International Journal of Computer Vision (2026) 134:71
https://doi.org/10.1007/s11263-025-02704-z

CAT++: Enhancing 3D Annotations with Hierarchical-Interleaved Encoding and Attention-Conditioned Implicit Representation

Xiaoyan Qian[1] · Chang Liu[1] · Xiaojuan Qi[1] · Siewchong Tan[1] · Edmund Y. Lam[1] · Ngai Wong[1]

Received: 22 October 2024 / Accepted: 2 September 2025

Abstract

This work presents CAT++, an enhanced Context-Aware Transformer (CAT), to address the need for more effective 3D point cloud annotation. CAT++ is an end-to-end 3D automatic annotator that lifts 2D weak annotations into 3D, thereby minimizing human annotation burdens. Specifically, it employs a hierarchical-interleaved encoder (HIE) to enhance its encoder-decoder architecture. The HIE alternates between intra- and inter-object Transformer blocks, integrating local and global object features by performing self-attentions along the sequence and batch dimensions. This process continuously combines local and global object features and creates a 3D representation that links interactions between objects at different scales, leading to a more comprehensive understanding of the scene. Moreover, CAT++ features an attention-conditioned implicit neural representation (INR) network, which has been specially engineered to enable more precise and efficient continuous modeling of 3D object surfaces. This improves CAT++'s multi-task learning and its ability to handle hard samples. Results on KITTI and nuScenes datasets show CAT++'s state-of-the-art performance, outperforming other methods by up to 2% AP_{3D} on the KITTI *test* split. Furthermore, CAT++ significantly reduces human annotation effort, by factors of 6.4 and 50 on the KITTI and nuScenes datasets for the Car category, respectively.

Keywords 3D Annotation · 3D Point Cloud · LiDAR · Transformer · Implicit Neural Representation

1 Introduction

Point cloud data has attracted increasing attention in 3D visual tasks due to the widespread use of LiDAR sensors in areas such as autonomous vehicles and robotics. This proliferation has spurred the evolution of 3D object detectors (Shi et al., 2019; Lang et al., 2019; Xu et al., 2021), designed to categorize and localize objects from 3D sensor data (e.g., LiDAR point clouds). However, these detectors necessitate extensive 3D ground-truth box annotations for reliable supervision signals. While modern LiDAR scanning technology enables the acquisition of 3D data, manually annotating these 3D point clouds is often labor-intensive, susceptible to errors, and at times impractical in certain real-world scenarios (Wei et al., 2021; Qin et al., 2020; Liu et al., 2022b). This motivates us to explore 3D automatic annotations with human-level labeling performance while only requiring lightweight human annotations.

Early studies on the 3D annotation have explored automatic annotation models that employ weak labels, e.g., 2D boxes (Wei et al., 2021; Caesar et al., 2020; Liu et al., 2022b, a; Qian et al., 2023; Paat et al., 2024), center-clicks (Meng et al., 2020, 2021), or 2D segmentation masks (McCraith et al., 2021; Wilson et al., 2020). While being effective, such intricate annotation methods have necessitated multi-stage training models (Wei et al., 2021; Liu et al., 2022b; Meng et al., 2020, 2021), additional point

Communicated by Jianfei Cai.

✉ Xiaoyan Qian
qianxy10@connect.hku.hk

Chang Liu
lcon7@connect.hku.hk

Xiaojuan Qi
xjqi@eee.hku.hk

Siewchong Tan
sctan@eee.hku.hk

Edmund Y. Lam
elam@eee.hku.hk

Ngai Wong
nwong@eee.hku.hk

[1] Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong, People's Republic of China

Published online: 20 January 2026

cloud processing (Meng et al., 2020; Liu et al., 2022b, a; Paat et al., 2024), and the creation of unique 3D operators and objective functions (Meng et al., 2020, 2021; Wilson et al., 2020). These multi-step designs bring complexity to the process, making it difficult to adapt and fine-tune the models. Moreover, as the accuracy of later stages heavily relies on the correctness of the initial stages, the error propagation could be substantial, leading to a cumulative negative impact on the final results.

The major obstacle to 3D annotation lies in point clouds' sparse nature, especially in hard samples with few data points. In response, multimodal annotators (Liu et al., 2022a, b; Paat et al., 2024) leverage images and point clouds to extract information from each modality and integrate them for comprehensive predictions in sparse point cloud tasks. Despite remarkable performance on the KITTI benchmark (Geiger et al., 2012), these methods tend to raise model intricacy and calibration errors for training as they merge multimodal information. Moreover, only about 5% of camera information aligns with the LiDAR point in the camera-to-LiDAR projection, and LiDAR-to-camera projections often result in a loss of in-depth information in 3D point clouds (Liu et al., 2022c). This disparity intensifies with sparser point clouds, as projections between camera and LiDAR in multimodal information combinations are often semantically or geometrically deficient. Furthermore, they often overlook the inter-object feature relation particularly informative to hard samples.

However, we have observed that employing inter-object feature relations can effectively guide global feature learning on point cloud instances. This is achieved through self-attention along the batch dimension. This approach is particularly informative for hard samples in hard tasks of 3D object detection. As a result, it eliminates the need for fusing multimodal information with semantic and geometric losses. Hard samples with few points can refine their features by interacting with other easy samples through inter-object relations instead of explicitly interpolating point clouds. The inter-object feature relations can enhance sample interactions (Hou et al., 2022a, b), thus aiding in annotating such hard samples that are heavily truncated, occluded, or distant. Existing research on 3D annotation has primarily neglected the exploration of inter-object relations, often focusing more on individual object characteristics and local features. This oversight can lead to a lack of contextual understanding of the objects within a scene, limiting the model's ability to recognize patterns and spatial relations. Consequently, this can hinder the model's performance in 3D automatic annotations, which requires a more comprehensive grasp of the object's characteristics.

Furthermore, we have examined that implicit neural representation (INR) can decipher such intricate characteristics of point cloud objects implicitly. INR can learn a continuous function representation of the entire object shape instead of dealing directly with individual discrete points (Mescheder et al., 2019; Peng et al., 2020; Sitzmann et al., 2020). This can implicitly supplement the gaps in the representation of hard samples and provide a more robust representation that is not as dependent on the specific points in the original point cloud. By encoding 3D object geometry into a neural network, INR simplifies the problem of learning a continuous function that represents the underlying structure of the point cloud (Mescheder et al., 2019). This shift in focus presents a promising avenue for enhancing the effectiveness of 3D automatic annotation by addressing its current limitations on hard samples.

To this end, this paper introduces CAT++, an advanced version of the context-aware transformer (CAT) model (Qian et al., 2023) tailored explicitly for end-to-end 3D point cloud annotation. CAT++ forms a unified model for 3D annotation tasks by integrating a Hierarchical-Interleaved Encoder (HIE) with an attention-conditioned INR network, specifically adapted for our multi-task learning objective. The HIE consists of alternately linked intra- and inter-object blocks. This arrangement continuously refines local and global features, establishing a hierarchical 3D representation that progressively abstracts intra-object details and inter-object relations. With a HIE, CAT++ can understand 3D structures across multiple scales and effectively manage hard samples with inter-object relations. Complementing the HIE, we extend CAT++ with INR capability, allowing it to learn a continuous representation of 3D objects rather than merely processing discrete points. In particular, the INR predicts point occupancy and generates a continuous decision boundary indicating the perceived surface of the object. This continuous representation offers a more comprehensive view of the object and significantly contributes to generating 3D bounding boxes, especially for hard samples. Utilizing HIE and INR, CAT++ can refine and enhance initially incomplete representations of hard samples, thereby improving the accuracy of 3D box generation. Consequently, CAT++ streamlines the annotation process by removing cumbersome designs such as 3D segmentation, cylindrical object proposal generation, point completion, multimodal information combination, and complicated loss function designs in existing annotation methods. This refined approach also removes the necessity for a CNN backbone on image features and solely relies on 3D point clouds and Transformers trained from scratch. CAT++ is marked by its simplicity and elegance, which sets a new performance benchmark on established datasets of KITTI and nuScenes.

Different from its conference counterpart (Qian et al., 2023), here we evolve CAT into CAT++, making significant enhancements that bring new state-of-the-art performance: (1) We propose the HIE, which facilitates a more effective exchange of information between local (intra-object) and global (inter-object) perspectives. Unlike CAT, which

adopted a rigid bottom-up hierarchy where all the intra-object encoding is completed before moving on to the inter-object encoding, CAT++ implements interleaved operations to create a continuous feedback loop and refine local and global features at multiple scales. This innovation enables the model to better merge information across different scales, crucial for understanding hard samples with hierarchical structures within and between objects. (2) In our CAT++ model, we refine the conventional INR network to better address the complexities of 3D annotation. Instead of using a PointNet (Qi et al., 2017a) for condition variables as in OccNet (Mescheder et al., 2019), we introduce a learnable INR token conditioned by an attention mechanism, as shown in Fig. 2. This novel approach allows for dynamic feature extraction and seamless information sharing between INR tokens, point tokens, and box tokens. It enriches the model's contextual and geometric comprehension, providing a more nuanced understanding of object relations and conditions. Consequently, this enhanced INR implementation enables CAT++ to offer a continuous and smooth spatial representation of objects, filling in the information gaps, especially in sparse areas of complex samples. (3) Our revised training procedure works as multi-task learning (see Fig. 2), focusing concurrently on discrete point-wise features from the HIE and the continuous function from the INR. We investigate the synergistic integration of discrete point-wise representation and continuous implicit representation as a novel approach to enhance the capabilities of 3D learning algorithms. This multi-faceted learning strategy strengthens CAT++'s overall robustness, resulting in more precise 3D bounding box estimations (as evidenced in Tables 5 and 8). Moreover, it improves the performance of PointRCNN - when trained using our annotations - in challenging tasks, as confirmed by the Hard Task sections in Tables 1 and 2. (4) Finally, we have extended the application of CAT++ to include the larger and sparser nuScenes dataset, as well as smaller objects within the Pedestrian category. This expansion has demonstrated the robustness and effectiveness of CAT++ across varied data environments and multiple categories.

2 Related Work

We first review some prior work in 3D point cloud representation learning and 3D object annotations from weak annotations.

2.1 Point Cloud Representation Learning

In contrast to 2D images, where pixels are organized in regular grids and quickly processed by Convolutional Neural Networks (CNNs), 3D point clouds present a unique set of challenges. These data sets are inherently sparse, unstructured, and dispersed across a 3D space, rendering traditional convolutional architectures less effective. Two tailored approaches have been commonly used to process point clouds: transforming irregular points into voxels or grids, and directly consuming point clouds. Voxel-based networks (Maturana & Scherer, 2015; Wu et al., 2015; Xie et al., 2018; Yang et al., 2018) divide point clouds into volumetric grids and employ 2D/3D CNNs for predictions. Point-based methods (Qi et al., 2017a, b; Lang et al., 2019; Shi et al., 2019; Qi et al., 2018) prefer to process raw point clouds directly and better preserve the original geometric information crucial for 3D tasks.

Recently, attention-based models have been introduced as an innovative means to represent 3D point clouds for a variety of 3D vision tasks, including retrieval (Zhang & Xiao, 2019; Ye et al., 2022), segmentation (Zhao et al., 2021; Wu et al., 2022; Lai et al., 2022; Park et al., 2022), object classification (Liu et al., 2020; Zhao et al., 2021; Wu et al., 2022; Yu et al., 2021), and 3D object detection (Misra et al., 2021; Pan et al., 2021; Park et al., 2022).

The original Point Transformer (Zhao et al., 2021) features a bespoke Transformer layer incorporating vector self-attention that is then integrated into an Unet-like architecture, specifically designed for 3D point cloud segmentation and classification. Building on this, Point Transformer V2 (Wu et al., 2022) introduces grouped vector attention and partition-based pooling, enhancing the model's ability to handle complex spatial hierarchies and improving scalability in processing large point clouds for 3D representation learning. Stratified Transformer (Lai et al., 2022) focuses on 3D point cloud segmentation by stratifying the processing of point clouds into more manageable sub-tasks, which improves both the accuracy and efficiency of segmentation tasks. PCT (Guo et al., 2021) introduces modifications to the Transformers for 3D, creating an offset-attention and neighbor embedding, thereby making the Transformer framework suitable for point cloud learning. PoinTr (Yu et al., 2021) leverages a Transformer encoder-decoder architecture to reposition point cloud completion as a set-to-set translation problem. These models take advantage of the Transformers' robust capabilities in processing long-range token-wise interactions and enhancing information communication, heralding a new era of their application in 3D tasks.

While Transformer-based models for 3D have exhibited promising potential, their current designs bring in handcrafted inductive biases (e.g., local feature aggregation, neighbor grouping, and embeddings). Moreover, these models overlook the crucial aspect of inter-object relations that are informative for effective learning in hard samples. Our focus is to extend the capabilities of the general Transformer for comprehensive extraction of both intra- and inter-object features in 3D.

Transformers have primarily focused on explicitly learning point representation for 3D. Recently, this has been generalized to INR that implicitly learns nonlinear decision boundaries for 3D object shapes/surfaces (Mescheder et al., 2019; Niemeyer et al., 2020; Yan et al., 2022). INR provides significant advantages over traditional discrete representations by defining continuous and differentiable signals, such as images, shapes, and audio, through neural network parameterization (Mescheder et al., 2019; Peng et al., 2020; Sitzmann et al., 2020). For 3D point clouds, INR directly learns the mapping from parametric dependence to the nonlinear decision boundaries on object surfaces. Thus, unlike classical methods that solve one dataset instance, they learn an entire family of object surfaces. Inspired by INR's capabilities in approximating functions for smooth representation of 3D structures, we explore combining discrete point-wise representation with continuous implicit representation for enhanced 3D learning. This integrated approach aims to yield an advanced understanding of point cloud data in 3D space, especially in challenging scenarios such as hard samples that are heavily truncated, occluded, or situated at a distance.

2.2 3D Object Annotation from Weak Labels

3D manual annotations for point cloud data is a time-consuming process (Meng et al., 2021; Wei et al., 2021; Liu et al., 2022a, b), limiting the ability to train high-quality 3D object detection models on large datasets (Geiger et al., 2012; Caesar et al., 2020; Sun et al., 2020). Acquiring 2D weak annotations is often easier and less costly than 3D labels, so research on 3D object annotation derived from weak labels is growing. Not many studies have explored 3D automatic annotation models. FGR (Wei et al., 2021) first uses 2D boxes and a heuristic algorithm to identify key points and edges in the sub-point-cloud frustum, construct coarse 3D segmentation, and finally estimate the 3D box. SDF (Zakharov et al., 2020) uses predefined models to estimate 3D annotations from 2D bounding boxes and segmented 3D point clouds. WS3D (Meng et al., 2021, 2020) utilizes center clicks as weak annotations to generate cylindrical object proposals on bird's eye view (BEV) maps and refine them to obtain 3D labels. MAP-Gen (Liu et al., 2022b) employs a three-stage automated 3D-box annotation workflow using 2D weak boxes: initial foreground segmentation, point generation, and final 3D box generation. MTrans (Liu et al., 2022a) uses LiDAR scans and camera images to generate precise 3D labels from weak 2D bounding boxes via a multi-task design of segmentation, point generation, and box regression. MEDL-U (Paat et al., 2024) is a recent advancement based on the multimodal transformer framework (MTrans). Building upon MTrans, MEDL-U introduces evidential deep learning techniques to quantify annotation uncertainty explicitly. Specifically, MEDL-U augments MTrans by adding uncertainty-aware prediction heads that estimate the reliability of the automatically generated 3D labels, thereby improving the robustness and interpretability of the annotation process.

While they have demonstrated state-of-the-art performance on the KITTI benchmark, such complicated annotation designs involve pipelined training models, intermediate processing points, designing special loss functions, and combining multimodal information. Moving away from these conventionally complex designs, we introduce CAT++, a model that brings a streamlined yet effective auto-labeling process into the domain of annotation methods. Current SOTA MEDL-U inherits the complexity of MTrans, including dependence on multimodal data inputs and a multi-stage pipeline. In contrast, our proposed CAT++ achieves comparable or superior performance through a simpler, unified, single-modality approach that does not rely on external sensor modalities or extensive multi-stage refinement. Our approach completely dispenses with the need for a CNN backbone on image features, relying solely on 3D point clouds and Transformers, which are trained from scratch. Our model is distinguished by its simplicity, stripping away convoluted implementations in favor of an approach that is efficient but also intuitive and easy to implement.

3 Methodology

3.1 Data Preparation

Our goal is to generate 3D pseudo labels that can be used to train any off-the-shelf 3D object detector (e.g., PointRCNN). Given LiDAR point clouds and weak 2D boxes, we first extract the frustum area tightly fitted with 2D bounding boxes, following (Qi et al., 2018; Wei et al., 2021; Liu et al., 2022a, b). With a known LiDAR-to-Image calibration matrix, a 3D point cloud in (x, y, z) can be projected onto its 2D image plane (m, n) by the mapping function f_{cal}. The 2D projected sub-cloud $\mathcal{P}_{2D} \in \mathbb{R}^{N\times 2}$ within a 2D box can be defined as:

$$\mathcal{P}_{2D} = \{(m, n) \mid (m, n) = f_{cal}(x, y, z), (m, n) \in \mathcal{B}\}. \quad (1)$$

Therefore, the frustum sub-cloud $\mathcal{P}_F \in \mathbb{R}^{N\times 3}$ within its 2D bounding box can be extracted as:

$$\mathcal{P}_F = \{(x, y, z) \mid f_{cal}(x, y, z) \in \mathcal{P}_{2D}\}, \quad (2)$$

where (x, y, z) is the point coordinate, and $\mathcal{B}$ is the 2D region in the 2D box. We use N to represent input frustum sub-cloud size (i.e., the number of points).

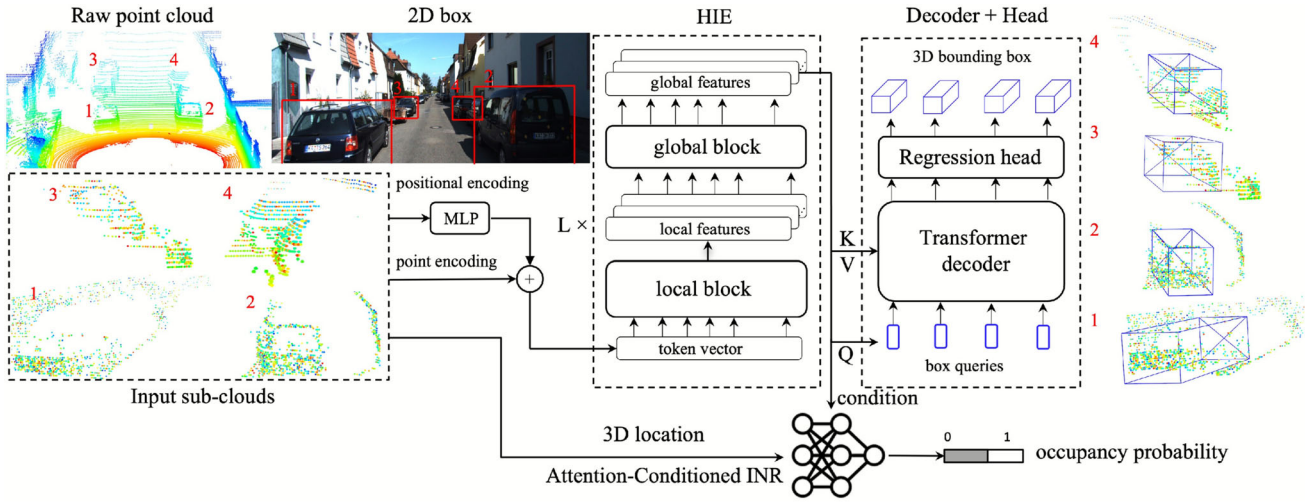

Fig. 1 CAT++ is an end-to-end trainable Transformer-based model designed for automatic 3D annotation. It intakes frustum sub-clouds within 2D boxes and outputs corresponding 3D bounding boxes. The model comprises L HIE stages, which alternate between intra-object (local) and inter-object (global) blocks. The local block captures intra-object information within a given object, while the global one extracts cross-object feature relations for point feature learning. CAT++ employs an attention-conditioned INR along with the HIE, facilitating multi-task learning. This INR represents an implicit occupancy field, namely, a continuous decision boundary of the object's surface. A decoder segment of the model receives box tokens corresponding to 3D box representations, called box queries (Q). These box queries attend to point features (K and V) output by the HIE. The decoder's results are passed to regression heads (i.e. MLPs), generating the final 3D bounding boxes

3.2 CAT++: Enhanced Context-Aware Transformer

CAT++ model processes frustum sub-clouds, denoted as $\mathcal{P}_F \in \mathbb{R}^{N\times 3}$, derived from 2D bounding boxes as its inputs. Each input comprises B sub-clouds $\mathcal{P}_F$, where B represents the batch size. CAT++ outputs a set of 3D bounding boxes, represented as $Box_{3D} \in \mathbb{R}^{B\times 7}$. A frustum sub-cloud input is essentially an unordered set of points associated with 3D coordinates (x, y, z). The number of points can vary significantly among different sub-clouds. Dense sub-clouds can exist in areas where LiDAR scans are complete, while other areas may contain fewer points due to occlusion, truncation, limitations in sensor resolution, or variations in the light reflection. We employ random sampling to standardize input size across diverse sub-clouds, as described in (Qi et al., 2018; Wei et al., 2021; Liu et al., 2022a, b). This process ensures that each sample maintains a consistent size, i.e., $N = 1024$ points. Following this, we utilize a multilayer perceptron (MLP) to transform these points into $d = 512$ dimensional point tokens. These data are subsequently passed through our CAT++ encoder (HIE) to create a corresponding set of point features.

Alongside these, CAT++ incorporates an attention-conditioned implicit neural network for INR learning, complementing point-wise discrete representations from HIE. This integration facilitates a multi-task learning strategy. The final point features encapsulate the local and global information from the HIE and the INR learned from the implicit neural network. We introduce additional box and INR tokens into the HIE for our box regression. These tokens act as box representations, INR conditions, and point transformations. Finally, a decoder utilizes such box tokens as queries (Q), while using point features as keys (K) and values (V) for the accurate prediction of 3D bounding boxes. A visual representation of CAT++ is depicted in Fig. 1.

3.3 HIE for Point-Wise Discrete Representation

The HIE in CAT++ is an architectural enhancement that serves a crucial role in interpreting 3D point clouds in a point-wise discrete manner. This novel configuration alternates between intra-object (local level) and inter-object (global level) blocks in Fig. 1. This internal hierarchical and external interleaved architecture encapsulates the essence of the HIE, vividly depicting the dynamic interaction between local object details and the broader scene context across multiple iterations.

3.3.1 Local Block

The input transformation step provides a set of N features of $d = 512$ dimensions using an MLP with two hidden layers, as point tokens. The intra-object block (local) in HIE inputs the 512-d point tokens, seven box tokens, and one INR token. This results in producing $N + 7 + 1$ features, each of 512 dimensions. These seven box tokens represent

the attributes of the 3D bounding box, including coordinates (x, y, z), dimensions (l, w, h), and yaw angle. This block involves a sequence of operations: layer normalization, followed by a multiheaded self-attention and an MLP. We use vanilla self-attention in (Yu et al., 2021) to extract the feature relations between all token vectors. The local block performs self-attention along the sequence dimension $(N + 7 + 1)$, learning to associate points that are close to each other or have similar features.

Our model facilitates communication between all point pairs within an object, ensuring each point's equal contribution to information exchange. We acknowledge that valuable context can be embedded anywhere within the object's architecture, not necessarily confined to apparent or foreground points. Therefore, our model also incorporates background points within the object, enabling them to participate actively in this communication. Although seemingly peripheral, these points offer crucial insights into defining the object's boundaries, thereby assisting in the precise generation of 3D boxes. We tap into the complete data spectrum by considering every point, foreground and background, painting a more comprehensive, nuanced picture of the object's 3D structure. Such an in-depth understanding, in turn, empowers our model to yield exact 3D bounding box generations.

3.3.2 Global Block

The inter-object (global) block, a novel aspect of HIE, enables communication between different objects, marking a significant advancement in handling global context. The global block within HIE is designed to capture interactions between different objects. It achieves this by implementing self-attention along the batch dimension (B). This contrasts with the local block, which focuses on the sequence dimension $(N + 7 + 1)$. Through this distinctive attention strategy, the global block can extract higher-order dependencies and relational features across multiple objects, facilitating a more comprehensive understanding of the scene. The local block generates output highlighting local feature relationships in the $B \times (N+7+1) \times d$ dimension. We transpose this output to $(N+7+1) \times B \times d$ before the inter-object block processes it. This transposition allows each object associated with its related box and INR tokens to engage with all other objects within the mini-batch during training.

Promoting the exchange of messages across multiple objects creates a more enriched and interconnected representation. As a result, our model can build an interconnected perspective of the entire scene, enhancing its understanding of inter-object interactions and contextual dynamics. This method of interlinking objects and recognizing their mutual influences is particularly advantageous when dealing with hard samples, where traditional methods often suffer. In such challenging cases, the model identifies interrelated objects through attention weights, utilizing their relationships to complement the features of hard samples. This holistic approach fills potential gaps in understanding hard samples and enhances the quality of generated 3D boxes.

3.3.3 Hierarchical-Interleaved Encoding

Within the structured framework of HIE, the intra-object block serves as the foundational layer of the hierarchy. It focuses on extracting and analyzing detailed features within each object. By scrutinizing these specific details, this block ensures that the nuances and subtleties inherent to each object are captured and encoded separately, which results in a fine-grained representation of each object's structure. In the HIE hierarchy, the inter-object block operates at a higher level, focusing on the broader relationships and interactions among multiple objects (higher-level features). This includes understanding how the objects relate to one another in a 3D scene, permitting a higher-level perspective. This local-global viewpoint forms the basis of the hierarchical encoding in the HIE, thus generating a multi-layered, progressively abstract 3D representation at multiple scales to encompass the detailed information of individual objects and their broader interrelations. This comprehensive perspective equips CAT++ with the ability to understand 3D structures and their relations at multiple levels of granularity, with proven benefits for complex 3D annotation tasks.

Interleaved Encoding is another component of HIE. This refers to how intra- and inter-object blocks are linked to conduct self-attentions along the sequence and batch dimensions. This unique feature enables the continual alternation between intra- and inter-object blocks, enabling concurrent examination of sequence and batch dimensions. Specifically, after the first block of intra-object encoding and the following inter-object encoding, the output of the inter-object encoding feeds back into the system for the next round of intra- and inter-object encoding. This interleaved process progressively refines local and global features by oscillating between the two dimensions of attention, permitting the model to appreciate granular details and broader relational context comprehensively. The HIE captures local geometric features and more general spatial relationships through this iterative, layer-by-layer processing of points, ultimately encoding a compact and multi-faceted representation of the entire 3D structure. This enhances CAT++'s capability to manage complex point cloud data (e.g., hard samples).

3.4 Attention-Conditioned Implicit Neural Network for INR

With INR, point clouds are represented as a neural function that maps continuous coordinates to the occupancy field. We incorporate an attention-conditioned implicit neu-

ral network inspired by the Occupancy Network (OccNet) framework (Mescheder et al., 2019), emphasizing learning INRs of point clouds. Implicit neural networks offer a method of predicting whether a point belongs to an object. This is achieved by learning a continuous decision boundary that predicts the likelihood of a point being inside (occupied) or outside (unoccupied). The main idea can be formulated as (Mescheder et al., 2019):

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1], \tag{3}$$

where f_θ is the occupancy network function;a $\mathbb{R}^3$ represents the 3D coordinates in space; $\mathcal{X}$ is the condition, or the latent code vector (e.g., point features); the range [0,1] represents the predicted occupancy probability of the 3D point (with 1 indicating the point inside the object, and 0 outside). The continuous decision boundary function f_θ encapsulates information about the underlying geometry and helps define the object's shape. We refer readers to OccNet (Mescheder et al., 2019) for more details on the occupancy field for continuous decision boundary learning.

In our attention-conditioned INR network, we take as input the 3D location coordinates (x, y, z), denoted as $\mathbb{R}^3$, and associate these with a corresponding initialized INR token. This token, a learnable vector parameter in $(B, 1, d)$ dimensions, acts as the conditional variable $\mathcal{X}$ for the implicit neural network in Eq. 3. We only utilize a single token for this dependent learning instead of resorting to a PointNet used in the original OccNet. The INR can be simply parameterized by a neural network f_θ that takes a pair $((x, y, z), INR\_token)$ as inputs and outputs an occupancy probability. Once formed, the INR token and point and box tokens are incorporated into our HIE process (see Fig. 2). This integrated handling with the attention mechanism allows for the realization of three key functionalities:

1. **Feature capture**: It enables the extraction of crucial information for INR conditions from observed data by facilitating interactions among point tokens. This process aids the capture of object-specific features for INR conditions in Eq. 3.
2. **Information transfer**: It encourages the seamless exchange of information between point tokens, box tokens, and condition tokens (INR tokens). This synergy allows the model to establish a meaningful and valuable relationship between discrete points and their continuous representation.
3. **Condition relations**: It paves the way for understanding the relations of different object conditions for INR learning through inter-object interactions. Global comprehension contributes to a holistic understanding of the entire object shape.

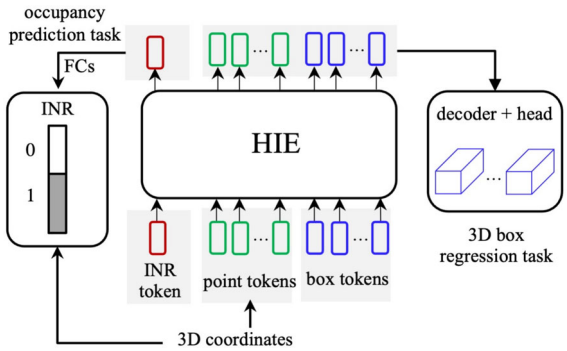

Fig. 2 Multi-task learning of 3D box regression and occupancy prediction. We employ a multi-task learning strategy that learns both explicit point-wise representation from the discrete data and a continuous implicit representation. The LiDAR point cloud data, characterized by 3D coordinates, is transformed into point tokens via an MLP. This data is then used alongside initialization tokens (learnable vector parameters), which consist of one INR token and seven box tokens, for our HIE and INR learning. The INR token is fine-tuned at the output of the HIE. With the 3D coordinates in place, it is then mapped to the conditions in the INR using layerwise fully connected layers (FCs), facilitating INR learning. Concurrently, the box and point tokens are fed into the decoder and the associated heads for 3D bounding box regression. Through this approach, our model effectively and concurrently learns both the discrete, point-wise features of the data and a continuous, implicit representation of the object's 3D structure. This combined learning process ultimately contributes to accurately generating the 3D bounding boxes

In CAT++, attention-conditioned INR works to learn this f_θ in Eq. 3, such that given a new 3D point and the associated condition, it can predict whether this point is inside or outside the object. It focuses on implicitly understanding an object's structure within a 3D space. Unlike traditional point cloud processing, which relies on discrete points, the implicit neural network is trained for the continuous decision boundary of classifier learning. This classifier creates a decision boundary, forming a continuous and differentiable occupancy function. This occupancy function serves as INR and allows CAT++ to learn a continuous approximation of an object's surface. This is particularly useful for modeling objects when point clouds are sparse or incomplete due to occlusion or truncation.

The integration of occupancy-based 3D representation with discrete point-wise learning in our multi-task strategy (as shown in Figs. 1 & 2) enriches the point cloud's feature learning, enhancing CAT++'s ability to generate accurate 3D bounding boxes even for hard samples. The INR supplements the 3D representation learning within CAT++, boosting automatic 3D annotation accuracy and efficiency.

3.5 Decoder for 3D Bounding Box Prediction

CAT++ model incorporates a decoder specifically designed to refine object-level features, ultimately enhancing the gen-

eration of 3D bounding boxes. The decoder uses box tokens as queries to attend to point features from the HIE output. This can help our model focus on the most relevant parts of the point cloud for 3D box regression.

Our approach follows previous work in treating the B objects in parallel within a batch using multiheaded self- and cross-attention mechanisms. The box queries are learned tokens encompassing seven components in (x, y, z, l, w, h, yaw). The decoder uses the final point features as keys and values (K, V) and box tokens as queries (Q). The output of the decoder is then employed to generate 3D boxes, $Box_{3D} \in \mathbb{R}^{B\times 7}$.

In our setup, the box queries represent 3D boxes in the 3D space, around which the final 3D boxes are generated. These boxes are produced by three separate MLP heads predicting locations (x, y, z), dimensions (l, w, h), and rotation along the z-axis (yaw), respectively. Each MLP head includes a linear layer and leaky ReLU nonlinearity. Thus, our 3D adaptation of the Transformer decoder aligns with existing methods (Yu et al., 2021, 2022) with minimal modifications.

3.6 Loss Function for Multi-Task Learning

In the multi-task learning setup, we use the HIE to regress 3D boxes and an attention-conditioned INR network for occupancy prediction. Each task contributes uniquely to the overall understanding of the 3D structure, and learning them simultaneously can enhance performance on both tasks, as demonstrated in Fig. 2. To train our CAT++ model, we construct a compound loss function, including a dIoU loss $(\mathcal{L}_{box})$, a direction loss $(\mathcal{L}_{dir})$, and the occupancy prediction loss $(\mathcal{L}_{occ})$.

The dIoU loss $\mathcal{L}_{box}$, as suggested by Zheng et al. (2020), measures the discrepancy between the predicted bounding boxes and the corresponding ground truth boxes. It helps ensure that the predicted 3D boxes accurately represent the objects in the point cloud. Next, the direction loss $\mathcal{L}_{dir}$ serves to tackle the inherent direction-invariance of the IoU metric, as identified by Liu et al. (2022a). We use a binary direction classification, labeling the orientation within the range of $[-\pi/2, \pi/2)$ as the front and $[\pi/2, \pi] \cup [-\pi, -\pi/2)$ as the back. This classification is conducted by a standalone MLP, aiding our model in accurately determining the direction of 3D boxes, an essential aspect for yaw regression. Cross-entropy loss is utilized for this direction loss $\mathcal{L}_{dir}$. The occupancy prediction loss $\mathcal{L}_{occ}$ is derived from the implicit neural network of the OccNet system. This loss measures the difference between the predicted and actual occupancy of a 3D point in space. By minimizing this loss, the model can more accurately predict whether a given point in space is within (occupied) or outside (unoccupied) a particular object, thus contributing to a more precise 3D representation.

These combine into a weighted sum to form the final loss function for CAT++:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{occ}\mathcal{L}_{occ} + \mathcal{L}_{dir}, \tag{4}$$

where the λ_{box} and λ_{occ} are weight coefficients that control the contribution of each loss component. Our work empirically sets λ_{box} to 5 in MTrans, MEDL-U, and CAT (Liu et al., 2022a; Paat et al., 2024; Qian et al., 2023), while λ_{occ} is tuned based on the model's performance on different validation sets.

KITTI: $\lambda_{occ} = 1$. We found that a weight of 1 provided a good balance on KITTI, whose point clouds are moderately dense (Velodyne 64-beam LiDAR). This value ensures the occupancy (shape) task has a comparable influence to the box regression task. We initially tried a range of values $(0.5-2.0)$ and observed that 1.0 yielded stable convergence without excessive favoring one task. Under $\lambda_{occ} = 1$, the gradients of the two tasks are of a similar order of magnitude during training.

nuScenes: $\lambda_{occ} = 2$. The nuScenes dataset is considerably sparser (32-beam LiDAR), providing limited point information per object. To address the inherent sparsity, we increased λ_{occ} to 2.0, emphasizing the implicit neural representation task. This higher setting significantly enhances the model's capability to infer continuous, complete object shapes from sparse points, thereby substantially improving annotation quality without negatively affecting annotation accuracy, as confirmed through cross-validation. This setting was tuned by cross-validation: we noticed that using the same weight as KITTI caused a minor drop in annotation accuracy, whereas 2 preserved 3D annotation performance while still learning meaningful occupancy patterns. Meanwhile, annotation metrics $mIoU$ and AP_{3D} remained nearly constant, indicating that even with a larger λ_{occ}, the model can jointly learn both tasks effectively.

ScanNet: $\lambda_{occ} = 0.5$. ScanNet provides dense 3D geometry via high-resolution RGB-D reconstruction. Under these conditions, the complementary role of the INR is less critical, as the detailed geometric structure is already well represented by the dense point cloud data. Thus, a reduced λ_{occ} of 0.5 effectively prevents overemphasis on detailed shape reconstruction, maintaining the model's primary focus on accurate 3D annotation. We tuned this by incrementally decreasing λ_{occ} until the occupancy accuracy on validation frames leveled off (below 0.5, we saw no further gain). This lower setting proved optimal through validation experiments, achieving a stable balance between annotation performance and shape detail representation.

By strategically incorporating these losses within a multi-task learning framework, the network is inherently motivated to learn features and representations that effectively contribute to both tasks. This symbiotic learning paradigm fos-

ters an enriched understanding of object structure, enabling the model to capture subtleties intrinsic to each task. Consequently, this unified approach strengthens the model's generalization capacity, effectively tackling various scenarios. The dual-task learning approach significantly improves the model's accuracy and robustness, especially when annotating challenging samples.

3.7 Position Encoding

In contrast to traditional methods that employ manually crafted positional encodings like sinusoidal position embedding or normalized range values, our CAT++ model uses a more adaptive approach tailored explicitly to 3D point cloud processing. Given the intrinsic 3D coordinates (x, y, z) in point clouds, we utilize an MLP to generate trainable, parameterized positional embeddings, leading to better adaptation to the 3D structure. We model our 3D position pos as follows:

$$pos = MLP(x, y, z). \tag{5}$$

This encoded position is then added to point tokens, followed by a concatenation of all tokens, forming a unified sequence $(box\_tokens, inr\_token, point\_tokens + pos)$ for the hierarchical-interleaved encoding process. To ensure dimensional consistency, we apply an additional MLP layer to enrich the positional context of each entity within the sequence. With its simplicity and effectiveness, this design consistently outperforms traditional encoding schemes.

4 Experiments

4.1 Experimental Setup

We conduct experiments on two widely used LiDAR 3D object detection datasets: KITTI (Geiger et al., 2012) and nuScenes (Caesar et al., 2020). Pre-processing datasets is performed as in Wei et al. (2021); Liu et al. (2022a, b), where we extract objects enclosed within 2D bounding boxes.

KITTI Dataset. The detection benchmark in KITTI comprises 3,712 training frames with 15,654 vehicle instances and 3,769 validation frames. For fairness in comparison, we adhere to the standard split of 500 frames for training and 3,769 for validation. We follow the official evaluation protocol provided by KITTI. Detection outcomes of CAT++-trained PointRCNN are evaluated on three standard tasks: Easy, Moderate, and Hard Tasks. We report the detection performance using the mean Average Precision (mAP) threshold of 0.7 mean Intersection over Union (mIoU) for 3D and Bird's Eye View (BEV).

nuScenes Dataset. The nuScenes dataset (Caesar et al., 2020) is a more recent large-scale outdoor 3D detection dataset. It contains 1,000 driving scenes with 40,157 annotated samples, including 28,130, 6,019, and 6,008 frames for training, validation, and testing. The sample consists of 10 categories with a long-tail distribution. nuScenes uses a 32-line LiDAR, which produces approximately 30k points per frame. It is way fewer than KITTI's 118k points per frame (64-line LiDAR). For 3D detection, the main metrics are mAP and nuScenes detection score (NDS) (Everingham et al., 2010). nuScenes calculates mAP based on the 2D center distance on the ground plane rather than IoU in KITTI. NDS is a weighted average of mAP, translation, scale, orientation, velocity, and other box attributes (Caesar et al., 2020). We compare CAT++-generated 3D bounding boxes with human annotations on the metrics of mIoU, NDS, Recall with an IoU threshold of 0.7, and Recall with a location error (LE) threshold of 0.5. These metrics measure the overlap level of our generated 3D bounding boxes and ground truth boxes, which ensures the quality of CAT++-generated annotations.

ScanNet Dataset. The ScanNet Dataset (Dai et al., 2017) comprises 1,513 RGB-D reconstructed indoor scenes with dense per-point semantic and instance-level annotations spanning 18 object categories. The official data split includes 1,201 scans for training and 312 scans for validation, providing a comprehensive benchmark for evaluating indoor 3D scene understanding tasks such as segmentation and detection. For the ScanNet dataset (Dai et al., 2017), we follow the evaluation protocol of recent indoor 3D object detectors (Kolodiazhnyi et al., 2024; Wang et al., 2025; Shen et al., 2023), using mean IoU thresholds of 0.25 and 0.5 to compare our pseudo labels with human annotations across different object categories.

Implementation Details. We implemented CAT++ using PyTorch (Paszke et al., 2019) and employed standard Transformer layers for the CAT++ encoder. HIE has $L = 5$ stages, each consisting of intra- and inter-object blocks. The local block consists of one-layer normalization, an eight-head self-attention, and an MLP with two linear layers and ReLU nonlinearity. The global closely follows the local block but with batch-dimension self-attention. We employed two layers for the decoder, each with multiheaded self- and cross-attention and an MLP. Three two-layer MLPs with a hidden size $d = 1024$ achieve the final box regression. For optimization, we used Adam optimizer with a learning rate of 10^{-4}, a cosine annealing scheduler, and a weight decay of 0.05. CAT++ was trained on one RTX 3090 GPU for 1000 epochs, with a batch size $B = 12$.

To adapt the model to the sparser objects in nuScenes compared to KITTI, we adjusted the stage $L = 5$ in HIE to $L = 4$ and doubled up a batch size $B = 24$ to mitigate overfitting. The decoder remained consistent with two layers.

We empirically set the input sub-cloud size $N = 350$ ($N = 1024$ in KITTI) to accommodate the sparse objects (Liu et al., 2022a).

Regarding implementing the INR in CAT++, we have adopted the fundamental principles in the OccNet model (Mescheder et al., 2019), with some adjustments to better suit our objectives. Rather than employing PointNet as the encoder or integrating post-processing components as OccNet does, we utilize a streamlined fully connected neural network with three ResNet blocks (He et al., 2016) and condition it on the input with learned INR tokens using conditional batch normalization (De Vries et al., 2017; Dumoulin et al., 2016). The attention-conditioned INR network is trained for the final decision boundary of classifier learning.

Standard data augmentations, such as random shifting, scaling, and flipping, were applied in both cases. We filtered objects with at least 30 total points and 15 foreground points, as annotating hard samples with less than 30 LiDAR points is not feasible (Liu et al., 2022b, a; Qian et al., 2023). 500 frames of labeled data are randomly chosen to train our model CAT++. CAT++, once trained, can be used as a 3D automatic annotator. We use trained CAT++ to re-label the KITTI training set on the Car class. For 3D object detection, we employed PointRCNN in KITTI. Since the nuScenes evaluation metric includes all 10 classes, we retrain the human annotations for the remaining classes and substitute only the Car annotation with those generated by CAT++. We then employ a re-implemented version of the widely used PointPillars object detector and train it using this newly annotated dataset. We employ the training and evaluation protocols outlined in the OpenPCDET (Team, 2020) to train and assess PointRCNN and PointPillar models on KITTI and nuScenes datasets, respectively.

4.2 Main Results and Comparison with SOTAs

Quantitative Results on KITTI Set (Car). We first present the 3D annotation results on the *test* set of KITTI, by submitting our results of CAT++ (PointRCNN) to the official evaluation server. As demonstrated in the KITTI *test* set (Table 1), we compare our method with fully supervised detection methods, which all use fully labeled training data (i.e., 3,712 precisely labeled frames with 15,654 vehicle instances). Our method achieves comparable performance despite using only 500 labeled frames (i.e., 2,176 human-annotated samples). PointRCNN trained on CAT++-generated pseudo labels can attain 99.60% of the version of the original model fully supervised with human annotations. Moreover, we can observe that CAT++ (PointRCNN) performance is very close to that of PointRCNN trained with human labels in Moderate and Hard tasks. This results in a six-fold reduction in the need for human annotations (i.e., 15,654 vehicle instances) on the KITTI training split for the Car.

Furthermore, in addition to experiments with PointRCNN, we conducted consistent experiments using PointPillars trained on our pseudo 3D annotations. Specifically, PointPillars trained with CAT++ annotations achieved 95% (V.S. 91.89% from Mtrans-PointPillars) of the performance of the fully supervised human-annotated PointPillars model. These results confirm that CAT++ annotations consistently achieve minimal performance gaps compared to manual annotations across multiple detection architectures, underscoring the practical advantages of CAT++ annotations in real-world scenarios.

Compared to current SOTAs (Wei et al., 2021; Meng et al., 2021, 2020; Liu et al., 2022a, b; Paat et al., 2024), CAT++ sets a new state-of-the-art for performance, even while eliminating the need for 3D segmentation, point completion, and multimodal information combination, which are common in baseline models. It outperforms the latest MEDL-U by improving the 3D detection accuracy on Moderate, and Hard tasks, registering improvements of 0.15%, and 1.8%, respectively. The hard task, challenged by data sparsity, significantly benefits from CAT++'s unique capacities of inter-object communications and continuous representation from our modified INR.

Similar results have been observed on KITTI *val* split, as shown in Table 2. CAT++-trained PointRCNN reaches performance levels on par with a self-supervised version of PointRCNN on the validation split. Remarkably, PointRCNN trained with CAT++ surpasses current annotation methods applied on the KITTI validation set, even those leveraging LiDAR and RGB inputs for multimodal learning. CAT++-trained PointPillars demonstrated comparable performance to the fully supervised model, echoing the effectiveness of CAT++-trained PointRCNN. This demonstrates the strength of CAT++ in exploiting point cloud data for high-quality annotation, independent of additional RGB information.

In challenging tasks (Hard tasks in Table 1 and 2) marked by data sparsity, CAT++ excels due to its specialized features: inter-object communication through its HIE and continuous object representation via an attention-conditioned INR. These features enable CAT++ to capture local and global contexts, improving its performance in sparse and complex 3D point clouds. Specifically, the HIE focuses on understanding interactions between different objects in a scene, while INR offers a continuous, implicit 3D object representation. Together, they make CAT++ robust and accurate, particularly in generating 3D bounding boxes for hard-to-annotate samples. This integrated approach combines both discrete and continuous representations to overcome the challenges posed by data sparsity and irregularities.

Quantitative Results on KITTI Set (Pedestrian). To further demonstrate the efficacy and generalizability of our MMCAT multimodal framework, we conduct experiments

Table 1 Results of KITTI official *test* set (Car), compared to the fully supervised PointRCNN and other weakly supervised baselines.

Method	Modality	Full Supervision	$AP_{3D}(IoU = 0.7)$			$AP_{BEV}(IoU = 0.7)$		
			Easy	Moderate	Hard	Easy	Moderate	Hard
PointRCNN Shi et al. (2019)	LiDAR	✓	*86.96*	*75.64*	*70.70*	*92.13*	*87.39*	*82.72*
PointPillarsLang et al. (2019)	LiDAR	✓	*82.58*	*74.31*	*68.99*	*90.07*	*86.56*	*82.81*
MV3DChen et al. (2017)	LiDAR	✓	74.97	63.63	54.00	86.62	78.93	69.80
F-PointNetQi et al. (2018)	LiDAR	✓	82.19	69.79	60.59	91.17	84.67	74.77
AVODKu et al. (2018)	LiDAR	✓	83.07	71.76	65.73	90.99	84.82	79.62
SECONDYan et al. (2018)	LiDAR	✓	83.34	72.55	65.82	89.39	83.77	78.59
SegVoxelNetYi et al. (2020)	LiDAR	✓	86.04	76.13	70.76	91.62	86.37	83.04
Part-A^2Shi et al. (2020a)	LiDAR	✓	87.81	78.49	73.51	91.70	87.79	84.61
PV-RCNNShi et al. (2020b)	LiDAR	✓	90.25	81.43	76.82	94.98	90.65	86.14
Comparison with other Autolabelers								
WS3D (PointRCNN)(2020) Meng et al. (2020)	LiDAR	BEV Centroid	80.15	69.64	63.71	90.11	84.02	76.97
WS3D(2021) (PointRCNN) Meng et al. (2021)	LiDAR	BEV Centroid	80.99	70.59	64.23	90.96	84.93	77.96
FGR (PointRCNN)(2021) Wei et al. (2021)	LiDAR	2D Box	80.26	68.47	61.57	90.64	82.67	75.46
MAP-Gen(PointRCNN)(2022) Liu et al. (2022b)	LiDAR+RGB	2D Box	81.51	74.14	67.55	90.61	85.91	80.58
MTrans (PointRCNN)(2022) Liu et al. (2022a)	LiDAR+RGB	2D Box	83.42	75.07	68.26	91.42	85.96	78.82
MTrans (PointPillars)(2022) Liu et al. (2022a)	LiDAR+RGB	2D Box	77.65	67.48	62.38	-	-	-
MEDL-U(PointRCNN)(2024) Paat et al. (2024)	LiDAR+RGB	2D Box	85.49	75.96	69.12	**91.86**	86.68	79.44
CAT (PointRCNN)(2023) Qian et al. (2023)	LiDAR	2D Box	84.84	75.22	70.05	91.48	85.97	80.93
CAT++ (PointRCNN)(2023)	LiDAR	2D Box	**85.64**	**76.11**	**70.29**	92.03	**86.99**	**81.90**
CAT (PointPillars) Qian et al. (2023)	LiDAR	2D Box	77.60	67.66	63.10	84.10	80.22	76.13
CAT++ (PointPillars)	LiDAR	2D Box	**78.68**	**70.21**	**64.01**	**86.11**	**82.81**	**78.84**

Bold numbers indicate the best performance in each column (highest value)

Table 2 Results of KITTI *val* set (Car), compared to the fully supervised PointRCNN and other weakly supervised baselines

Method	Modality	Full Supervision	$AP_{3D}(IoU = 0.7)$		
			Easy	Moderate	Hard
PointRCNN Shi et al. (2019)	LiDAR	✓	88.88	78.63	77.38
PointPillarsLang et al. (2019)	LiDAR	✓	86.10	76.58	72.79
WS3D (PointRCNN)Meng et al. (2020)	LiDAR	BEV Centroid	84.04	75.10	73.29
WS3D(2021) (PointRCNN)Meng et al. (2021)	LiDAR	BEV Centroid	85.04	*75.94*	*74.38*
FGR (PointRCNN)Wei et al. (2021)	LiDAR	2D Box	*86.68*	73.55	67.91
MAP-Gen (PointRCNN) Liu et al. (2022b)	LiDAR+RGB	2D Box	87.87	77.98	76.18
MTrans (PointRCNN) Liu et al. (2022a)	LiDAR+RGB	2D Box	88.72	78.84	77.43
MTrans (PointPillars) Liu et al. (2022a)	LiDAR+RGB	2D Box	**86.69**	76.56	72.38
MEDL-U(PointRCNN)(2024) Paat et al. (2024)	LiDAR+RGB	2D Box	89.26	75.27	78.05
CAT (PointRCNN) Qian et al. (2023)	LiDAR	2D Box	89.19	79.02	77.74
CAT++ (PointRCNN)	LiDAR	2D Box	**89.33**	**79.33**	**78.21**
CAT (PointPillars) Qian et al. (2023)	LiDAR	2D Box	86.05	76.06	72.59
CAT++ (PointPillars)	LiDAR	2D Box	86.11	**76.58**	**72.95**

Bold numbers indicate the best performance in each column (highest value)

Table 3 Results of KITTI *Val* set (Pedestrian), compared to the fully supervised PointRCNN and other autolabelers

Method	Modality	Full Supervision	$AP_{3D}(IoU = 0.5)$			$AP_{BEV}(IoU = 0.5)$		
			Easy	Moderate	Hard	Easy	Moderate	Hard
PointRCNN Shi et al. (2019)	LiDAR	✓	63.70	69.43	58.13	68.89	63.54	57.63
PointPillarsLang et al. (2019)	LiDAR	✓	66.73	61.06	56.50	71.97	67.84	62.41
Part-A^2Shi et al. (2020a)	LiDAR	✓	70.73	64.13	57.45	-	-	-
STDYang et al. (2019)	LiDAR	✓	73.90	66.60	62.90	75.09	69.90	66.00
VoxelNetZhou and Tuzel (2018)	LiDAR	✓	-	-	-	70.76	62.73	55.05
Comparison with other Autolabeler								
WS3D (PointRCNN) Meng et al. (2021)	LiDAR	BEV Centroid	74.65	69.96	66.49	74.79	70.17	66.75
CAT (PointRCNN) Qian et al. (2023)	LiDAR	2D Box	75.15	70.06	67.09	74.79	71.27	66.75
CAT++ (PointRCNN)	LiDAR	2D Box	**75.55**	**70.92**	**68.42**	**75.05**	**72.05**	**67.82**
CAT (PointPillars) Qian et al. (2023)	LiDAR	2D Box	66.70	61.10	57.10	71.96	67.80	62.75
CAT++ (PointPillars)	LiDAR	2D Box	**66.75**	**61.10**	**58.02**	**71.99**	**68.23**	**63.60**

Bold numbers indicate the best performance in each column (highest value)

on the Pedestrian class of KITTI. It's important to note that, for a predicted 3D box to be considered acceptable in the Pedestrian category, it must achieve an IoU greater than 0.5 with the ground truth, contrasting with a higher threshold of 0.7 sets for the Vehicle category. Of the 3,712 training scenes in KITTI, only 951 scenes contain pedestrian labels. Considering the small number of training samples, we randomly choose 23% (515) of 2,257 samples. Compared to previously fully supervised PointRCNN, which leverages all 951 exhaustively annotated scenes with 2,257 pedestrian samples, we use far fewer and weaker 2D supervisions. As evidenced in Table 3, CAT++ indicates auspicious results in the Pedestrian class, surpassing the majority of existing fully supervised models while utilizing only 13.5% labeled data under 2D weak supervision. CAT++ surpasses the performance of CAT, particularly in moderate and hard scenarios. This is evidenced by an improvement of 1.33% in AP_{3D} on Hard Task from CAT++-PointRCNN and 1% from CAT++-PointPillars, highlighting CAT++'s enhanced capability to accurately detect and annotate 3D objects under challenging conditions. This verifies CAT++'s good generalizability across diverse object classes and efficiency in handling smaller objects, such as Pedestrians.

Quantitative Results on nuScenes Set (Car). The CAT++ model is further evaluated on the nuScenes dataset, facilitating a comprehensive assessment of its efficiency. As depicted in Table 4, CAT++ demonstrates remarkable efficiency by requiring only 500 annotated frames to attain performance levels comparable to those achieved by the original PointPillars, which was trained with 28,130 human-annotated frames. Specifically, the NDS of PointPillars trained with CAT++ reaches 90.89% of the NDS achieved by the original model. CAT++ closely matches and even outperforms the

Table 4 Results of nuScenes *val* set (Car). We show the nuScenes detection score (NDS) and mean Average Precision (mAP).

Method	Modality	Fully Supervision	mAP	NDS
PointPillarsLang et al. (2019)	LiDAR	✓	41.26	55.54
MTrans Liu et al. (2022a)	LiDAR+RGB	2D Box	33.74	50.48
CAT Qian et al. (2023)	LiDAR	2D Box	33.70	50.45
CAT++	LiDAR	2D Box	**34.35**	**51.30**

Bold numbers indicate the best performance in each column (highest value)

Table 5 CAT++ generated 3D boxes in comparison with nuScenes human annotations. We show mIoU, Recall with an IoU threshold of 0.7, and Recall with a location error (LE) threshold of 0.5.

	mIoU↑	Recall (IoU = 0.7)	Recall (LE = 0.5)
CAT (train)	72.69	73.31	87.51
CAT++ (train)	**73.01**	**73.68**	**87.78**
CAT (val)	70.80	70.92	87.34
CAT++ (val)	**72.59**	**73.22**	**87.54**

Bold numbers indicate the best performance in each column (highest value)

Table 6 CAT++ generated 3D boxes in comparison with ScanNet human annotations. We show mIoU, Recall with an IoU threshold of 0.5, and Recall with an IoU threshold of 0.25

	mIoU↑	Recall (IoU = 0.5)	Recall (IoU = 0.25)
CAT (train)	72.65	81.42	87.51
CAT++ (train)	**75.23**	**84.91**	**89.27**
CAT (val)	70.33	81.02	87.94
CAT++ (val)	**75.12**	**84.32**	**90.03**

Bold numbers indicate the best performance in each column (highest value)

performance of the latest 3D annotator, MTrans, while relying solely on point cloud data and eliminating the need for multimodal training using RGB information. Remarkably, it achieves this with 50 times less human annotation effort on the nuScenes dataset for the Car category.

To evaluate the influence of class-specific factors, such as pedestrians, we directly compared the CAT++ Car annotations with the manual ground truth labels, with results presented in Table 5. Evaluation metrics include mIoU, Recall with an IoU threshold of 0.7, and Recall with an LE threshold of 0.5. The 3D bounding boxes generated by CAT++ across the entire nuScenes training split achieve a mean IoU of 73.01% and a recall rate of 73.68%. Given the greater sparsity and larger size of the nuScenes dataset compared to KITTI (118k points vs. 30k points), the INR in CAT++ demonstrates its value by effectively learning a more generalized representation of car surfaces. This learning enables the model to comprehensively relabel the whole training set of nuScenes, effectively handling the 27,630 frames of unseen data, while utilizing 500 frames for training. This suggests that CAT++ could serve as a 3D automatic annotator for the nuScenes dataset, significantly reducing the necessity for time-consuming and labor-intensive human annotation during data preparation, thus underlining the generalization capability of our framework.

Quantitative Results on ScanNet. In addition to the 3D outdoor datasets, KITTI and the sparser nuScenes, we further evaluated the annotation capability of CAT++ on the denser indoor object dataset ScanNet. As shown in Table 6, we adopted evaluation metrics including mean IoU, recall at an IoU threshold of 0.5, and recall at an IoU threshold of 0.25, following standard indoor detection protocols. On the ScanNet training set, CAT++-generated annotations achieved a mean IoU of 75.23% and a recall (IoU = 0.5) of 84.91%, with consistent performance observed on the validation set. These results are comparable to those achieved on KITTI (IoU = 75.76%) and notably better than the performance on the sparser nuScenes dataset. This further confirms the generalizability of CAT++ to indoor environments, demonstrating its flexibility beyond outdoor settings.

While CAT++ shows promising performance on indoor datasets, we respectfully emphasize that its core design and primary use case target outdoor LiDAR-based datasets, particularly those relevant to autonomous driving scenarios. In such contexts, where manual annotation of large-scale 3D point clouds is time-consuming and labor-intensive, CAT++ provides an efficient alternative, significantly reducing annotation cost while preserving high annotation quality.

CAT++ exhibits strong generalization and robust performance across diverse 3D environments, including outdoor datasets such as KITTI and nuScenes, as well as the indoor dataset ScanNet. However, it shows more pronounced improvements, up to a 2% increase in 3D AP, when compared to other methods on the KITTI hard tasks. We surmise that the nuScenes dataset's reliance on 32-line LiDAR, which yields about 30k LiDAR points per frame (approximately one-sixth of the points provided by the KITTI dataset), could limit the potential enhancements achievable through feature relations from the HIE and INR learning and refinement.

Table 7 Annotation Speed and Training Time

Methods	Inference (s)	Training (h)	GPU Memory (GB)
Human Annotation Song et al. (2015)	114	–	–
WS3D Meng et al. (2021)	2.5	10.0	–
MTrans Liu et al. (2022a)	0.04	6.5	10
MEDL-U Paat et al. (2024)	0.04	7	10.5
CAT Qian et al. (2023)	0.02	6.5	8
CAT++ (ours)	0.02	6.5	8.3

Annotation Speed. An analysis of the annotation speed based on our model's inference time with detailed deficiency is included in Table 7. In particular, CAT++ was able to relabel the KITTI train set, which contains 3,712 frames with 15,654 vehicle instances, in approximately 3 minutes. CAT++ annotates at 0.02 seconds per instance, in contrast, when using human annotations, it takes around 114 seconds per instance or 30 seconds with the assistance of a 3D object detector, as reported in prior studies (Huang et al., 2019; Song et al., 2015; Meng et al., 2021). CAT++ demonstrates a significant speed enhancement, which translates to a reduction of over 1500× in time required for precise annotations traditionally. Moreover, our model is 600× more efficient than WS3D (Meng et al., 2021) that takes 2.5 seconds to label one vehicle instance. More details are in the Table 7. Notably, adding our attention-conditioned INR module has not resulted in a discernible latency increase, with a negligible difference of just 0.001 seconds per instance compared to CAT. This trend consistently holds on the nuScenes dataset, maintaining an annotation speed of 0.02 seconds per car with no marked difference from CAT.

Additionally, CAT++ uses approximately 8.3 GB of GPU memory during training on KITTI (batch size 4), which is only a small increase (+6%) over the baseline CAT (without occupancy). Compared to other multi-task or dense 3D prediction methods, our memory footprint is on par or lower - for example, it is about 15% less than the memory required by the multimodal MTrans model (which processes images and dense point clouds), as that method must maintain image feature maps and additional point generation features in memory. We attribute CAT++'s modest memory usage to our design choice of an implicit occupancy representation (INR) instead of a large 3D voxel grid - we do not store a full 3D volume, but rather predict occupancy at queried points on-the-fly, which is memory-efficient. In inference mode, CAT++ runs with 4 GB of GPU memory, which is well within the limits of typical deployment scenarios (and nearly identical to the baseline annotator).

Qualitative Results. Fig. 3 presents visualized examples of pseudo labels on the Car category generated by CAT++ on the KITTI Car validation split, alongside ground truth boxes in blue on hard samples with sparse points. Even for instances where the subjects are heavily truncated, occluded, or significantly distanced, CAT++ can generate high-quality bounding boxes compared to existing SOTAs. This is achieved by focusing on easy samples (those with less occlusion) and utilizing inter-object relations from the HIE and INR to refine the initially incomplete representations of hard samples.

In Fig. 4, we also present visual examples of CAT++'s performance relabeling the KITTI training set for the Pedestrian category. The first two rows (dense points) illustrate the model's precision in generating 3D bounding boxes for clear, non-occluded pedestrians at a moderate distance characterized by dense point clouds. Our model effectively segments the foreground and background, efficiently distinguishing background points to generate 3D bounding boxes for pedestrians. As shown in the last row (hard samples), CAT++ also accurately identifies heavily occluded or very far pedestrian samples, significantly sparser than vehicles. This demonstrates CAT++'s capability to process vehicles and adapt to other categories within autonomous driving contexts, even with minimal and more readily available supervision, particularly for challenging samples.

These visualizations underscore the effectiveness of CAT++ and further validate the feasibility of point-wise discrete point representation from the HIE and INR for 3D automatic annotations. This blend of multi-level information helps CAT++ in delivering reliable and accurate 3D annotations, even for complex scenes.

4.3 Ablation Studies

4.3.1 Effectiveness of Individual Components within CAT++

Various experiments are conducted to validate the efficacy of each component within the CAT++ model. These include evaluating the intra-object encoder (local block), which handles local information; the inter-object encoder (global block), responsible for global information; the decoder; the hierarchical and interleaved encoding techniques (HIE); and the INR learning. We also investigate the optimal position embedding strategies for CAT++ training. The effectiveness of these elements is assessed by comparing the 3D pseudo labels generated by our model with human annotations. The

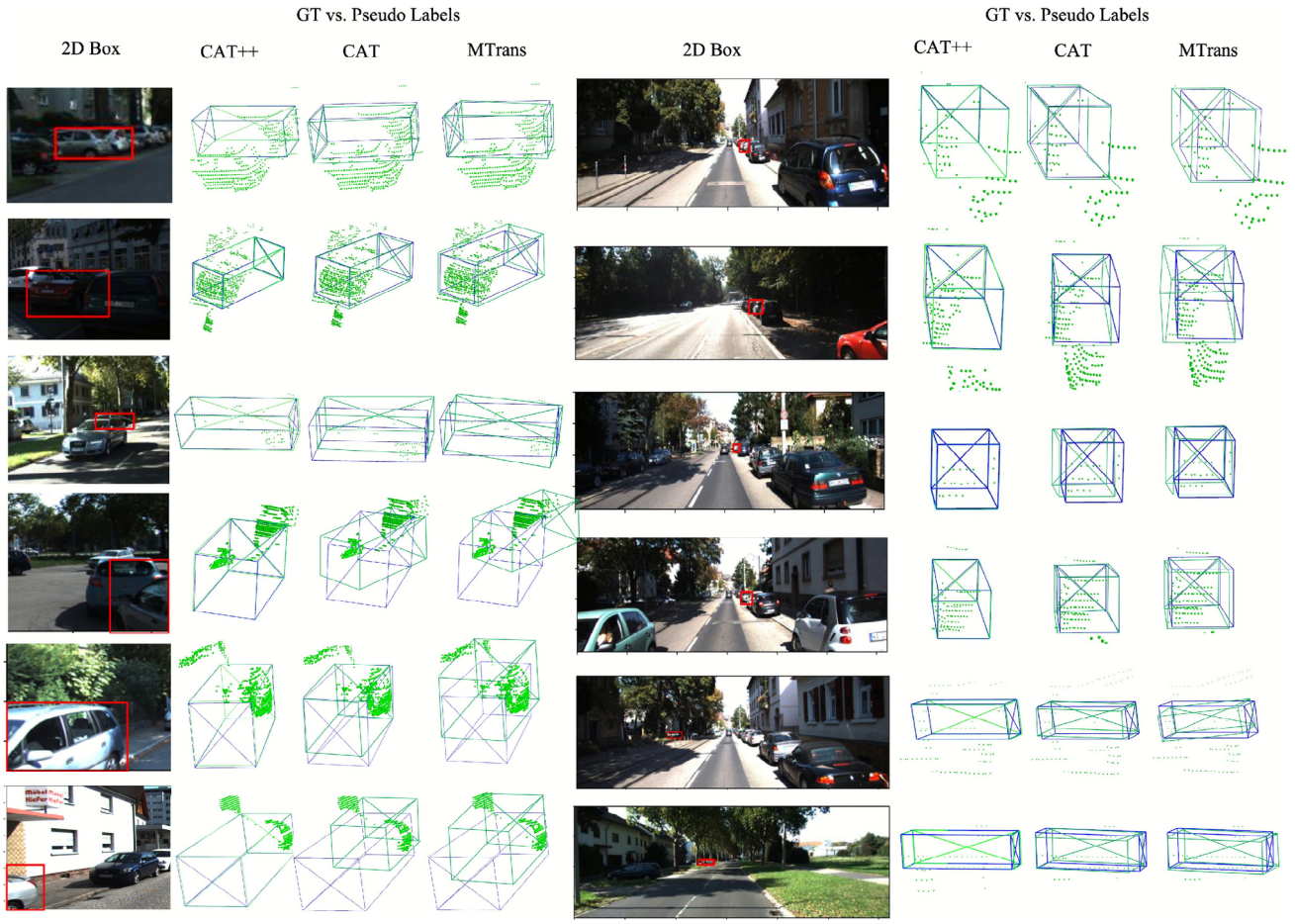

Fig. 3 Qualitative results (Car) using CAT++ on the KITTI *train* split. For the sake of visualization, CAT++ operates solely on point cloud data, and image information is included in our presentation. CAT++ demonstrates considerable robustness and effectiveness when generating 3D bounding boxes, even for hard samples. This includes samples where the objects of interest are heavily occluded (as observed in the first three rows), heavily truncated (left part of the final three rows), or located at considerable distances from the sensor (on the right). In these challenging situations, CAT++ can generate amodal boxes (in green) that capture the full extent of cars. The ground truth bounding boxes are marked blue

metrics used for this comparison include mIoU, recall with an IoU threshold of 0.7, mAP, and mAP_{R40}, specifically for the Car category. The mIoU measures the average overlap accuracy between the generated pseudo labels and human annotations. Recall (IoU=0.7) evaluates the percentage of ground-truth boxes correctly detected by pseudo labels at an IoU threshold of 0.7. The mAP represents the average detection precision across multiple IoU thresholds, reflecting overall detection accuracy. The mAP_{R40} calculates the mean average precision using 40 uniformly sampled recall points for more precise performance evaluation. These comprehensive evaluations allow us to ascertain the contribution of each component in our model and its collective impact on performance.

In Table 8, we regard the standard Transformer encoder as a baseline. Model A (local block): The local encoder, which is responsible for extracting intra-object interactions among points, is demonstrated in this model. Model B (local + global blocks): The incorporation of global block results in performance boosts of 5.69%, 10.71%, 26.89%, and 26.65% on mIoU, Recall, mAP, and mAP_{R40}, respectively. This indicates that the global successfully models inter-object relations and enhances the interaction among samples, which is particularly beneficial for hard samples. Model C (local + global blocks + decoder): Further improvement in mIoU by 2.31% is observed with the decoder's inclusion, signifying that the decoder refines the object-level representations for more accurate box regression. Model E (local + global blocks + decoder + pos enc.): This model achieves an impressive 72.78% Recall and 87.04% mAP with an MLP embedding approach. This supports the importance of positional encoding in 3D coordinates to adapt to the 3D structure. Notably, the more straightforward, parameterizable, and trainable

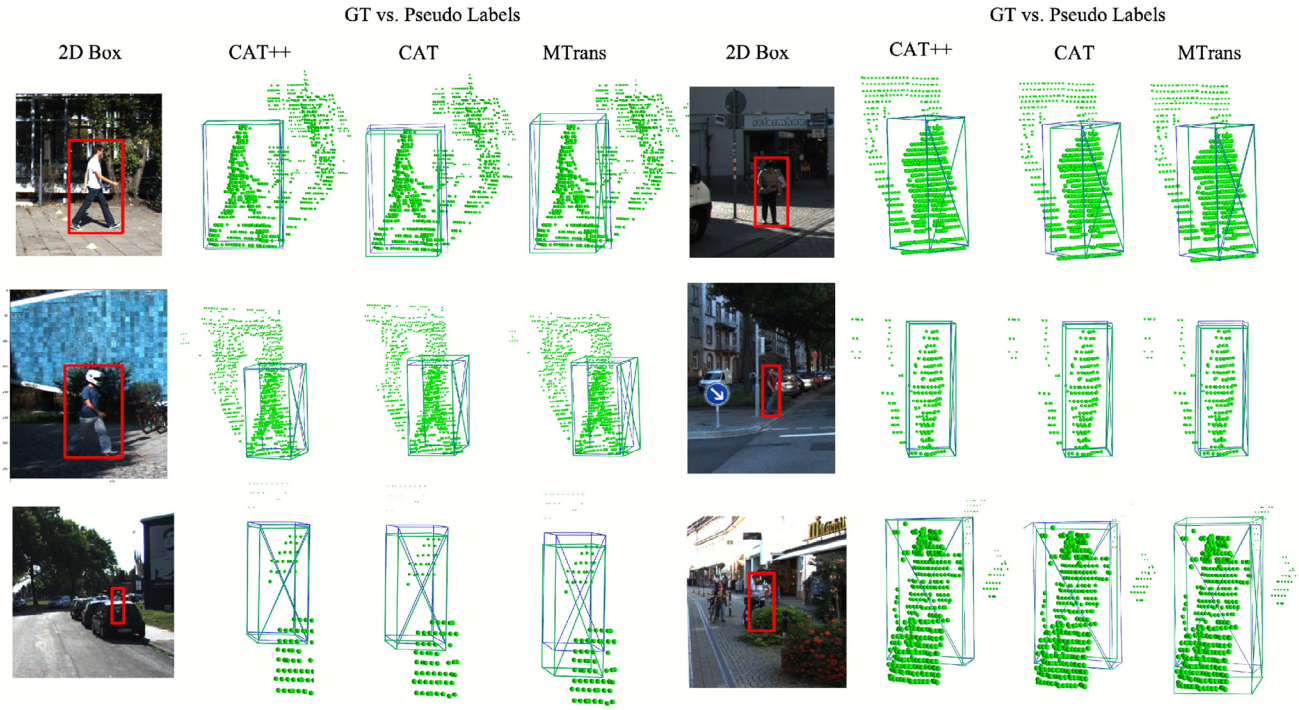

Fig. 4 Qualitative results (Pedestrian) using CAT++ on the KITTI *train* set. The first two rows display the model's precision (in green) with clear, dense point cloud pedestrians. Our model effectively segments foreground and background to accurately generate 3D pedestrian bounding boxes. Furthermore, the last rows reveal their effectiveness in identifying sparse, heavily occluded, or distant pedestrians (hard samples). The ground truth bounding boxes are marked blue

positional embedding of MLP outperforms other manually designed position encoding schemes in Model D in our case.

Model F (HIE): The local and global blocks in CAT++ are operated in a hierarchical-interleaved manner, resulting in performance improvements of 1.56% and 0.52% on Recall and mIoU, respectively. This validates the hierarchical-interleaved encoding's capability to refine local and global features by alternately focusing on these two dimensions of attention. Combining the advantages of all the above components and using the attention-conditioned INR to implicitly represent 3D, we achieve an impressive final performance of 75.76% and 75.98% on mIoU and Recall, respectively.

Furthermore, we assess the effectiveness of box and INR tokens in CAT++. Box tokens, 7 in total, represent the seven elements of a box (x, y, z location, l, w, h size, and yaw orientation). The omission of either box tokens or the INR token within the CAT++ encoder leads to a substantial decline in performance, specifically a drop of over 3.6% in mIoU without box tokens. This considerable degradation in performance illustrates box tokens' crucial role in refining object-level features. The INR token plays a crucial role in learning continuous representations. It effectively bridges the gap between point tokens and condition tokens, enhancing their collective utility. This indicates that these tokens are pivotal in enabling CAT++ to delineate 3D object boundaries more accurately and effectively.

4.3.2 Balancing 3D Box Regression and INR Tasks in Multi-Task Learning

Our multi-task design inherently allows CAT++ to automatically balance the 3D box regression and the occupancy INR tasks. Specifically, both tasks share a common backbone feature encoder, which promotes mutual reinforcement rather than competition. Empirically, we found that CAT++ dynamically adjusts the emphasis between these tasks based on input data characteristics, particularly the point cloud sparsity.

Gradient Analysis on Difficulty Levels: To further substantiate this dynamic balancing behavior, we performed a detailed gradient analysis during training. Table 9 presents the averaged $L2$ norm of the gradients back-propagated from the occupancy INR task across different difficulty levels on the KITTI training set. The results reveal that when the model encounters extremely sparse or challenging samples (e.g., distant or occluded objects with $\leq$ 100 foreground points), the gradient magnitude associated with the INR task significantly increases. This suggests that CAT++ naturally prioritizes INR learning under sparse conditions, effectively enriching the deficient discrete point features.

Behavior under Dense Point Clouds: Conversely, when processing dense point clouds (e.g., indoor scans from Scan-

Table 8 Ablation Results on KITTI *val* Split (Car)

	local block	global block	decoder	pos. enc.	HIE	INR	Metric			
							mIoU	Recall	mAP	mAP_{R40}
Model A	✓						65.01	54.50	56.31	59.24
Model B	✓	✓					70.70	65.21	83.20	85.89
Model C	✓	✓	✓				72.88	70.84	85.83	87.72
Model D	✓	✓	✓	Sine			72.13	69.66	85.21	87.34
Model E	✓	✓	✓	MLP			73.71	72.78	87.04	89.80
Model F	✓	✓	✓	MLP	✓		74.23	74.34	87.52	89.99
Ours (500)	✓	✓	✓	MLP	✓	✓	**75.76**	**75.98**	**87.61**	**90.10**
Ours (all)	✓	✓	✓	MLP	✓	✓	80.12	85.21	92.42	94.68

Bold numbers indicate the best performance in each column (highest value)

Table 9 Gradient magnitude (L2 norm) of the occupancy INR task across difficulty levels in the KITTI training set (Vehicle category). The increase in gradient magnitude for hard samples suggests stronger reliance on INR when foreground points are sparse

Difficulty Level	Avg. Foreground Points	INR Gradient (L2 Norm)
Easy	> 300	1.23
Moderate	100 - 300	2.01
Hard	≤ 100	**3.74**

Bold numbers indicate the best performance in each column (highest value)

Table 10 Ablation study on occupancy loss weight λ_{occ}, showing its effect on mIoU and 3D AP on the KITTI dataset

λ_{occ}	mIoU (%)	mAP (%)
0.0	73.71	87.04
0.5	74.23	87.22
1.0	**75.76**	**87.61**
2.0	75.20	86.76

Bold numbers indicate the best performance in each column (highest value)

Net), the box regression task gradients dominate while the INR gradients diminish, indicating that the model adapts its focus toward direct point-based features where occupancy inference becomes less critical.

Ablation Study on Occupancy Loss Weight λ_{occ}: To validate the robustness of the automatic balancing mechanism, we conducted an ablation study varying the occupancy loss weight λ_{occ} across different values ($\lambda_{occ} \in 0.0, 0.5, 1.0, 2.0$) on the KITTI dataset (see Table 10). We evaluated both the mIoU and the mAP.

The results demonstrate that CAT++ maintains stable 3D annotation performance across different λ_{occ} values, with mAP consistently remaining within a narrow range (maximum variation $< 1\%$). Meanwhile, the mIoU progressively improves with increasing λ_{occ} up to 1.0, achieving the best overall performance at $\lambda_{occ} = 1.0$. These findings highlight that incorporating the occupancy task enhances 3D feature quality without compromising annotation accuracy. Notably, an excessively large λ_{occ} (e.g., 2.0) slightly degrades the mAP, underscoring the importance of balanced task contributions.

Robustness to Sparse Point Clouds: The INR component in CAT++ was specifically designed to mitigate challenges posed by sparse or incomplete point clouds. On the nuScenes dataset (approximately $10\times$ sparser than KITTI), CAT++ maintained competitive mAP with only modest performance degradation (2%). Qualitative results (Figs. 3 and 4) further show that CAT++ can reliably reconstruct complete object shapes even from as few as 20 LiDAR points per object. This validates the capability of implicit neural representations to substantially enrich sparse data and sustain robust annotation quality under challenging conditions.

CAT++ achieves effective automatic balancing between 3D box regression and INR occupancy prediction tasks. Explicit gradient analysis and ablation studies demonstrate that CAT++ dynamically prioritizes INR learning for sparse samples and adapts to dense scenes when necessary. These capabilities, together with evaluations on both outdoor and indoor datasets, clearly validate the robustness and general applicability of CAT++ as a reliable 3D automatic annotation system.

4.4 Limitation and Future Work

Despite the general accuracy of our box generation technique, CAT++ experiences difficulties under certain challenging conditions. This section highlights these challenges and outlines future directions to enhance the framework's performance.

Challenges with small objects. 1) For hard samples with sparse foreground points of fewer than 15, our model struggles to accurately model these objects' data distribution and overall appearance. As detailed in the implementation section of our manuscript, we excluded these samples that were too sparse (< 15 points) for training CAT++. This was based on the observation that the model struggles to accurately capture the data distributions and appearances of objects with such sparse data, leading to unreliable results. For instance,

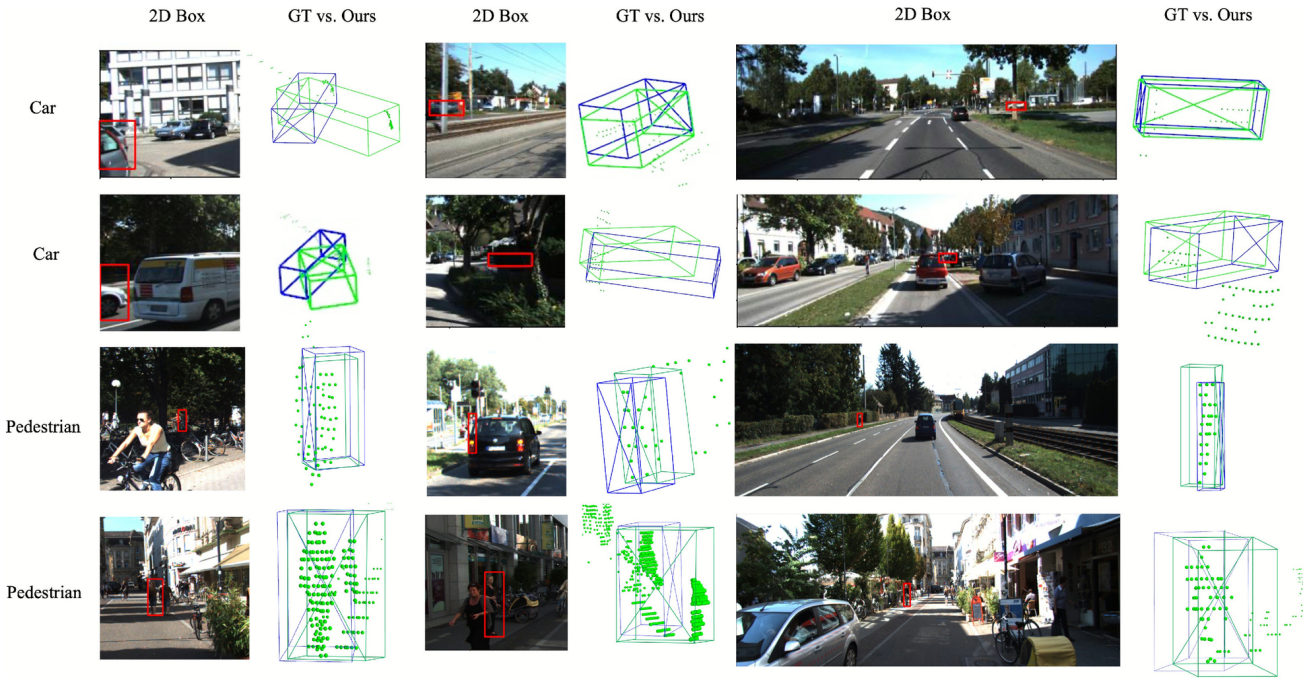

Fig. 5 Failure cases of 3D car and pedestrian annotations on the KITTI *val* set, highlighting the significant impact of sparse foreground point clouds (fewer than 15 points) on the accuracy of vehicle location, yaw, and pedestrian dimensions in our 3D box predictions (indicated in green). The ground truth bounding boxes are marked blue

the first two rows in Fig. 5 demonstrate extremely challenging samples, where our model incorrectly interprets partial vehicle segments as entire objects. Specifically, the first four columns of these rows show how our model inaccurately positions portions of the point clouds from the vehicle's front or rear as the central focus of the entire point cloud. This error significantly impacts the accuracy of location and yaw in our 3D box predictions (depicted in green). Moreover, the fifth and sixth columns emphasize that sparse foreground point clouds, particularly those from distant instances, lead to pronounced location inaccuracies. This misidentification significantly impacts detection performance, as detailed alignment is critical in 3D object detection. 2) Annotating pedestrians presents even greater challenges, frequently leading to location and dimension errors. The final two rows of Fig. 5 showcase examples where CAT++ confounds the foreground and background in pedestrian samples, mistakenly incorporating elements like nearby plants or walls into the pedestrian annotation. Furthermore, as depicted in the last row, pedestrian samples near each other often result in merged 3D box annotations. This aggregation complicates the accurate detection of individuals, exacerbating the difficulty of distinguishing separate entities.

Future work. To mitigate these limitations and refine CAT++'s capabilities, we plan to continue our work in the following directions. 1) Enhanced Contextual Information: Integrating more comprehensive contextual cues is essential. We aim to incorporate additional supervisory signals, such as visual camera feeds and object tracking data, to improve the representation of 3D point clouds, especially for extremely challenging samples. 2) Efficient Multimodal Learning Strategies: Exploring multimodal learning frameworks could unlock new potentials in point cloud processing with enhanced contextual information from vision cues. By adopting text-image, text-point, and image-point integration frameworks, we can develop more efficient frameworks that leverage diverse data types to enhance 3D box prediction accuracy.

5 Conclusion

This paper has developed Enhanced Context-Aware Transformer CAT++, an end-to-end 3D automatic labeling system that significantly alleviates human annotation effort with minimal labeled data input. The model leverages a Hierarchical-Interleaved Encoder (HIE) and an attention-conditioned Implicit Neural Representation (INR) network, integrated within a multi-task learning framework. This synthesis fosters a nuanced understanding of objects at multiple scales. The HIE alternates between intra- and inter-object blocks, providing a robust and hierarchical 3D representation. Simultaneously, the INR facilitates a continuous approximation of objects' shapes, enhancing the model's

ability to tackle hard samples. Our validation of CAT++ on the KITTI and nuScenes datasets evidences its superiority over existing benchmarks while simplifying 3D annotation processing.

Lastly, the new approach introduces challenges, especially in annotating smaller point cloud objects. Our future research will aim to overcome these challenges, drawing upon the meticulous design and refinement strategies employed in existing 3D automatic annotation models.

Data Availability The datasets generated in this study are available from the KITTI website (Geiger et al., 2012), the nuScenes website (Caesar et al., 2020), and the indoor ScanNet Dataset (Dai et al., 2017).

References

Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11621–11631.

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1907–1915.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., & Niessner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., & Courville, A.C. (2017). Modulating early visual processing by language. Advances in Neural Information Processing Systems *30*.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., & Courville, A. (2016). Adversarially learned inference. arXiv preprint arXiv:1606.00704.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision,88*, 303–338.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR).

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., & Hu, S.-M. (2021). Pct: Point cloud transformer. *Computational Visual Media,7*(2), 187–199.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Hou, Z., Yu, B., & Tao, D. (2022). Batchformer: Learning to explore sample relationships for robust representation learning. In: CVPR.

Hou, Z., Yu, B., Wang, C., Zhan, Y., & Tao, D. (2022). Batchformerv2: Exploring sample relationships for dense representation learning. arXiv preprint arXiv:2204.01254.

Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., & Yang, R. (2019). The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence,42*(10), 2702–2719.

Kolodiazhnyi, M., Vorontsova, A., Skripkin, M., Rukhovich, D., & Konushin, A. (2024). Unidet3d: Multi-dataset indoor 3d object detection. arXiv preprint arXiv:2409.04234.

Ku, J., Mozifian, M., Lee, J., Harakeh, A., & Waslander, S.L. (2018). Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1–8. IEEE.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., & Jia, J. (2022). Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8500–8509.

Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12697–12705.

Liu, C., Qian, X., Huang, B., Qi, X., Lam, E., Tan, S.-C., & Wong, N. (2022). Multimodal transformer for automatic 3d annotation and object detection. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII, pp. 657–673. Springer.

Liu, C., Qian, X., Qi, X., Lam, E.Y., Tan, S.-C., & Wong, N. (2022). Map-gen: An automated 3d-box annotation flow with multimodal attention point generator. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 1148–1155. IEEE

Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., & Han, S. (2022). Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. arXiv preprint arXiv:2205.13542.

Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., & Bai, X. (2020). Tanet: Robust 3d object detection from point clouds with triple attention. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11677–11684.

Maturana, D., & Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928 https://doi.org/10.1109/IROS.2015.7353481.

McCraith, R., Insafutdinov, E., Neumann, L., & Vedaldi, A. (2021). Lifting 2d object locations to 3d by discounting lidar outliers across objects and views. arXiv preprint arXiv:2109.07945.

Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., & Van Gool, L. (2021). Towards a weakly supervised framework for 3d point cloud object detection and annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., & Van Gool, L. (2021). Towards a weakly supervised framework for 3d point cloud object detection and annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., & Dai, D. (2020). Weakly supervised 3d object detection from lidar point cloud. In: European Conference on Computer Vision, pp. 515–531. Springer.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4460–4470.

Misra, I., Girdhar, R., Joulin, A. (2021). An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2906–2917.

Niemeyer, M., Mescheder, L., Oechsle, M., & Geiger, A. (2020). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3504–3515.

Paat, H., Lian, Q., Yao, W., & Zhang, T. (2024). Medl-u: Uncertainty-aware 3d automatic annotation based on evidential deep learning. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 13976–13982. IEEE.

Pan, X., Xia, Z., Song, S., Li, L.E., & Huang, G. (2021). 3d object detection with pointformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7463–7472.

Park, C., Jeong, Y., Cho, M., & Park, J. (2022). Fast point transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16949–16958.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems *32*.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., & Geiger, A. (2020). Convolutional occupancy networks. In: European Conference on Computer Vision, pp. 523–540. Springer.

Qi, C.R., Liu, W., Wu, C., Su, H., & Guibas, L.J. (2018). Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 918–927.

Qi, C. R., Su, H., Mo, K., & Guibas, L.J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. (pp. 652–660).

Qi, C.R., Yi, L., Su, H., & Guibas, L.J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413.

Qian, X., Liu, C., Qi, X., Tan, S.-C., Lam, E., & Wong, N. (2023). Context-Aware Transformer for 3D Point Cloud Automatic Annotation.

Qin, Z., Wang, J., & Lu, Y. (2020). Weakly supervised 3d object detection from point clouds. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 4144–4152.

Shen, Y., Geng, Z., Yuan, Y., Lin, Y., Liu, Z., Wang, C., Hu, H., Zheng, N., & Guo, B. (2023). V-detr: Detr with vertex relative position encoding for 3d object detection. arXiv preprint arXiv:2308.04409.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10529–10538.

Shi, S., Wang, X., & Li, H. (2019). Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 770–779.

Shi, S., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems,33*, 7462–7473.

Song, S., Lichtenberg, S.P., & Xiao, J. (2015). Sun rgb-d:A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., & Anguelov, D. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2446–2454.

Team, O.D. (2020). OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet.

Wang, C., Yang, W., Liu, X., & Zhang, T. (2025). State space model meets transformer: A new paradigm for 3d object detection. arXiv preprint arXiv:2503.14493.

Wei, Y., Su, S., Lu, J., & Zhou, J. (2021). Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 4348–4354. IEEE

Wilson, B., Kira, Z., & Hays, J. (2020). 3d for free: Crossmodal transfer learning using hd maps. arXiv preprint arXiv:2008.10592.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920.

Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022). Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems,35*, 33330–33342.

Xie, J., Zheng, Z., Gao, R., Wang, W., Zhu, S.-C., & Wu, Y.N. (2018). Learning descriptor networks for 3d shape synthesis and analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8629–8638.

Xu, M., Ding, R., Zhao, H., & Qi, X. (2021). Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3173–3182.

Yan, S., Yang, Z., Li, H., Guan, L., Kang, H., Hua, G., & Huang, Q. (2022). Implicit autoencoder for point cloud self-supervised representation learning. arXiv preprint arXiv:2201.00785.

Yang, Y., Feng, C., Shen, Y., & Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 206–215.

Yang, Z., Sun, Y., Liu, S., Shen, X., & Jia, J. (2019). Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1951–1960.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors,18*(10), 3337.

Ye, T., Yan, X., Wang, S., Li, Y., & Zhou, F. (2022). An efficient 3-d point cloud place recognition approach based on feature point extraction and transformer. *IEEE Transactions on Instrumentation and Measurement,71*, 1–9.

Yi, H., Shi, S., Ding, M., Sun, J., Xu, K., Zhou, H., Wang, Z., Li, S., & Wang, G. (2020). Segvoxelnet: Exploring semantic context and depth-aware features for 3d vehicle detection from point cloud. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2274–2280. IEEE.

Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., & Zhou, J. (2021). Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12498–12507.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., & Lu, J. (2022). Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19313–19322

Zakharov, S., Kehl, W., Bhargava, A., & Gaidon, A. (2020). Autolabeling 3d objects with differentiable rendering of sdf shape priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12224–12233

Zhang, W., & Xiao, C. (2019). Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12436–12445.

Zhao, H., Jiang, L., Jia, J., Torr, P.H.S., & Koltun, V. (2021). Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16259–16268.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12993–13000

Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4490–4499.