DQFormer: Toward Unified LiDAR Panoptic Segmentation With Decoupled Queries for Large-Scale Outdoor Scenes

Yu Yang, Jianbiao Mei, Siliang Du, Yilin Xiao, Huifeng Wu, *Member, IEEE*, Xiao Xu, and Yong Liu, *Member, IEEE*

***Abstract*— LiDAR panoptic segmentation (LPS) performs semantic and instance segmentation for *things* (foreground objects) and *stuff* (background elements), essential for scene perception and remote sensing. While most existing methods separate these tasks using distinct branches (i.e., semantic and instance), recent approaches have unified LPS through a query-based paradigm. However, the distinct spatial distributions of foreground objects and background elements in large-scale outdoor scenes pose challenges. This article presents DQFormer, a novel framework for unified LPS that employs a decoupled query workflow to adapt to the characteristics of things and stuff in outdoor scenes. It first utilizes a feature encoder to extract multiscale voxel-wise, point-wise, and bird's eye view (BEV) features. Then, a decoupled query generator proposes informative queries by localizing things/stuff positions and fusing multilevel BEV embeddings. A query-oriented mask decoder uses masked cross-attention to decode segmentation masks, which are combined with query semantics to produce panoptic results. Extensive experiments on large-scale outdoor scenes, including the vehicular datasets nuScenes and SemanticKITTI, as well as the aerial point cloud dataset DALES, show that DQFormer outperforms superior methods by +1.8%, +0.9%, and +3.5% in panoptic quality (PQ), respectively. Code is available at https://github.com/yuyang-cloud/DQFormer**

***Index Terms*— Decoupled queries, large-scale outdoor scenes, LiDAR panoptic segmentation (LPS), point cloud segmentation.**

I. Introduction

SCENE perception and understanding are fundamental in geoscience and remote sensing. With advancements in 3-D data acquisition techniques, LiDAR point clouds have become the primary resource for collecting large-scale geospatial data, revealing detailed geometric structures of real-world 3-D environments. As a key task in scene perception, LiDAR segmentation involves point-level predictions to interpret the entire scene. Among these tasks, LiDAR panoptic segmentation (LPS) predicts not only point-wise semantic labels for *stuff* classes (e.g., roads and vegetation) but also labels and instance IDs for *thing* classes (e.g., cars and people). By unifying semantic and instance segmentation within a single architecture, LPS plays a crucial role in scene understanding and has a wide range of applications, such as autonomous driving, urban modeling, and remote sensing.

Most existing LPS methods [1], [2], [3], [4], [5] explicitly separate semantic and instance segmentation tasks, utilizing two branches to implement panoptic segmentation. As illustrated in Fig. 1(c), the semantic branch predicts semantic labels for each point, while the instance branch employs detection or clustering techniques to assign instance IDs. Inspired by the recent success of query-based methods in the 2-D segmentation domain [6], [7], [8], [9], [10], MaskPLS [11] and PUPS [12] propose using a set of learnable queries to achieve unified LPS. This approach predicts a set of nonoverlapping binary masks and semantic classes for either a stuff class or a potential object, as illustrated in Fig. 1(d).

However, directly applying standard query-based methods to LPS overlooks the significant distinctions between *things* and *stuff* in outdoor scenes, particularly in large-scale aerial point clouds under remote sensing scenarios, as shown in Fig. 1(a) and (b).

1) *Disparate spatial distributions:* Stuff, i.e., background elements, are typically distributed throughout the scene (e.g., roads and vegetation) and constitute a larger proportion of the point cloud. In contrast, foreground objects are significantly smaller and concentrated in specific local regions.
2) *Different geometric features:* Various stuff classes exhibit distinct geometric attributes (e.g., flat road surfaces versus uneven vegetation points), which can serve as valuable distinguishing features for semantic segmentation. In contrast, instances of the same category share similar geometric properties and lack distinctive textures or colors, complicating instance segmentation.

Due to the distinctions between *things* and *stuff* in point clouds, vanilla query-based methods face significant challenges in large-scale outdoor scenes.

Received 30 September 2024; revised 16 January 2025; accepted 29 March 2025. Date of publication 8 April 2025; date of current version 17 April 2025. This work was supported by the National Natural Science Foundation of China under Grant U21A20484. *(Yu Yang and Jianbiao Mei contributed equally to this work.) (Corresponding authors: Yong Liu; Xiao Xu.)*

Yu Yang, Jianbiao Mei, and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: yu.yang@zju.edu.cn; jianbiaomei@zju.edu.cn; yongliu@iipc.zju.edu.cn).

Siliang Du is with Huawei Technologies Company Ltd., Wuhan 430415, China (e-mail: dusi@whu.edu.cn; xiaoyilin@whu.edu.cn).

Yilin Xiao is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China (e-mail: yilin.xiao@connect.polyu.hk).

Huifeng Wu is with the Institute of Intelligent and Software Technology, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: whf@hdu.edu.cn).

Xiao Xu is with the Institute of Industrial Technology Research, Zhejiang University, Hangzhou 310027, China (e-mail: xuxiao0224@126.com).

Digital Object Identifier 10.1109/TGRS.2025.3558951

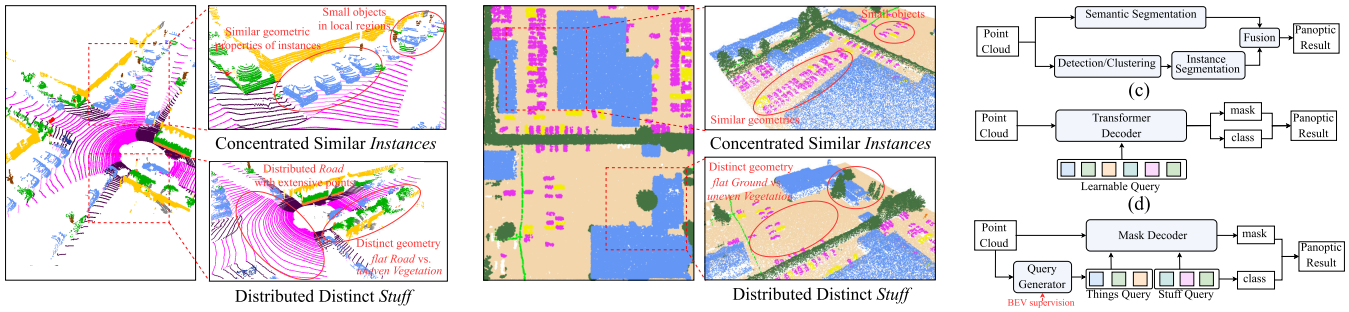

Fig. 1. (a) and (b) Distinction between *things* and *stuff* in vehicular and aerial LiDAR scenes: instances with similar geometries are typically concentrated in local regions, whereas distributed stuff exhibits distinct geometries. (c) Existing semantic/instance separation paradigm. (d) Existing learnable query-based methods ignore these distinctions. (e) We propose a decoupled-query workflow to mitigate competition between classification and segmentation.

1) *Mutual competition between things and stuff using unified queries:* Standard query-based methods utilize unified queries to segment both *things* and *stuff* simultaneously. This approach often prioritizes larger areas (i.e., stuff) for high recall, making it challenging to segment multiple small instances.
2) *Unbalanced proportion for mask supervision: Stuff* classes typically constitute a large proportion of the point cloud, while *things* consist of a limited number of points. This extreme imbalance between positive and negative samples creates challenges for binary mask supervision in query-based methods.
3) *Ambiguity between classification and segmentation:* Vanilla query-based methods employ learnable queries to simultaneously predict semantic classes and binary masks, leading to ambiguity among instances. Classification supervision causes query features to become more similar among different objects within the same category, complicating the distinction between distinct instances.

Based on these observations, we propose decoupling queries (DQs) into *things* and *stuff* queries according to their individual properties. As illustrated in Fig. 1(e), we design a query generator that produces two types of queries and their corresponding semantics by localizing and classifying objects in bird's eye view (BEV). Our key insights are: 1) localizing foreground objects in BEV allows for efficiently generating distinct queries for each instance; 2) aggregating background features in BEV maintain a large receptive field while incurring minimal computational overhead; and 3) classifying objects in BEV provides semantic labels for queries that can be used solely for segmentation decoding.

Specifically, we propose a novel framework termed DQFormer, which adopts a decoupled-query paradigm for unified LPS. DQFormer consists of a multiscale feature encoder, a decoupled query generator, and a query-oriented mask decoder. The feature encoder extracts voxel-wise features and point-wise embeddings, while multiresolution BEV features are generated through the voxel-to-BEV (V2B) operation. The query generator localizes and classifies objects in BEV, extracting multilevel BEV features from their corresponding positions and integrating them into informative queries. Once object-featured queries are obtained, a query-oriented mask decoder predicts the segmentation masks using a masked cross-attention mechanism guided by the queries. These masks are combined with the semantic classes associated with the queries to generate panoptic results.

We evaluate our method on large-scale outdoor scenes, including vehicular datasets nuScenes [13] and SemanticKITTI [14], as well as the aerial dataset DALES [15], which shows that DQFormer outperforms previous superior methods by $+1.8\%$, $+0.9\%$, and $+3.5\%$ in panoptic quality (PQ), respectively, demonstrating the effectiveness of our method for autonomous driving and remote sensing. Our main contributions are as follows.

1) We propose a framework called DQFormer that introduces a novel decoupled-query paradigm to reduce mutual competition for unified LPS.
2) We design a multiscale query generator that generates semantic-aware queries by localizing thing/stuff positions and fusing multilevel BEV embeddings.
3) We propose a query-oriented mask decoder that uses informative queries to guide the segmentation process via a masked cross-attention mechanism.

II. Related Work

A. LiDAR Panoptic Segmentation

Most existing LPS methods can be classified into four types of frameworks: detection-based, clustering-based, center-based, and query-based. The first three are semantic/instance separation paradigms, while the last represents a unified paradigm.

1) Detection-Based Methods: [16], [17], and [18] directly assign a unique ID to the points classified as the foreground *thing* classes within a 3-D bounding box to generate instance masks, whereas other methods [2], [19] propose using the point-box index and bounding box feature to refine the segmentation result further. While these methods predict instance positions and sizes, the semantic branch remains essential for point extraction.

2) Clustering-Based Methods: [1], [3], [4], [10], [20], [21], [22], [23], [24], [25], [26], [27], [28], and [29] use heuristic clustering algorithms to assign instance IDs. These methods mainly focus on enhancing clustering by improving the accuracy of center regression or clustering embeddings.

However, they typically treat semantic and instance segmentation separately, while our DQFormer provides a unified approach for predicting both object and stuff classes.

3) Center-Based Method: [5] utilizes object centers as queries to segment instances, eliminating detection or clustering processes. However, semantic segmentation remains indispensable for retrieving object centers in this method, while DQFormer directly proposes things queries from the BEV space.

4) Query-Based Methods: [11], [12], and [30] use learnable queries for unified LPS, predicting nonoverlapping binary masks and semantic classes for both stuff and potential objects. MaskRange [30] and MaskPLS [11] focus on range-based and point-based segmentation, while PUPS [12] employs point-level classifiers for semantic masks and instance groups. P3Former [31] introduces a mixed-parameterized positional embedding for iterative mask prediction and query updates. However, these methods overlook the distinction between things and stuff in 3-D scenes, leading to mutual competition. In contrast, our DQFormer decouples things and stuff queries based on their characteristics, allowing separate decoding to alleviate competition.

B. Query-Based 2-D/3-D Segmentation

Following the success of DETR [32], [33] in 2-D detection, query-based segmentation methods [34], [35], [36] have emerged to enhance segmentation accuracy and efficiency, such as Panoptic-FCN [6], K-Net [9], MaskFormer [7], Mask2Former [8], and Panoptic SegFormer [10]. These methods use queries to guide segmentation and incorporate techniques like kernel updates and masked attention.

Based on these 2-D advancements, some methods adapt the query-based paradigm for 3-D segmentation [37], [38], [39], [40]. DyCo3D [41], [42] and DKNet [43] use 1-D kernels for 3-D instance mask decoding, while CenterLPS [5] proposes instance queries based on object centers. Mask4D [44] and Mask4Former [45] extend 3-D panoptic segmentation to 4-D by reusing queries from previous scans. Our DQFormer preserves the intrinsic properties of point clouds within a unified query-based framework.

C. Large-Scale Scene Segmentation and Remote Sensing

Large-scale scene segmentation is challenging due to the complexity and diversity of real-world 3-D scenes and their fundamental role in remote sensing. 3-D scene segmentation tasks mainly consist of semantic segmentation, instance segmentation, and panoptic segmentation. The semantic segmentation methods [46], [47], [48] primarily utilize airborne LiDAR point clouds to predict point-wise labels for both foregrounds (e.g., car and building) and background items (e.g., road and vegetation), but do not distinguish between various instances. Some methods [49] explore multimodal semantic segmentation using point clouds and images. Instance segmentation methods [50] solely provide object-centric segmentation results, such as distinguishing individual buildings. In this work, we focus on the LPS task, which not only predicts point-wise semantic labels for both *things* and *stuff*, but also distinguishes different foreground objects, providing a more comprehensive understanding of scenes for remote sensing.

III. Method

A. Overview

As illustrated in Fig. 2, DQFormer consists of three key modules: a multiscale feature encoder, a decoupled query generator, and a query-oriented mask decoder. Specifically,

1) The feature encoder (Section III-B) extracts voxel-wise features and point-wise embeddings at multiple resolutions.
2) The query generator (Section III-C) is designed to produce informative thing/stuff queries that are assigned semantics based on their positions and embeddings in BEV space.
3) The mask decoder (Section III-D) decodes segmentation masks by performing masked cross-attention between queries and point embeddings. Finally, the decoded masks are combined with the semantics of queries to produce the panoptic results.

In this section, we elaborate on the above components as well as the training scheme.

B. Multiscale Feature Encoder

We introduce a sparse backbone to encode input point clouds, extracting multiscale voxel-wise and point-wise features.

Specifically, given an input point cloud $P \in \mathbb{R}^{N_p \times 4}$ (coordinates and intensity), we perform 3-D grid voxelization to obtain the voxel-point indices. Then, voxel features are extracted by feeding point representations f_p (which combine coordinates, intensity, and offsets to the voxel center) within the same voxel into MLPs and applying max-pooling. Consequently, we obtain sparse voxel features $F^v \in \mathbb{R}^{N_v \times 32}$ with a dense spatial resolution of $H \times W \times D$, where N_v is the number of sparse voxels, H, W, and D represent the length, width, and height of the voxelized space, respectively.

Furthermore, we use a UNet-like architecture to extract multiresolution voxel features. Each resolution level$_i$ includes an encoder to aggregate long-range information using radial window self-attention [51], a down-sampling module for sub-voxel features extraction, and a decoder to up-sample and integrate voxel features at resolution level$_i$. In practice, we implement a four-layer feature extractor to encode multiscale voxel features $(F_1^v, F_2^v, F_3^v, F_4^v)$. These voxel features are interpolated to the original point cloud using a k-nearest-neighbor weighted summation, denoted as voxel-to-point (V2P) operation. This operation produces multiscale point-wise embeddings $(F_1^p, F_2^p, F_3^p, F_4^p)$ that capture multiscale contextual and geometric information.

C. Decoupled Query Generator

Due to the sparsity of point clouds, generating informative queries for decoding corresponding segmentation masks is crucial. In this work, we generate query proposals that encapsulate the features of instances/stuff based on their positions and embeddings in the BEV space.

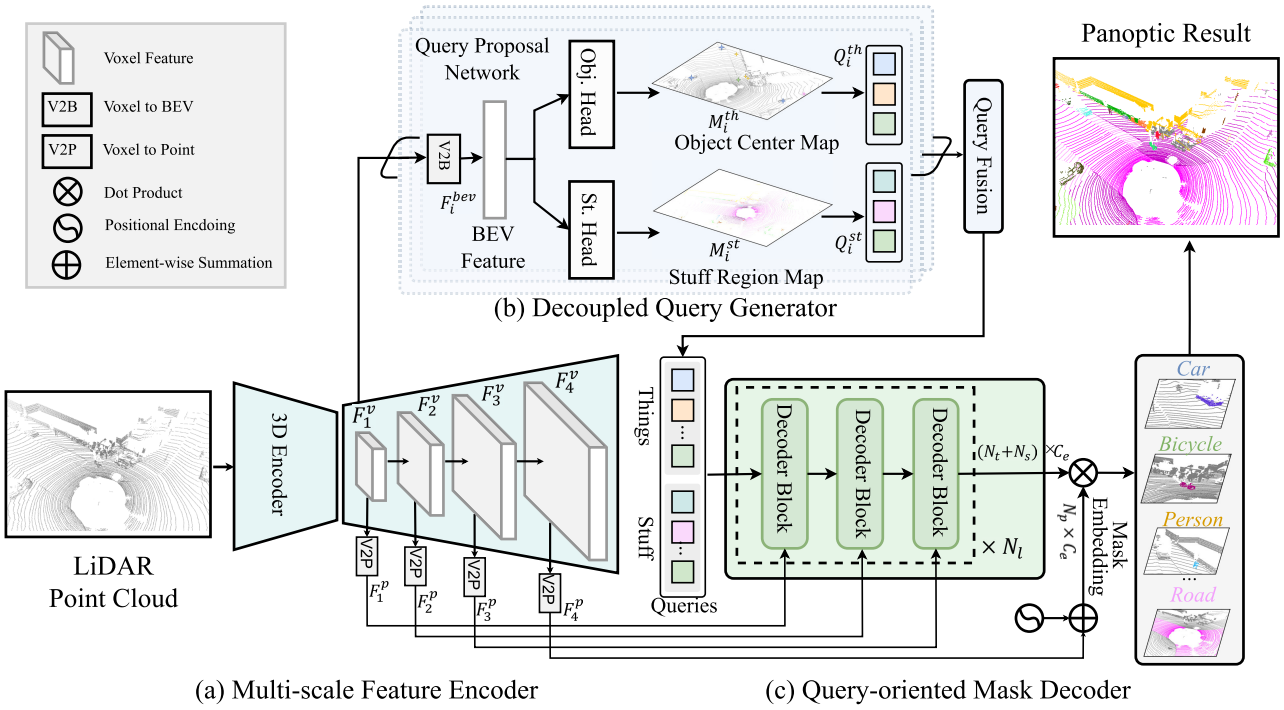

Fig. 2. Overview of DQFormer. (a) Feature encoder is applied to extract voxel features and point embeddings at multiresolutions. (b) Query generator is designed to produce informative things/stuff queries assigned with semantics according to their positions and embeddings in BEV space. (c) Mask decoder performs masked cross-attention between queries and multilevel point embeddings to decode segmentation masks. Finally, the decoded masks are combined with the semantics of the queries to produce the panoptic result. Details of the decoder block are illustrated in Fig. 4.

Decoupled Query Proposal: We propose a query proposal network that generates things/stuff queries from BEV embeddings of different resolutions to explicitly localize instances while enlarging the receptive field for background elements.

1) BEV Embedding Extraction: We project voxel features along the z-axis to generate BEV features through the V2B operation. For a voxel feature F_i^v at level$_i$ with spatial resolution $H_i \times W_i \times D_i$, we concatenate the height dimension D_i with the feature dimension C_i and use stacks of 2-D CNNs with channel-wise and spatial attention to encode the BEV embedding $F_i^{\text{bev}} \in \mathbb{R}^{C_e \times H_i \times W_i}$, where C_e represents the feature dimension in the embedding space. This BEV embedding serves as the shared feature map for locating and classifying objects.

2) BEV Heatmap Prediction: Following [6], we use *object centers* to indicate the positions of potential instances and *stuff regions* for background elements. As illustrated in Fig. 3, we introduce an object center head, consisting of 2-D convolutions, to predict the object center heatmap $M_i^{\text{th}} \in \mathbb{R}^{N_{\text{th}} \times H_i \times W_i}$ at level$_i$, where N_{th} is the number of foreground object categories. Each channel represents potential centers for one class, and different channels denote different semantic classes. Additionally, a stuff region head that uses shallow 2-D transformer decoder layers [10] predicts the stuff region map $M_i^{\text{st}} \in \mathbb{R}^{N_{\text{st}} \times H_i \times W_i}$, with N_{st} denoting the number of stuff categories. Each channel represents the regions of a stuff class from the BEV perspective. The M^{th} and M^{st} serve as heatmaps for localizing and classifying foreground objects and background elements, providing priors of locations and semantic categories for query generation.

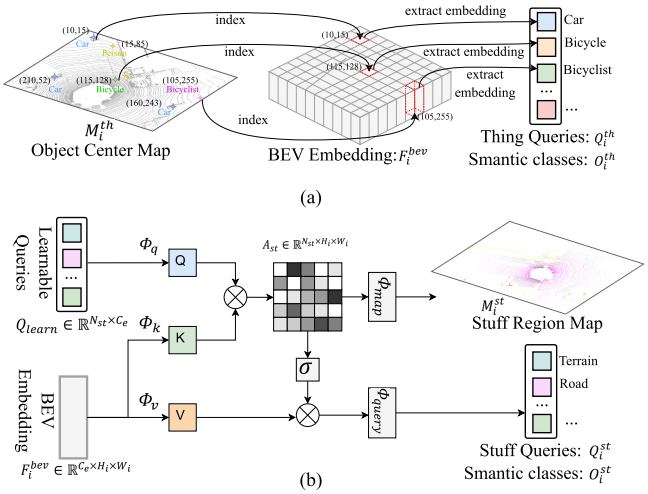

Fig. 3. Details of query proposal generation. Things queries are extracted from the BEV embedding at the corresponding positions. Stuff queries are generated using the learnable-query approach within the BEV space. (a) Things query generation. (b) Stuff query generation.

3) Query Proposals Generation: Next, we generate informative things/stuff queries based on heatmaps M_i^{th} and M_i^{st}, along with the BEV embedding F_i^{bev}.

For things queries, we apply the argmax function along the N_{th} dimension of M^{th} to obtain the semantic category and corresponding confidence score for each position on the BEV, as illustrated in Fig. 3(a). A position with a high score indicates a potential instance location; therefore, we extract embeddings

from F_i^{bev} at the positions with top-N_q scores to represent the query weights and assign the corresponding semantic classes to represent the query categories. For example, assuming a candidate position (x_c, y_c) in the cth channel of M_i^{th}, the embedding $F_i^{\text{bev}}[:, x_c, y_c] \in \mathbb{R}^{C_e \times 1 \times 1}$ serves as the query weight for this instance, with the semantic class assigned as c. This results in things query proposals at level$_i$, denoted as $Q_i^{\text{th}} \in \mathbb{R}^{N_q \times C_e}$ with predicted semantic categories O_i^{th}, where N_q is the number of query proposals.

For stuff queries, since stuff points are widely distributed, it is crucial to incorporate global context. As depicted in Fig. 3(b), we initialize class-fixed learnable queries $Q^{\text{learn}} \in \mathbb{R}^{N_{\text{st}} \times C_e}$, where N_{st} is the number of stuff categories. These queries perform cross-attention with BEV features to predict stuff region maps $M^{\text{st}} \in \mathbb{R}^{N_{\text{st}} \times H \times W}$, where each channel represents the regions of a stuff class from the BEV perspective. This establishes correspondences between each stuff query and BEV positions, which are used to extract and fuse the relative BEV embeddings to update the query weight. This process is formulated as follows:

$$A^{\text{st}} = \frac{\phi_q(Q^{\text{learn}}) \cdot \phi_k\left(F_i^{\text{bev}}\right)}{\sqrt{C_e}} \tag{1}$$

$$M_i^{\text{st}} = \phi_{\text{map}}(A^{\text{st}}), \quad Q_i^{\text{st}} = \phi_{\text{query}}\left[\sigma(A^{\text{st}}) \cdot \phi_v\left(F_i^{\text{bev}}\right)\right]. \tag{2}$$

Here, ϕ_* represents linear layers, σ denotes the softmax function, and $A^{\text{st}} \in \mathbb{R}^{N_{\text{st}} \times H_i W_i}$ are the attention maps. The stuff query proposals at level$_i$ are denoted as $Q_i^{\text{st}} \in \mathbb{R}^{N_{\text{st}} \times C_e}$, associated with their semantic categories O_i^{st}. This formulation effectively enables stuff queries to capture more scene information while maintaining limited computational overhead in BEV.

Decoupled Query Fusion: With the queries generated from various BEV resolutions, we further design a query fusion module to merge multiscale query proposals effectively.

1) *Things query proposals fusion:* We merge object queries at similar positions from multiscale BEV embeddings to enhance individual instance representations. To maintain intra-semantic consistency, we only fuse queries that share the same semantics. Specifically, we employ average pooling to fuse queries whose positions are within the same small window in the BEV and whose cosine similarities between their embeddings exceed a given threshold θ_{th}. This approach ensures that the window constrains the *geometric-consistency* while cosine similarity maintains *instance-awareness* in the embedding space. This results in fusing the multiscale query proposals into an integral set of things queries $Q^{\text{th}} \in \mathbb{R}^{N_t \times C_e}$ with predicted semantic categories $O^{\text{th}} \in \mathbb{R}^{N_t}$, where N_t represents the predicted number of objects.
2) *Stuff query proposals fusion:* We merge queries with the same semantics to integrate multiscale global context for each stuff class. We first identify the presence of each stuff class based on the stuff region maps, where response scores on M^{st} of a class exceeding a threshold θ_{st} indicate the existence of corresponding background elements. We then fuse existing stuff queries with the same semantic categories using average summation, enhancing each query with global receptive fields while maintaining semantic consistency. This yields the set of stuff queries $Q^{\text{st}} \in \mathbb{R}^{N_s \times C_e}$ and their semantic categories $O^{\text{st}} \in \mathbb{R}^{N_s}$, where N_s represents the number of existing stuff classes.

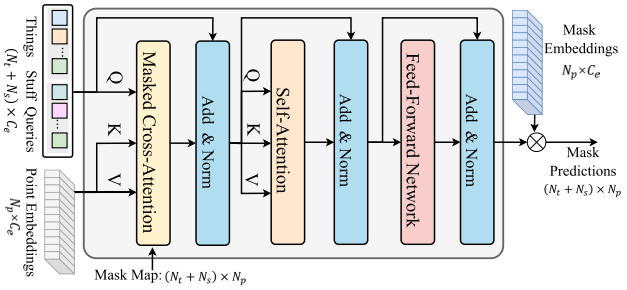

Fig. 4. Detailed pipeline of the decoder block consisting of masked cross-attention, self-attention, and an FFN.

D. Query-Oriented Mask Decoder

Given decoupled queries for objects and backgrounds that encapsulate informative features, we introduce a query-oriented mask decoder to predict segmentation masks through multilevel masked cross-attention.

As depicted in Fig. 4, our mask decoder comprises multiple blocks, each consisting of masked cross-attention at a specific resolution, followed by self-attention and a feed-forward network (FFN). Specifically, in the decoder block at level$_i$, we perform masked cross-attention between the concatenated queries $Q \in \mathbb{R}^{(N_t+N_s) \times C_e}$ and point embeddings $F_i^p \in \mathbb{R}^{N_p \times C_e}$. The mask map $M_i \in \mathbb{R}^{(N_t+N_s) \times N_p}$, indicating the noteworthy key points, is generated from the previous block. A self-attention layer is utilized to establish context between queries, and the FFN is employed to enhance the query representations. Finally, the segmentation mask is generated via the dot product between the output queries and point-wise mask embeddings $E \in \mathbb{R}^{N_p \times C_e}$, along with the sigmoid activation function. The mask decoding process is expressed as follows:

$$Q' = \sigma\left[\frac{\phi_q(Q) \cdot \phi_k\left(F_i^p\right)^T}{\sqrt{C_e}} \odot M_{i-1}\right] \cdot \phi_v\left(F_i^p\right) + Q \tag{3}$$

$$Q'' = \text{FFN}\left[\sigma\left(\frac{\varphi_q(Q') \cdot \varphi_k(Q')^T}{\sqrt{C_e}}\right) \cdot \varphi_v(Q') + Q'\right] \tag{4}$$

$$M_i = \text{Sigmoid}(Q'' \cdot E^T) \tag{5}$$

where ϕ_* and φ_* represent linear layers, $\odot$ denotes the Hadamard product, and σ represents the softmax function. For simplicity, we omit LayerNorm in the formula. It is worth noting that the mask embedding E is composed of the summation of full-resolution point features F_4^p and point-wise positional encoding [52] $P_e \in \mathbb{R}^{N_p \times C_e}$, defined as $E = F_4^p + P_e$.

Deep supervision is applied to the multilevel mask predictions $\{M_1, M_2, \ldots, M_L\}$ during training. In the inference phase, we utilize the masks from the last decoder block and apply the mask fusion module [5] to integrate duplicate masks. These binary masks are combined with the query semantics $\{O^{\text{th}}, O^{\text{st}}\}$ to generate the panoptic results.

E. Loss Function

In training, we supervise the *object center* and *stuff region* heatmaps for the localization and classification of objects and background regions

$$\mathcal{L}_{\text{hm}} = \sum_i \text{FL}\left(M_i^{\text{th}}, Y_i^{\text{th}}\right)/N_q + \sum_i \text{FL}\left(M_i^{\text{st}}, Y_i^{\text{st}}\right)/H_i W_i \tag{6}$$

where $\text{FL}(\cdot,\cdot)$ denotes the focal loss [53], Y_i^{th} and Y_i^{st} represent the ground truth for M_i^{th} and M_i^{st}, respectively. Following [6] and [54], we assign the center of an instance with semantic category c to the cth channel of Y_i^{th} using a Gaussian kernel. Y_i^{st} is generated by interpolating the one-hot semantic label in the BEV space to the corresponding sizes.

Meanwhile, we also supervise the mask predictions for segmentation using binary cross-entropy and dice loss

$$\mathcal{L}_{\text{mask}} = \sum_i \text{BCE}(M_i, Y) + \sum_i \text{Dice}(M_i, Y) \tag{7}$$

where Y represents the ground truth masks matched with predictions. Specifically, instance masks are matched with ground truth through the BEV positions, while stuff masks are matched in a one-to-one manner.

To enhance the point-wise embeddings, we also add an auxiliary MLP head to F_4^p and employ a semantic loss $\mathcal{L}_{\text{sem}}$ to guide the class distribution of points. Overall, DQFormer can be trained end-to-end with the above loss

$$\mathcal{L} = \lambda_{\text{hm}} \cdot \mathcal{L}_{\text{hm}} + \lambda_{\text{mask}} \cdot \mathcal{L}_{\text{mask}} + \lambda_{\text{sem}} \cdot \mathcal{L}_{\text{sem}} \tag{8}$$

where λ_{hm}, λ_{mask}, and λ_{sem} are factors to balance various loss items, and they are set to 1, 5, and 2, respectively.

IV. Experiments

We first present the datasets, including vehicular and aerial LiDAR datasets, along with the evaluation metrics (Section IV-A). Next, we provide implementation details (Section IV-B), followed by our main results and analysis (Section IV-C), qualitative results and discussions (Section IV-D), ablation studies (Section IV-E), and detailed benchmark results (Section IV-F).

A. Datasets and Metrics

nuScenes [13] dataset is a comprehensive urban driving dataset comprising 1000 LiDAR scenes, each spanning a duration of 20 s, captured using a 32-beam LiDAR sensor. It includes 850 scenes for training and validation, with 150 scenes for testing. The LPS task features 16 annotated point-wise labels, comprising ten *thing* categories and six *stuff* categories.

SemanticKITTI [14] is derived from the KITTI [75] odometry dataset, featuring 22 LiDAR sequences captured with a Velodyne HDL-64 laser scanner. It allocates ten sequences for training, one for validation, and 11 for testing. The dataset includes 19 annotated point-wise labels for LPS, comprising eight *thing* classes and 11 *stuff* classes.

DALES [15] is a large-scale aerial LiDAR dataset with over 500 million points spanning an area of 10 km^2. DALES is one of the newest large-scale aerial laser scanner (ALS) benchmarks, significantly larger than traditional ALS benchmarks, such as ISPRS [76]. The dataset consists of 40 tiles, each covering about 0.5 km^2, which are randomly split into 29 training tiles and 11 testing tiles. The *thing* classes are *buildings*, *cars*, *trucks*, *power lines*, *fences*, and *poles*, while *ground* and *vegetation* are classified as *stuff* classes.

Metrics: The metrics [77] for LiDAR-based panoptic segmentation include PQ, segmentation quality (SQ), and recognition quality (RQ), which are formulated as

$$\text{PQ} = \underbrace{\frac{\sum_{\mathbf{TP}} \mathbf{IoU}}{|\mathbf{TP}|}}_{\text{SQ}} \times \underbrace{\frac{|\mathbf{TP}|}{|\mathbf{TP}| + \frac{1}{2}|\mathbf{TN}| + \frac{1}{2}|\mathbf{FP}|}}_{\text{RQ}}. \tag{9}$$

These metrics are also calculated separately for *thing* and *stuff* classes indicated by PQ^{Th}, SQ^{Th}, RQ^{Th} and PQ^{St}, SQ^{St}, RQ^{St}. In addition, we also report $\text{PQ}^{\dagger}$, as defined in [78], which utilizes SQ as PQ for *stuff* classes.

B. Implementation Details

For the voxelization process, we discretize the 3-D space within $[[\pm 51.2], [\pm 51.2], [-4, 2.4\ \text{m}]]$ into voxels with a resolution of $[0.05, 0.05, 0.05\ \text{m}]$. Multiscale voxel features F_i^v are obtained at resolutions corresponding to $\{(1/8), (1/4), (1/2), 1\}$ of the original dense resolutions. We set the number of instance queries per scan (N_q) to 150. The thresholds $\theta_{\text{th}} = 0.85$ and $\theta_{\text{st}} = 0.5$ are used to discriminate cosine similarities for *things* queries and to indicate the existence of *stuff* regions, respectively. The mask decoder consists of $N_l = 3$ decoder blocks, each employing point embeddings interpolated from voxel features with resolutions of $\{(1/8), (1/4), (1/2)\}$. The models are trained for 80 epochs using the AdamW optimizer [79] on 8 NVIDIA RTX A6000 GPUs, with an initial learning rate of $1e^{-4}$, decayed by a factor of 10 at epoch 60.

C. Comparison With the State of the Art

Results on nuScenes: Tables I and II present the comparison results between our DQFormer and other state-of-the-art methods on the nuScenes [13] test and validation sets.

1) Compared With Detection-Based Methods: DQFormer shows significant improvements over single-architecture methods, outperforming EfficientLPS [2] and AOPNet [17] by +17.2% and +7.4% in PQ^{Th}. This underscores the effectiveness of our query generator in localizing and classifying instances without the need for bounding box predictions. Compared to two-architecture methods (semantic segmentation models + 3-D detection method), DQFormer exceeds them in SQ^{Th} by +2.2% and 0.8%. We assert that our query-oriented mask decoder leverages informative queries to establish affinities with all points, resulting in more precise segmentation masks than those derived solely from bounding box predictions.

2) Compared With Clustering-Based Methods: These methods [1], [3], [4], [20], [22] typically utilize separate semantic and instance branches. In contrast, DQFormer

TABLE I

COMPARISON OF LPS PERFORMANCE ON THE TEST SET OF nuScenes. ALL RESULTS IN [%]

Type	Method	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}
Detection-based	PanopticTrackNet [16]	51.6	56.1	63.3	80.4	45.9	56.1	81.4	61.0	75.4	79.0
	EfficientLPS [2]	62.4	66.0	74.1	83.7	57.2	68.2	83.6	71.1	84.0	83.8
	AOP-Net [17]	68.3	-	78.2	86.9	67.3	75.6	88.6	69.8	82.6	84.0
	SPVNAS [55] + CenterPoint [56]	72.2	76.0	81.2	88.5	71.7	79.4	89.7	73.2	84.2	86.4
	Cylinder3D++ [57] + CenterPoint [56]	76.5	79.4	85.0	89.6	76.8	84.0	91.1	76.0	86.6	87.2
	(AF)2-S3Net [58] + CenterPoint [56]	76.8	80.6	85.4	89.5	79.8	86.8	91.8	71.8	83.0	85.7
Clustering-based	Panoptic-PolarNet [3]	63.6	67.1	75.1	84.3	59.0	69.8	84.3	71.3	83.9	84.2
	PolarStream [59]	70.9	74.4	81.7	85.9	70.3	80.3	86.7	71.7	84.2	84.4
	LCPS(LiDAR) [27]	72.8	76.3	81.7	88.6	72.4	80.0	90.2	73.5	84.6	86.1
	CPSeg [60]	73.2	76.3	**82.7**	88.1	72.9	**81.3**	89.2	74.0	**85.0**	**86.3**
Query-based	MaskPLS [11]	61.1	64.3	68.5	86.8	54.3	58.8	87.8	72.4	63.4	85.1
	DQFormer (Ours)	**73.9**	**76.8**	82.4	**89.6**	**74.4**	80.7	**91.9**	**78.2**	**85.0**	85.8

TABLE II

COMPARISON OF LPS PERFORMANCE ON THE VALIDATION SET OF nuScenes. ALL RESULTS IN [%]

Type	Method	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}
Detection-based	PanopticTrackNet [16]	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3
	EfficientLPS [2]	62.0	65.6	73.9	83.4	56.8	68.0	83.2	70.6	83.6	83.8
Clustering-based	DS-Net [1]	42.5	51.0	50.3	83.6	32.5	38.3	83.1	59.2	70.3	84.4
	GP-S3Net [61]	61.0	67.5	72.0	84.1	56.0	65.2	85.3	66.0	78.7	82.9
	Panoptic-PolarNet [3]	63.4	67.2	75.3	83.9	59.2	70.3	84.1	70.4	83.5	83.6
	PVCL [62]	64.9	67.8	77.9	81.6	59.2	72.5	79.7	67.6	79.1	77.3
	SCAN [63]	65.1	68.9	75.3	85.7	60.6	70.2	85.7	72.5	83.8	85.7
	Panoptic-PHNet [4]	74.7	77.7	84.2	**88.2**	74.0	82.5	89.0	75.9	86.9	86.8
Query-based	MaskPLS-M [11]	57.7	60.2	66.0	71.8	64.4	73.3	84.8	52.2	60.7	62.4
	SAL [64]	70.5	–	80.8	85.9	79.4	–	–	61.7	–	–
	PUPS [12]	74.7	77.3	83.3	89.4	75.4	81.9	91.8	73.6	85.6	85.6
	P3Former [31]	75.9	78.9	84.7	89.7	76.8	83.3	**92.0**	75.4	87.1	86.0
	DQFormer (Ours)	**77.7**	**79.5**	**89.2**	86.8	**77.8**	**89.5**	86.7	**77.5**	**88.6**	**87.0**

achieves superior results through a unified query-based segmentation approach, demonstrating a substantial gain of 3.8% in PQ^{Th}. We note that these methods often rely on heuristic clustering algorithms to group instance points, which can lead to incomplete masks, particularly for large objects with scattered points. In contrast, DQFormer establishes affinities between queries and point features in the embedding space, remaining unaffected by the geometric locations of points.

3) Compared With Query-Based Methods: Compared to P3Former, which uses a mixed-parameterized positional embedding to distinguish various instances, DQFormer improves performance for both things and stuff, achieving gains of +1.0% on PQ^{Th} and +2.1% on PQ^{St}. This demonstrates that the query generator effectively extracts practical queries for both things and stuff through two decoupled branches, validating the effectiveness of our query decoupling strategy.

Results on SemanticKITTI: We validate our method's effectiveness and generalization on the SemanticKITTI [14] test and validation sets, as shown in Tables III and IV.

DQFormer outperforms all detection-based and clustering-based methods, achieving 5.7% higher PQ than EfficientLPS [2] and 1.6% higher than Panoptic-PHNet [4]. It also surpasses the recent center-based method CenterLPS [5], which uses object queries for instance segmentation, by 1.5% in PQ. We explain that while the center-based method shares a similar approach with DQFormer by predicting object queries based on BEV position to generate masks, DQFormer further integrates this method for stuff classes and provides more global receptive fields for stuff queries, resulting in more complete masks.

Compared to query-based methods, DQFormer demonstrates notable improvements in PQ over MaskPLS and PUPS, with gains of 4.9% and 0.9% on the test split, respectively. Although it slightly lags behind PUPS in RQ due to PUPS's

TABLE III

COMPARISON OF LPS PERFORMANCE ON THE TEST SET OF SemanticKITTI [14]. ALL RESULTS IN [%]

Type	Method	PQ	PQ†	RQ	SQ
Detection-based	RangeNet++ [65] + PointPillars [66]	37.1	45.9	47.0	75.9
	KPConv [67] + PointPillars [66]	44.5	52.5	54.4	80.0
	EfficientLPS [2]	57.4	63.2	68.7	83.0
Clustering-based	LPSAD [20]	38.0	47.0	48.2	76.5
	Panoster [22]	52.7	59.9	64.1	80.7
	Panoptic-PolarNet [3]	54.1	60.7	65.0	81.4
	DS-Net [1]	55.9	62.5	66.7	82.3
	CPSeg [68]	57.0	63.5	68.8	82.2
	SCAN [63]	61.5	67.5	72.1	84.5
	Panoptic-PHNet [4]	61.5	67.9	72.1	84.8
Center-based	CenterLPS [5]	61.6	67.9	72.6	84.0
Query-based	MaskPLS-M [11]	58.2	63.3	68.6	83.9
	PUPS [12]	62.2	65.8	72.8	84.2
	DQFormer (ours)	63.1	67.9	73.6	85.0
	DQFormer† (ours)	**64.9**	**69.5**	**75.1**	**85.8**

TABLE IV

COMPARISON OF LPS PERFORMANCE ON THE VALIDATION SET OF SemanticKITTI [14]. BOLD AND UNDERLINED INDICATE THE BEST AND SECOND-BEST PERFORMANCES. ALL RESULTS IN [%]

Type	Method	PQ	PQ†	RQ	SQ
Detection-based	RangeNet++ [65] + PointPillars [66]	36.5	–	44.9	73.0
	PanopticTrackNet [16]	40.0	–	48.3	73.0
	KPConv [67] + PointPillars [66]	41.1	–	50.3	74.3
	EfficientLPS [2]	59.2	65.1	69.8	75.0
Clustering-based	LPSAD [20]	36.5	46.1	–	–
	Panoster [22]	55.6	–	66.8	79.9
	DS-Net [1]	57.7	63.4	68.0	77.6
	Panoptic-PolarNet [3]	59.1	64.1	70.2	78.3
Center-based	CenterLPS [5]	62.1	67.0	72.0	80.7
Query-based	SAL [64]	59.5	–	69.2	75.7
	MaskPLS-M [11]	59.8	–	69.0	76.3
	PUPS [12]	**64.4**	**68.6**	**74.1**	81.5
	DQFormer (ours)	63.5	67.2	73.1	**81.7**

TABLE V

COMPARISON OF LIDAR SEGMENTATION PERFORMANCE ON THE TEST SET OF DALES [15]. mIoU_{th}, mIoU_{st}, AND mIoU_{all} REPRESENT mIoU VALUES FOR THINGS, STUFF, AND ALL CATEGORIES, RESPECTIVELY. ALL RESULTS IN [%]

Method	Semantic Segmentation		
	mIoU_{th}	mIoU_{st}	mIoU_{all}
ConvPoint [69]	58.4	94.4	67.4
PointNet++ [70]	60.2	92.7	68.3
SuperCluter [71]	-	-	77.3
SPT [72]	74.5	94.9	79.6
KPConv [67]	76.3	95.6	81.1
PCE [73]	77.1	95.6	81.7
MCTNet [74]	**79.0**	**96.5**	**83.3**
DQFormer (Ours)	78.3	96.2	82.2
Method	Panoptic Segmentation		
	PQ	RQ	SQ
SuperCluster [71]	61.2	68.6	87.1
DQFormer (Ours)	**64.7**	**70.9**	**90.5**

use of a CutMix strategy for data augmentation, DQFormer still enhances SQ by 0.8% and 0.2% on the test and validation sets. This underscores DQFormer's effectiveness in decoupling things and stuff queries to reduce competition between classification and segmentation, a factor overlooked by previous query-based methods, thereby promoting SQ.

Results on DALES: In Table V, we compare the semantic and panoptic segmentation performances of various methods on the DALES test set [15]. For semantic segmentation, we report the mIoU metric for things (mIoU_{th}), stuff (mIoU_{st}), and all categories (mIoU_{all}). Our DQFormer achieves a comparable performance of a mIoU_{all} of 82.2% compared to MCTNet [74], which specifically focuses on the semantic segmentation task.

In panoptic segmentation, DQFormer outperforms the previous SoTA method, SuperCluster [71], achieving a notable improvement with a 3.5% boost in PQ and a 3.4% increase in SQ. This improvement is attributed to the localize-then-segmentation paradigm of DQFormer, which first localizes objects on the BEV and then predicts segmentation masks, resulting in more complete and cohesive results. In contrast, SuperCluster relies on clustering algorithms to group instance points, which are widely scattered and can lead to the problem of over-segmentation in large-scale outdoor scenes.

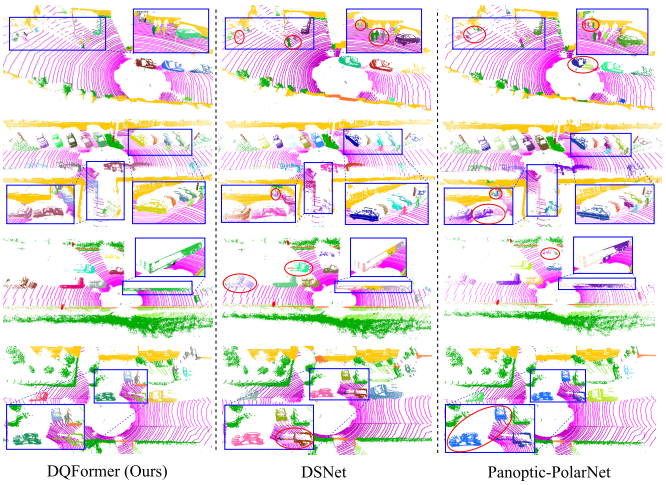

Fig. 5. Qualitative comparisons of panoptic segmentation between DQFormer with DSNet [1] and Panoptic-PolarNet [3], on SemanticKITTI test split.

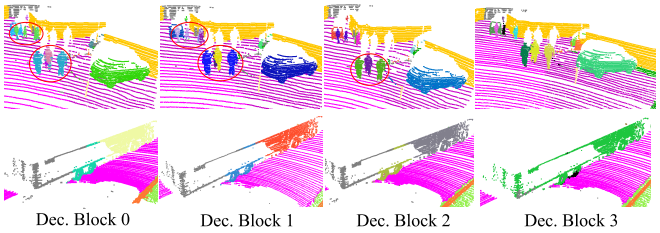

Fig. 6. Qualitative comparisons of mask predictions across different decoder blocks on the SemanticKITTI test split.

D. Qualitative Results and Discussion

This section presents visualization results, including qualitative results in vehicular and aerial scenes, mask predictions across decoder blocks, object center predictions, and attention maps for things and stuff queries.

1) Panoptic Segmentation in Vehicular Scenes: Fig. 5 provides qualitative comparisons of our DQFormer with DSNet and Panoptic-PolarNet using the SemanticKITTI test set. The first two rows highlight DQFormer's superior ability to segment small instances in local regions. In contrast, DSNet and Panoptic-PolarNet struggle with under-segmentation, particularly for adjacent instances with similar geometries. This demonstrates the effectiveness of our query generator in distinguishing individual objects. In the third row, DQFormer efficiently segments large objects like buses and trucks, while the other methods face over-segmentation issues with sparse instances. The last row illustrates DQFormer's accuracy in identifying rare objects, such as trolleys, and its proficiency in distinguishing widely distributed stuff points based on their attributes.

2) Comparisons Across Decoder Blocks: Fig. 6 visualizes mask predictions from different decoder blocks. Shallow blocks struggle with the under-segmentation of adjacent small objects and produce fragmented masks for large targets. In contrast, deeper blocks generate more precise masks for

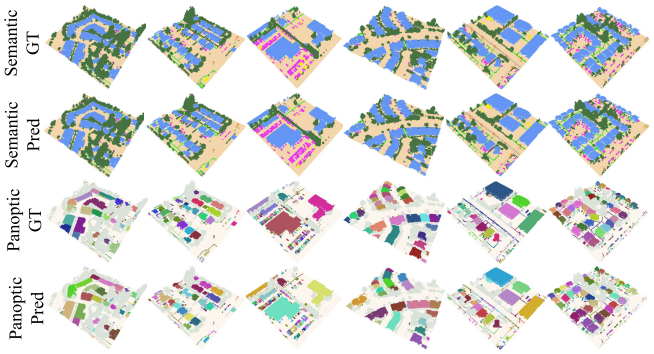

Fig. 7. Qualitative results in aerial LiDAR point cloud, including semantic segmentation and panoptic segmentation on the DALES test split.

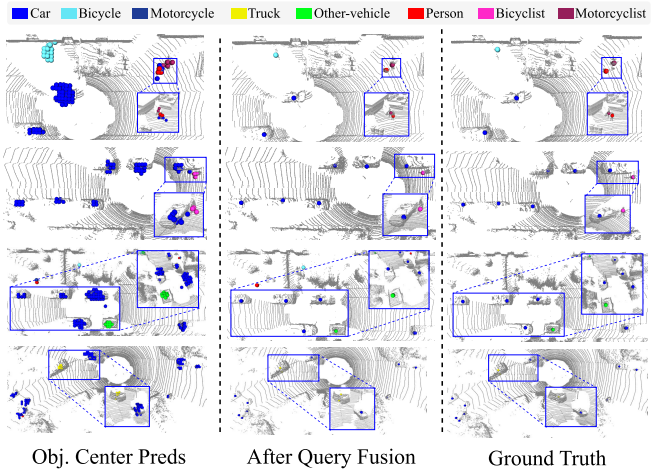

Fig. 8. Visualization of object centers extracted from the BEV heatmap, after the query fusion module, alongside the corresponding ground truth.

both objects and background elements, demonstrating the effectiveness of the masked cross-attention mechanism, which helps queries focus on key points for improved segmentation accuracy.

3) Panoptic Segmentation in Aerial Scenes: Fig. 7 presents qualitative results from the DALES [15] test set, a large-scale aerial scan dataset collected by an airborne LiDAR system. This demonstrates that our method predicts accurate semantic and panoptic results, underscoring the generalizability of DQFormer in large-scale outdoor scenes. Notably, our method distinguishes adjacent instances with similar geometries, such as cars and buildings, demonstrating the effectiveness of DQFormer in aerial remote sensing.

4) Visualization of the Object Centers: Fig. 8 shows predicted object centers from the query fusion module alongside their ground truth, using distinct colors for different categories. Comparing the first two columns reveals the efficiency of our query fusion module in merging duplicated queries, resulting in compact queries that align closely with the ground truth.

The first two rows demonstrate our query generator's ability to localize adjacent small instances and effectively fuse their queries through geometric consistency and feature similarity constraints. The third and fourth rows highlight the generator's capability to distinguish adjacent instances with similar geometric attributes. In the last row, a failure case is noted where

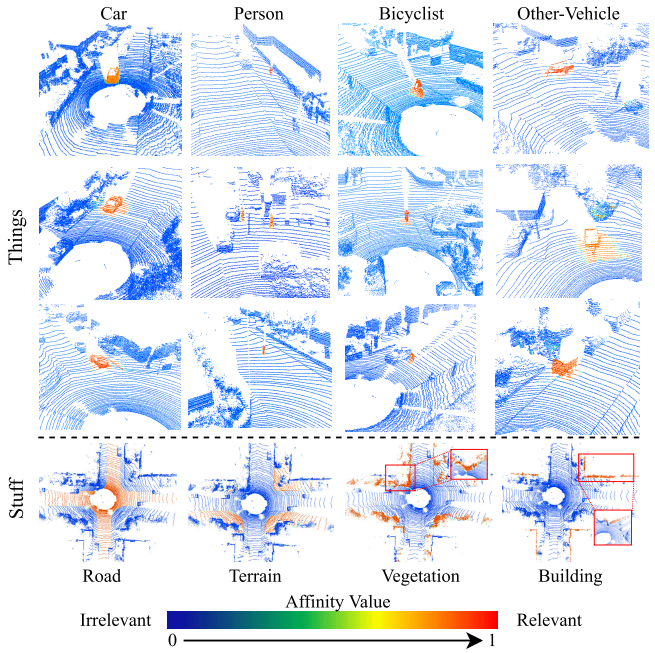

Fig. 9. Visualization of attention maps between queries and points cloud.

TABLE VI
ABLATION ON THE NETWORK COMPONENTS ON SEMANTICKITTI VALIDATION SET. ALL SCORES ARE IN [%]

Mask Decoder	Mask Embed	Query Generator Stuff / Thing / Fusion			Mask Fusion	PQ	RQ	SQ	PQ^{Th}	PQ^{St}
						60.6	70.4	76.0	63.1	58.8
✓						60.8	70.7	76.0	62.3	59.7
✓	✓					61.3	71.4	76.6	63.7	59.6
✓	✓	✓				61.7	71.5	76.4	64.2	**59.9**
✓	✓	✓	✓			62.5	72.0	76.6	66.4	59.6
✓	✓	✓	✓	✓		63.1	72.9	81.5	67.8	59.6
✗	✓	✓	✓	✓		62.1	71.5	76.5	65.8	59.4
✓	✓	✓	✓	✓	✓	**63.5**	**73.1**	**81.7**	**68.8**	59.6

all queries for a truck are not fused into a single embedding, emphasizing the critical role of the mask fusion process.

5) Visualization of Attention Maps: In Fig. 9, we explore the relationship between queries and the point cloud, with red indicating high correlations and blue indicating low correlations. The visualization shows that things queries align with locally concentrated points, while stuff queries focus on points distributed throughout the scene. This highlights the effectiveness of query embeddings in capturing relevant features, facilitating precise segmentation mask generation through the masked-attention mechanism.

E. Ablation Study

1) Effects of Network Components: Table VI presents the ablation results of our proposed components.

1) *Baselines:* We establish a clustering-based baseline (line 1) that integrates the dynamic shift module [1], and two query-based baselines (lines 2 and 3) that utilize a vanilla mask decoder. Results indicate that query-based methods outperform the clustering method (line 1 versus line 2), and incorporating masking embedding further

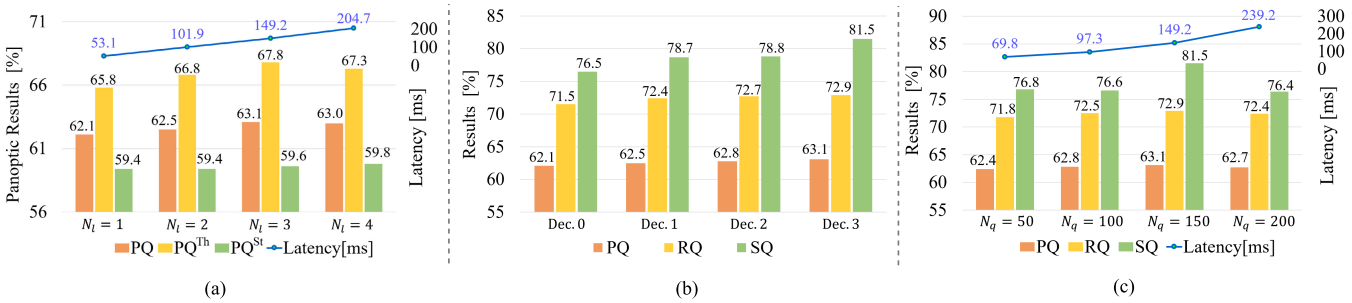

Fig. 10. Ablation on the number of decoder blocks and things queries on the SemanticKITTI validation set, along with the latency of mask decoding. (a) Ablation on the number of decoder blocks. (b) Ablation on the output of different decoder blocks. (c) Ablation on the number of things queries.

TABLE VII

ABLATION ON THE DECOUPLING STRATEGY ON SEMANTICKITTI VALIDATION SET, WITH DQ, HM, AND DC CLS.&SEG. WE DO NOT USE THE MASK FUSION FOR A FAIR COMPARISON. ALL SCORES ARE IN [%]

Variants	DQ	Match HM / BEV	DC cls.&seg. Thing / Stuff		PQ	RQ	SQ	PQ^{Th}	PQ^{St}
Baseline	✗	✓			61.3	71.4	76.6	63.7	59.6
1	✓	✓			61.4	71.5	76.6	64.0	59.6
2	✓	✓		✓	61.7	71.5	76.4	64.2	**59.9**
3	✓		✓	✓	62.4	72.0	76.6	66.5	59.5
4	✓		✓	✓	**63.1**	**72.9**	**81.5**	**67.8**	59.6

TABLE VIII

PER-CLASS PQ RESULTS OF SMALL INSTANCES ON THE SEMANTICKITTI VALIDATION SET. ALL SCORES ARE IN [%]

Variants	PQ^{Th}	Bicycle	Motorcycle	Person	Bicyclist
Baseline	63.7	52.4	59.6	77.1	90.3
Ours	**68.8**$_{\uparrow 5.1}$	**55.5**$_{\uparrow 3.1}$	**67.6**$_{\uparrow 8.0}$	**79.5**$_{\uparrow 2.4}$	**91.5**$_{\uparrow 1.2}$

TABLE IX

ABLATION ON QUERY FUSION USING POSITIONAL INFORMATION AND COSINE SIMILARITY ON THE SEMANTICKITTI VALIDATION SET

Position	Cosine	PQ^{Th}	Bicycle	Motorcycle	Person	Bicyclist
✗	✗	67.3	54.6	65.2	78.8	91.1
✓	✗	68.2	54.9	67.2	79.0	91.4
✗	✓	67.8	55.1	66.6	79.3	91.3
✓	✓	**68.8**$_{\uparrow 1.5}$	**55.5**$_{\uparrow 0.9}$	**67.6**$_{\uparrow 2.4}$	**79.5**$_{\uparrow 0.7}$	**91.5**$_{\uparrow 0.4}$

TABLE X

ABLATION ON THE SIMILARITY THRESHOLDS θ_{th} FOR THINGS QUERIES FUSION ON THE SEMANTICKITTI VALIDATION SET. SEMANTIC-AWARE DENOTES ONLY FUSING QUERIES WITH THE SAME SEMANTICS. ALL SCORES ARE IN [%]

Semantic-aware	θ_{th}	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}
✓	1.00	62.8	72.6	78.5	67.3	73.1	85.4
✓	0.95	63.1	72.9	81.5	67.9	73.7	92.5
✓	0.90	63.2	73.0	81.6	68.2	74.0	92.6
✓	0.85	**63.5**	**73.1**	**81.7**	**68.8**	**74.2**	**92.9**
✓	0.80	63.4	**73.1**	81.6	68.6	**74.2**	92.7
✗	0.85	63.3	72.9	78.8	68.3	73.6	86.1

TABLE XI

ABLATION ON QUERY FUSION AND MASK FUSION MODULES ON THE SEMANTICKITTI VALIDATION SET. ALL SCORES ARE IN [%]

query fusion	mask fusion	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}
✗	✗	62.5	72.0	76.6	66.4	72.0	80.4
✗	✓	62.8	72.6	78.5	67.3	73.1	85.4
✓	✗	63.1	72.9	81.5	67.8	73.7	92.4
✓	✓	**63.5**	**73.1**	**81.7**	**68.8**	**74.2**	**92.9**

TABLE XII

ABLATION ON THE NUMBER OF SAMPLE POINTS IN MASK LOSS ON THE nuScenes VALIDATION SET. ALL SCORES ARE IN [%]

Sub-Sample	S_{th}	S_{all}	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}
✗	-	-	77.2	88.8	86.5	77.1	89.0	86.3
✓	100	20,000	76.7	88.4	86.4	76.4	88.3	86.3
✓	500	20,000	77.4	88.8	**86.8**	77.4	88.9	**86.8**
✓	1,000	20,000	**77.7**	**89.2**	**86.8**	**77.8**	**89.5**	86.7
✓	500	10,000	77.0	88.7	86.4	76.8	88.8	86.2
✓	1,000	10,000	77.3	88.9	86.6	77.4	89.1	86.5

TABLE XIII

ABLATION ON THE MASK LOSS FUNCTIONS ON THE nuScenes VALIDATION SET. BCE DENOTES THE BINARY CROSS-ENTROPY LOSS. ALL SCORES ARE IN [%]

BCE	Focal	Dice	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}
✓	-	✓	**77.7**	**89.2**	**86.8**	**77.8**	**89.5**	**86.7**
✓	✓	-	76.4	87.8	86.5	76.0	87.6	86.4
-	✓	✓	77.1	88.7	86.5	77.0	89.0	86.3
✓	✓	✓	77.4	89.0	86.6	77.5	89.3	86.5

enhances performance (line 2 versus line 3) by integrating positional information.

2) *Effects of the Query Generator:* Lines 4 and 5 introduce the query generator for *stuff* and *things*, both achieving PQ gains. Specifically, the query generator for *stuff* enhances PQ^{St} by 0.3% (line 3 versus line 4), while the generator for *things* boosts PQ^{Th} by 2.7% (line 3 versus. line 5), demonstrating its effectiveness in generating practical query proposals. Additionally, including the query fusion module (line 6) improves PQ, highlighting the advantages of multiscale query fusion.

3) *Effects of the Mask Decoder:* Line 7 replaces the query-oriented mask decoder with a single-layer transformer decoder, leading to a 5% decrease in SQ compared to line 6. This underscores the importance of the multilevel masked cross-attention mechanism.

4) *Effects of the Mask Fusion:* Line 8 employs the mask fusion module, yielding a 1.0% improvement in PQ^{Th}

TABLE XIV

ABLATION ON MODEL SETTINGS AND EFFICIENCY BETWEEN DQFORMER AND EXISTING METHODS. HERE, N_L REPRESENTS THE NUMBER OF DECODER BLOCKS, N_q DENOTES THE INSTANCE QUERY NUMBER, AND C_e INDICATES THE FEATURE DIMENSION. ALL EXPERIMENTS ARE CONDUCTED ON THE nuScenes VALIDATION SPLIT. † INDICATES THAT WE MEASURE THE LATENCY ON OUR HARDWARE USING THE OFFICIALLY RELEASED CODES

Type	Method	N_L	N_q	C_e	PQ	RQ	SQ	PQ^{Th}	PQ^{St}	Params	FPS
Detection-based	EfficientLPS [2]	-	-	-	62.0	73.9	83.4	56.8	70.6	43.8M	4.0†
Clustering-based	DS-Net [1]	-	-	-	42.5	50.3	83.6	32.5	59.2	56.5M	2.2†
Query-based	MaskPLS-M [11]	-	-	-	57.7	66.0	71.8	64.4	52.2	31.5M	4.7†
Query-based	DQFormer (Ours)	1	150	128	76.2	88.1	86.1	75.7	77.1	41.9M	**5.3**
		2	150	256	76.9	88.9	86.2	76.6	**77.6**	47.3M	4.6
		3	150	256	**77.7**	**89.2**	**86.8**	**77.8**	77.5	50.5M	4.5
		3	200	256	77.0	88.8	86.3	76.6	77.5	50.5M	4.4

over line 6 by merging duplicated instance masks, which enhances SQ.

2) Effects of Decoupling Strategy: Table VII validates the effectiveness of our decoupling strategy, which includes decoupling *things* and *stuff* queries, as well as disentangling classification and segmentation.

1) *Effects of DQs:* The baseline employs coupled queries with Hungarian matching (HM), achieving a PQ of 61.3%. Variant 1, using independent learnable queries, shows only marginal improvements due to limited initial query information, making it less effective in outdoor scenarios. In Variant 4, decoupled queries based on BEV positions and embeddings achieve significant gains of 1.8% in PQ and 4.1% in PQ^{Th}. This improvement results from better alignment of queries with their properties, enabling the model to focus on specific areas and extract relevant features.
2) *Effects of decoupling classification/segmentation (DC Cls.&Seg.):* Variants 2 and 3 assign semantic classes to queries based on BEV prediction, resulting in increases of 0.3% and 1.0% in PQ over Variant 1. This approach reduces similarities between different objects of the same class, enhancing instance distinction.

3) Effects on Segmenting Small Instances: Table VIII compares the PQ of small instances between DQFormer and the baseline model, showing significant improvements. This confirms that our decoupling strategy effectively reduces mutual competition between objects and large background elements, enhancing the segmentation of smaller instances.

4) Effects of the Decoder Blocks and Things Queries: Fig. 10 compares mask decoder block settings and object queries, highlighting latency on an RTX A6000 GPU. In Fig. 10(a), more decoder blocks enhance PQ and PQ^{Th} due to the masked cross-attention mechanism and deep supervision. Fig. 10(b) indicates that deeper blocks yield better performance, showing gradual convergence to keypoints. Fig. 10(c) assesses object queries N_q, with optimal performance at $N_q = 150$; too few queries miss difficult objects, while too many cause over-segmentation and higher costs.

5) Effects of Fusion Constraints: Table IX examines the impact of query fusion constraints based on positional information and cosine similarities. Both constraints enhance performance, with their combination achieving optimal results. The positional constraint fuses geometrically close queries, while the cosine similarity constraint integrates characteristically similar ones. This combination ensures both *geometric-consistency* and *instance-awareness*. Fig. 11 shows that the cosine similarity constraint effectively differentiates adjacent individuals by highlighting gaps in their feature similarity.

Fig. 11. Comparison of query fusion with and without cosine similarity. The red balls represent person queries, while the blue balls indicate car queries. (a) w/o cosine similarity. (b) w/ cosine similarity.

6) Effects of Query Fusion and Mask Fusion: Table X shows that the optimal similarity threshold for fusing things queries is $\theta_{th} = 0.85$ with semantic-aware fusion, while semantic-agnostic fusion results in a 6.8% decrease in SQ^{Th} due to the risk of merging instances of different semantics. Additionally, Table XI compares our query fusion module with the mask fusion module from [5]. These modules integrate queries and merge duplicated instance masks, respectively, with results indicating that combining both techniques achieves optimal performance.

7) Effects of Mask Loss: Instances are usually concentrated in local regions, resulting in an imbalance between positive and negative samples for mask supervision. To mitigate this, we randomly sample S_{th} points from objects and S_{all} points from the scene for mask loss calculation to ensure balanced supervision. Ablation studies in Table XII demonstrate that this sub-sampling strategy significantly enhances SQ (SQ^{Th}) and PQ (PQ^{Th}). Additionally, Table XIII shows that combining binary cross-entropy and dice loss yields the best performance for mask supervision.

8) Effects of Model Settings and Efficiency: Table XIV evaluates the performance and efficiency of our DQFormer against three types of methods: detection-based, clustering-based, and query-based methods, all tested on an NVIDIA RTX A6000 GPU using the nuScenes validation set. The results show that DQFormer significantly outperforms previous methods

TABLE XV

DETAILED PER-CLASS RESULTS ON THE SEMANTICKITTI VALIDATION SET, WHICH INCLUDE BOTH SEMANTIC SEGMENTATION AND PANOPTIC SEGMENTATION. BLUE NUMBERS INDICATE THE BEST RESULTS, WHILE BOLD NUMBERS INDICATE THE SECOND-BEST RESULTS. ALL RESULTS IN [%]

Method	Metrics	Car	Bicycle	Motorcycle	Truck	Other Vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic Sign	Mean
PolarNet [80]	IoU	91.5	30.7	38.8	46.4	24.0	54.1	62.2	0.0	92.4	47.1	78.0	**1.8**	89.1	45.5	85.4	59.6	72.3	58.1	42.2	53.6
SalsaNext [81]	IoU	90.5	44.6	49.6	86.3	54.6	74.0	81.4	0.0	93.4	40.6	69.1	0.0	84.6	53.0	83.6	64.3	64.2	54.4	39.8	59.4
RangeViT [82]	IoU	94.7	44.1	61.4	71.8	37.7	65.3	75.5	0.0	**95.5**	**48.8**	83.1	0.0	88.3	60.0	86.3	65.3	72.7	63.1	42.7	60.9
SPVNAS [55]	IoU	96.5	44.8	63.1	59.9	**64.3**	72.0	86.0	0.0	93.9	42.4	75.9	0.0	88.8	59.1	88.0	67.5	73.0	63.5	44.3	62.3
Cylinder3D [57]	IoU	96.4	**61.5**	**78.2**	66.3	**69.8**	**80.8**	**93.3**	0.0	94.9	41.5	78.0	1.4	87.5	50.0	86.7	**72.2**	68.8	63.0	42.1	64.9
AMVNet [83]	IoU	95.6	48.8	65.4	88.7	54.8	70.8	86.0	0.0	**95.5**	**53.9**	**83.2**	0.2	90.9	62.1	87.9	66.8	74.2	**64.7**	49.3	65.2
He et al. [84]	IoU	**96.7**	50.6	**76.2**	**93.1**	62.8	**78.2**	90.5	**0.1**	94.3	47.9	81.8	**5.2**	**91.6**	**63.9**	**88.1**	**70.4**	**74.7**	64.1	**52.5**	**67.5**
DQFormer(Ours)	IoU	**96.8**	48.8	64.3	**93.6**	63.7	76.2	**93.3**	**0.1**	**94.9**	44.2	82.5	0.0	**91.7**	**62.1**	**88.2**	69.5	**74.8**	**67.9**	**51.7**	**66.6**
	PQ	94.3	55.5	67.6	91.1	60.7	79.5	91.5	10.1	94.4	40.3	79.3	0.0	89.1	26.8	87.4	55.4	59.1	64.1	59.8	63.5
	RQ	99.0	67.8	73.2	92.4	64.9	88.7	97.6	10.2	99.8	54.3	92.5	0.0	97.1	36.9	99.8	73.9	78.5	84.5	78.1	73.1
	SQ	95.2	81.8	92.5	98.6	93.6	89.7	93.7	98.6	94.6	74.3	85.6	0.0	91.7	72.7	87.6	75.0	75.2	75.8	76.6	81.7

TABLE XVI

DETAILED PER-CLASS RESULTS ON THE NUSCENES VALIDATION SET, WHICH INCLUDE BOTH SEMANTIC SEGMENTATION AND PANOPTIC SEGMENTATION. BOLD NUMBERS INDICATE THE BEST RESULTS. ALL RESULTS IN [%]

Method	Metrics	Barrier	Bicycle	Bus	Car	Construction	Motorcycle	Pedestrian	Traffic Cone	Trailer	Truck	Drive Surface	Other Flat	Sidewalk	Terrain	Manmade	Vegetation	Mean
Panoptic-PHNet [4]	PQ	53.5	77.5	75.4	90.8	48.6	87.3	91.0	87.0	56.5	72.6	96.7	58.3	72.4	54.9	88.7	84.8	74.7
	RQ	67.7	89.4	80.6	95.5	60.5	95.0	97.4	95.2	65.4	78.6	99.8	67.8	88.0	69.6	99.0	97.0	84.2
	SQ	79.1	86.7	93.5	95.0	80.4	91.9	93.5	91.3	86.4	92.3	96.8	85.9	82.3	78.9	89.6	87.4	88.2
	IoU	77.9	52.4	93.5	93.0	57.0	88.1	83.9	69.9	69.6	86.3	96.9	75.3	76.3	75.3	90.7	88.7	**79.7**
P3Former [31]	PQ	65.0	68.9	77.1	94.1	61.3	85.2	93.0	91.5	60.2	73.0	96.2	59.6	69.3	57.5	86.9	82.9	75.9
	RQ	77.3	79.3	80.3	97.5	67.5	91.5	97.9	97.0	67.5	77.6	99.9	69.3	85.7	73.5	98.3	95.9	84.7
	SQ	84.1	86.9	96.0	96.6	90.8	93.1	95.0	94.3	89.3	94.2	96.3	86.0	80.8	78.2	88.4	86.5	**89.8**
	IoU	68.2	40.3	92.4	83.2	57.0	84.1	76.3	65.1	73.2	85.3	96.5	71.5	74.1	74.8	89.6	87.2	76.8
DQFormer(Ours)	PQ	63.5	74.8	86.8	91.6	57.2	85.1	88.4	81.5	68.1	81.0	96.7	65.8	71.1	59.6	87.2	84.8	**77.7**
	RQ	82.3	92.3	93.2	97.8	72.5	96.2	98.1	96.5	77.3	88.5	99.9	75.5	86.4	75.4	97.9	96.8	**89.2**
	SQ	77.1	81.0	93.1	93.7	79.0	88.5	90.1	84.5	88.1	91.6	96.8	87.2	82.2	79.0	89.1	87.7	86.8
	IoU	73.7	37.8	93.5	87.6	38.4	82.3	71.3	48.9	74.1	83.7	96.9	71.9	74.8	74.3	89.6	86.5	74.1

while maintaining faster speeds than detection and clustering approaches and comparable speeds to MaskPLS. This highlights the efficiency of our unified workflow, which avoids time-consuming detection and clustering processes.

F. Detailed Benchmarks

The class-wise performance of our DQFormer on the SemanticKITTI and nuScenes datasets is detailed in Tables XV and XVI. The results indicate that DQFormer excels in the RQ metric, showcasing its effectiveness in locating and recognizing objects in large-scale scenes. It also performs comparably on the SQ metrics and sets a new state of the art in overall PQ. These findings highlight the success of our decoupled-query paradigm in distinguishing objects and achieving precise segmentation, advancing scene understanding.

V. CONCLUSION

We propose a novel framework named DQFormer using a decoupled query paradigm for unified LPS, aiming to address the challenges posed by foreground objects and background elements in large-scale outdoor scenes. Specifically, we introduce a multiscale query generator that generates semantic-aware queries based on the positions and embeddings of *things* and *stuff* in BEV space. Moreover, we design a query fusion module to integrate queries from multiple BEV resolutions. Finally, we propose a query-oriented mask decoder by utilizing informative queries to guide the segmentation process. Comprehensive experiments on both vehicular and aerial point cloud datasets demonstrate that our DQFormer achieves state-of-the-art performance. Extensive ablation studies and visualization results further demonstrate the effectiveness of our method.

REFERENCES

[1] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "LiDAR-based panoptic segmentation via dynamic shifting network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13085–13094.

[2] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, "EfficientLPS: Efficient LiDAR panoptic segmentation," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1894–1914, Jun. 2022.

[3] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-PolarNet: Proposal-free LiDAR point cloud panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13189–13198.

[4] J. Li, X. He, Y. Wen, Y. Gao, X. Cheng, and D. Zhang, "Panoptic-PHNet: Towards real-time and high-precision LiDAR panoptic segmentation via clustering pseudo heatmap," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11799–11808.

[5] J. Mei et al., "CenterLPS: Segment instances by centers for LiDAR panoptic segmentation," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 1884–1894.

[6] Y. Li et al., "Fully convolutional networks for panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 214–223.

[7] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. NIPS*, Dec. 2021, pp. 17864–17875.

[8] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1290–1299.

[9] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-Net: Towards unified image segmentation," in *Proc. NIPS*, 2021, pp. 10326–10338.

[10] Z. Li et al., "Panoptic SegFormer: Delving deeper into panoptic segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 1280–1289.

[11] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss, "Mask-based panoptic LiDAR segmentation for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1141–1148, Feb. 2023.

[12] S. Su et al., "PUPS: Point cloud unified panoptic segmentation," 2023, *arXiv:2302.06185*.

[13] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[14] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9297–9307.

[15] N. Varney, V. K. Asari, and Q. Graehling, "DALES: A large-scale aerial LiDAR data set for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 717–726.

[16] J. Valeria Hurtado, R. Mohan, W. Burgard, and A. Valada, "MOPT: Multi-object panoptic tracking," 2020, *arXiv:2004.08189*.

[17] Y. Xu, H. Fazlali, Y. Ren, and B. Liu, "AOP-net: All-in-one perception network for LiDAR-based joint 3D object detection and panoptic segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–7.

[18] A. Agarwalla et al., "LiDAR panoptic segmentation and tracking without bells and whistles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 7667–7674.

[19] D. Ye et al., "LiDARMultiNet: Towards a unified multi-task network for LiDAR perception," 2022, *arXiv:2209.09385*.

[20] A. Milioto, J. Behley, C. McCool, and C. Stachniss, "LiDAR panoptic segmentation for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 8505–8512.

[21] E. Li, R. Razani, Y. Xu, and B. Liu, "SMAC-seg: LiDAR panoptic segmentation via sparse multi-directional attention clustering," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 9207–9213.

[22] S. Gasperini, M. N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari, "Panoster: End-to-end panoptic segmentation of LiDAR point clouds," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3216–3223, Apr. 2021.

[23] X. Li, G. Zhang, B. Wang, Y. Hu, and B. Yin, "Center focusing network for real-time LiDAR panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13425–13434.

[24] J. Mei, Y. Yang, M. Wang, X. Hou, L. Li, and Y. Liu, "PANet: LiDAR panoptic segmentation with sparse instance proposal and aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 7726–7733.

[25] L. Nunes et al., "Unsupervised class-agnostic instance segmentation of 3D LiDAR data for autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8713–8720, Oct. 2022.

[26] Y. Zhao, X. Zhang, and X. Huang, "A divide-and-merge point cloud clustering algorithm for LiDAR panoptic segmentation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 7029–7035.

[27] Z. Zhang, Z. Zhang, Q. Yu, R. Yi, Y. Xie, and L. Ma, "LiDAR-camera panoptic segmentation via geometry-consistent and semantic-aware alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 3662–3671.

[28] G. Xian et al., "Location-guided LiDAR-based panoptic segmentation for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1473–1483, Feb. 2023.

[29] F. Hong, L. Kong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "Unified 3D and 4D panoptic segmentation via dynamic shifting networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3480–3495, May 2024.

[30] Y. Gu, Y. Huang, C. Xu, and H. Kong, "MaskRange: A mask-classification model for range-view based LiDAR segmentation," 2022, *arXiv:2206.12073*.

[31] Z. Xiao, W. Zhang, T. Wang, C. C. Loy, D. Lin, and J. Pang, "Position-guided point cloud panoptic segmentation transformer," *Int. J. Comput. Vis.*, vol. 133, no. 1, pp. 275–290, Jan. 2025.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 213–229.

[33] Z. Zong, G. Song, and Y. Liu, "DETRs with collaborative hybrid assignments training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2023, pp. 6748–6758.

[34] A. Petrovai and S. Nedevschi, "Semantic cameras for 360-degree environment perception in automated urban driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17271–17283, Oct. 2022.

[35] J. Mei, M. Wang, Y. Lin, Y. Yuan, and Y. Liu, "TransVOS: Video object segmentation with transformers," 2021, *arXiv:2106.00588*.

[36] J. Mei, M. Wang, Y. Yang, Z. Li, and Y. Liu, "Learning spatiotemporal relationships with a unified framework for video object segmentation," *Int. J. Speech Technol.*, vol. 54, no. 8, pp. 6138–6153, Apr. 2024.

[37] Y. Liu et al., "Multi-space alignments towards universal LiDAR segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 14648–14661.

[38] A. Athar, E. Li, S. Casas, and R. Urtasun, "4D-former: Multimodal 4D panoptic segmentation," in *Proc. Conf. Robot Learn.*, Jan. 2023, pp. 2151–2164.

[39] J. Zhao et al., "SemanticFlow: Semantic segmentation of sequential LiDAR point clouds from sparse frame annotations," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5701611.

[40] Y. Yang et al., "Driving in the occupancy world: Vision-centric 4D occupancy forecasting and planning via world models for autonomous driving," 2024, *arXiv:2408.14197*.

[41] T. He, C. Shen, and A. van den Hengel, "DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 354–363.

[42] T. He, C. Shen, and A. van den Hengel, "Dynamic convolution for 3D point cloud instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5697–5711, May 2023.

[43] Y. Wu, M. Shi, S. Du, H. Lu, Z. Cao, and W. Zhong, "3D instances as 1D kernels," in *Proc. 17th Eur. Conf. Comput. Vision*, Tel Aviv, Israel. Cham, Switzerland: Springer, Jan. 2022, pp. 235–252.

[44] R. Marcuzzi, L. Nunes, L. Wiesmann, E. Marks, J. Behley, and C. Stachniss, "Mask4D: End-to-end mask-based 4D panoptic segmentation for LiDAR sequences," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7487–7494, Nov. 2023.

[45] K. Yilmaz, J. Schult, A. Nekrasov, and B. Leibe, "Mask4Former: Mask transformer for 4D panoptic segmentation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 9418–9425.

[46] Z. Zeng, H. Qiu, J. Zhou, Z. Dong, J. Xiao, and B. Li, "PointNAT: Large-scale point cloud semantic segmentation via neighbor aggregation with transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5704618.

[47] L. He, J. Shan, and D. Aliaga, "Generative building feature estimation from satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4700613.

[48] Z. Luo, Z. Zeng, W. Tang, J. Wan, Z. Xie, and Y. Xu, "Dense dual-branch cross attention network for semantic segmentation of large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5700216.

[49] Z. Guo, R. Xu, C.-C. Feng, and Z. Zeng, "PIF-Net: A deep point-image fusion network for multimodality semantic segmentation of very high-resolution imagery and aerial point cloud," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5700615.

[50] W. Yang, Y. Zhang, X. Liu, and B. Gao, "Scene adaptive building individual segmentation based on large-scale airborne LiDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5706015.

[51] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for LiDAR-based 3D recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17545–17555.

[52] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[53] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[54] Z. Zhou, X. Zhao, Y. Wang, P. Wang, and H. Foroosh, "CenterFormer: Center-based transformer for 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel. Cham, Switzerland: Springer, Oct. 2022, pp. 496–513.

[55] H. Tang et al., "Searching efficient 3D architectures with sparse point-voxel convolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 685–702.

[56] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11784–11793.

[57] X. Zhu et al., "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9939–9948.

[58] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "$(AF)^2$-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12542–12551.

[59] Q. Chen, S. Vora, and O. Beijbom, "PolarStream: Streaming object detection and segmentation with polar pillars," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, May 2021, pp. 26871–26883.

[60] E. Li, R. Razani, Y. Xu, and B. Liu, "CPSeg: Cluster-free panoptic segmentation of 3D LiDAR point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 8239–8245.

[61] R. Razani, R. Cheng, E. Li, E. Taghavi, Y. Ren, and L. Bingbing, "GP-S3Net: Graph-based panoptic sparse semantic segmentation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16056–16065.

[62] M. Liu et al., "Prototype-voxel contrastive learning for LiDAR point cloud panoptic segmentation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 9243–9250.

[63] S. Xu, R. Wan, M. Ye, X. Zou, and T. Cao, "Sparse cross-scale attention network for efficient LiDAR panoptic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, Jan. 2022, pp. 2920–2928.

[64] A. Ošep, T. Meinhardt, F. Ferroni, N. Peri, D. Ramanan, and L. Leal-Taixé, "Better call SAL: Towards learning to segment anything in LiDAR," 2024, *arXiv:2403.13129*.

[65] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.

[66] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[67] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[68] E. Li, R. Razani, Y. Xu, and B. Liu, "CPSeg: Cluster-free panoptic segmentation of 3D LiDAR point clouds," 2021, *arXiv:2111.01723*.

[69] A. Boulch, "ConvPoint: Continuous convolutions for point cloud processing," *Comput. Graph.*, vol. 88, pp. 24–34, May 2020.

[70] C. R. Qi, Y. Li, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1–18.

[71] D. Robert, H. Raguet, and L. Landrieu, "Scalable 3D panoptic segmentation as superpoint graph clustering," in *Proc. Int. Conf. 3D Vis. (3DV)*, Mar. 2024, pp. 179–189.

[72] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D semantic segmentation with superpoint transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17195–17204.

[73] H. Dai, X. Hu, J. Zhang, Z. Shu, J. Xu, and J. Du, "Large-scale ALS point cloud segmentation via projection-based context embedding," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5704216.

[74] B. Guo, L. Deng, R. Wang, W. Guo, A. H.-M. Ng, and W. Bai, "MCTNet: Multiscale cross-attention-based transformer network for semantic segmentation of large-scale point cloud," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5704720.

[75] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[76] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of LiDAR data and building object detection in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 152–165, Jan. 2014.

[77] J. Behley, A. Milioto, and C. Stachniss, "A benchmark for LiDAR-based panoptic segmentation based on KITTI," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13596–13603.

[78] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8277–8286.

[79] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[80] Y. Zhang et al., "PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9601–9610.

[81] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds," in *Proc. Int. Symp. Visual Comput.*, San Diego, CA, USA. Cham, Switzerland: Springer, Oct. 2020, pp. 207–222.

[82] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5240–5250.

[83] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation," 2020, *arXiv:2012.04934*.

[84] X. He, X. Li, P. Ni, W. Xu, Q. Xu, and X. Liu, "Radial transformer for large-scale outdoor LiDAR point cloud semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5708012.

Yu Yang received the B.S. degree in control science and engineering from China University of Geosciences, Wuhan, Hubei, China, in 2021. He is currently pursuing the Ph.D. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His research interests include 3-D perception, generative models, and autonomous driving.

Jianbiao Mei received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 2021, where he is currently pursuing the Ph.D. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering.

His research interests include video segmentation, 3-D perception, and autonomous driving.

Siliang Du received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, Hubei, China, in 2018.

Currently, he is a Researcher with Huawei Technologies Company Ltd., Wuhan. His research interests include 3-D reconstruction, visual location, autonomous driving, and AI infra.

Yilin Xiao received the bachelor's degree from Dalian University of Technology, Dalian, China, in 2019, and the master's degree from Wuhan University, Wuhan, China, in 2022. He is currently pursuing the Ph.D. degree in computer science with The Hong Kong Polytechnic University, Hong Kong.

From 2022 to 2024, he was with Huawei Technologies Company Ltd., Wuhan, with a focus on 3-D vision. His research interests include computer vision, large-language models, and graph neural networks.

Huifeng Wu (Member, IEEE) received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2006.

He is currently a Professor with the Institute of Intelligent and Software Technology, Hangzhou Dianzi University, Hangzhou. His research interests include software development methods and tools, software architecture, embedded systems, intelligent control and automation, and the industrial Internet of Things.

Xiao Xu is currently an Associate Researcher with the Institute of Industrial Technology Research, Zhejiang University, Hangzhou, China. He is also the Deputy Director of the Institute of Technology Transfer, Zhejiang University. His research interests include intelligent control and automation, industrial control systems, and the transfer of scientific and technological achievements.

Yong Liu (Member, IEEE) received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, Zhejiang, China, in 2001 and 2007, respectively.

He is currently a Professor with the Institute of Cyber-Systems and Control, Zhejiang University. His main research interests include robot perception and vision, deep learning, big data analysis, multisensor fusion, machine learning, computer vision, information fusion, and robotics.