. Ren, "Bevdiffuser: Plug-and-play diffusion model for bev denoising with ground-truth guidance," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1495–1504.

[32] C. Pan, B. Yaman, S. Velipasalar, and L. Ren, "Clip-bevformer: Enhancing multi-view image-based bev detector with ground truth flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 216–15 225.