HyperDet3D: Learning a Scene-conditioned 3D Object Detector

Yu Zheng[1,3] Yueqi Duan[2†] Jiwen Lu[1,3] Jie Zhou[1,3] Qi Tian [4]
[1]Department of Automation, Tsinghua University
[2]Department of Electronic Engineering, Tsinghua University
[3]Beijing National Research Center for Information Science and Technology
[4]Huawei Cloud & AI, China
zhengyu19@mails.tsinghua.edu.cn, {duanyueqi,lujiwen,jzhou}@tsinghua.edu.cn, tian.qi1@huawei.com

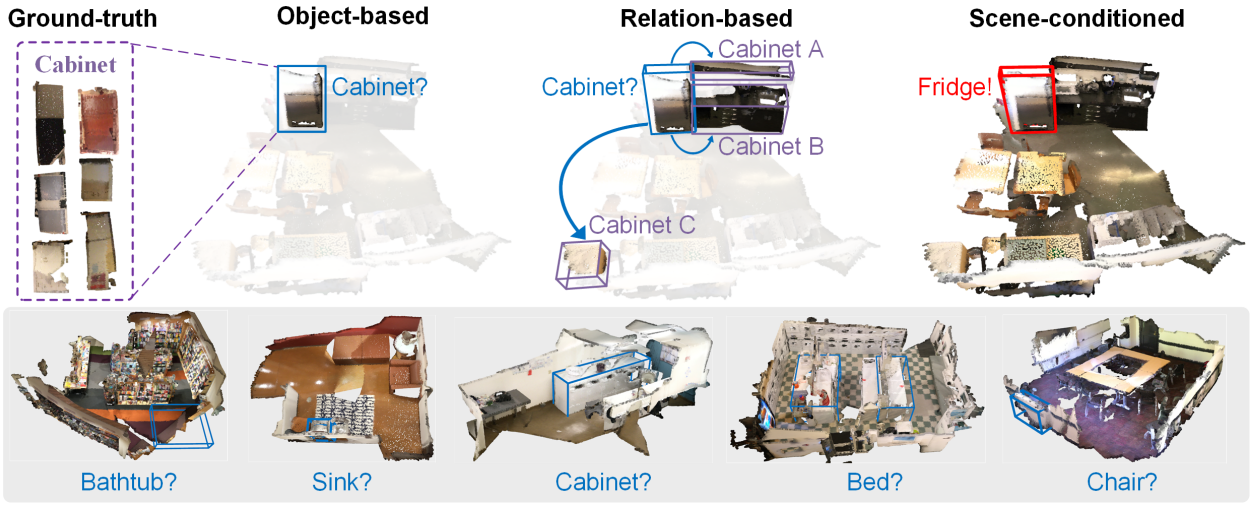

Figure 1. Exemplified predictions that highlight the importance of scene-conditioned knowledge. In the upper example, by observing the detection candidate in the object level, we can easily recognize it as *cabinet* by comparing it with groundtruth cabinets, or relating it with other surrounding cabinets. However, conditioned on the prior knowledge that the object candidate lies in a kitchen-like scene, we may infer that it is a *fridge*. We also illustrated 5 wrong detections which go against the scene-conditioned knowledge in the lower half, which are *bathtub* in a library, *sink* in an office, *cabinet* or *bed* in a laundry room, and *chair* embedded in the wall of a meeting room. Note that point clouds are all colored only for easy illustration and not utilized in our method. (*Best viewed in color.*)

Abstract

A bathtub in a library, a sink in an office, a bed in a laundry room – the counter-intuition suggests that scene provides important prior knowledge for 3D object detection, which instructs to eliminate the ambiguous detection of similar objects. In this paper, we propose HyperDet3D to explore scene-conditioned prior knowledge for 3D object detection. Existing methods strive for better representation of local elements and their relations without scene-conditioned knowledge, which may cause ambiguity merely based on the understanding of individual points and object candidates. Instead, HyperDet3D simultaneously learns scene-agnostic embeddings and scene-specific knowledge through scene-conditioned hypernetworks. More specifically, our HyperDet3D not only explores the sharable abstracts from various 3D scenes, but also adapts the detector to the given scene at test time. We propose a discriminative Multi-head Scene-specific Attention (MSA) module to dynamically control the layer parameters of the detector conditioned on the fusion of scene-conditioned knowledge. Our HyperDet3D achieves state-of-the-art results on the 3D object detection benchmark of the ScanNet and SUN RGB-D datasets. Moreover, through cross-dataset evaluation, we show the acquired scene-conditioned prior knowledge still takes effect when facing 3D scenes with domain gap.

1. Introduction

3D object detection has gained much attention in recent years, which is fundamental for applications such as autonomous driving, robotic navigation and augmented reality. Early works adopt sliding window [42] or 2D prior [16]

†Corresponding author

to locate objects from RGB-D data. However, the orderless and sparse characteristic of point cloud makes it hard to directly employ the recent advances in 2D detection. To tackle this, view-based methods [4] project the points into multiple 2D planes and apply standard 2D detectors. Volumetric convolution-based methods [18, 23] split points into regular grids, which is feasible for 3D convolutions.

Different from the aforementioned view-based and volumetric convolution-based methods, PointNet++ [35] focuses on the local geometries while elegantly consuming raw point cloud, and thus widely used as backbone network in 3D detectors. Built on the PointNet++ network, VoteNet [32] yields outstanding results by regressing offset votes to object centers from seed coordinates and corresponding local features. Following works incorporate probabilistic voting [8], multi-level contextual learning [9, 48, 49] and self-attention based transformer [22, 24, 28] to further enhance the local representations. These methods underline the importance of exploiting object-based and relation-based representation of local elements, such as individual points, detection candidates and irregular local geometries in a given point scan.

However, the attributes of similar objects are ambiguous if we only look at themselves or relations. In this paper, we discover that the scene-level information provides prior knowledge to eliminate such ambiguity. As shown in Figure 1, with the absence of scene-conditioned knowledge, inferring the object-level features or their relations is inadequate for detecting the object candidate, which may lead to counter-intuitive detection results in the aspect of scene-level understanding. To our best knowledge, the acquisition of such scene-level information among various scenes by 3D detectors is yet to be fully studied.

To this end, we propose HyperDet3D for 3D object detection on point cloud which leverages hypernetwork-based structure. Compared with the existing methods that focus on point-wise or object-level representation, our HyperDet3D learns the scene-conditioned information as prior and incorporates such scene-level knowledge into network parameters, so that our 3D object detector is dynamically adjusted in accordance with different input scenes. Specifically, the scene-conditioned knowledge can be factorized into two levels: scene-agnostic and scene-specific information. For the **scene-agnostic** knowledge, we maintain a learnable embedding which is consumed by a hypernetwork and iteratively updated along with the parsing of various input scenes during training. Such sharable scene-agnostic knowledge generally abstracts the characteristics of training scenes and can be utilized by the detector at test time. Moreover, since conventional detectors maintain the same set of parameters when recognizing objects in different scenes, we propose to incorporate the **scene-specific** information which adapts the detector to the given scene at test time. To this end, we attentionally measure how well the current scene matches a general representation (or how much they differ) by using the specific input data as query. We simultaneously learn the two levels of scene-conditioned knowledge by proposing a Multi-head Scene-Conditioned Attention (MSA) module. The learned prior knowledge is aggregated with object candidate features by late fusion, therefore providing more powerful guidance to detect the objects. Extensive experiments on the widely used ScanNet [7] and SUN RGB-D [41] datasets demonstrate that our method surpasses state-of-the-art methods by an obvious margin. Moreover, through cross-dataset evaluation, we show the scene-conditioned prior knowledge acquired by our HyperDet3D still takes effect when faced with domain gap.

2. Related Work

3D Object Detection for Point clouds: Since spatial information is better preserved in point cloud, most state-of-the-art approaches consume raw 3D coordinates as input [19, 37, 51, 54]. Early methods group point cloud into stacked 3D voxels [23, 55] to generate more structured data, or restricts the grouping operation within the ground plane to achieve real-time detection [17]. RCNN methods [5, 19, 37, 38] adopt PointNet-based [34, 35] module or use hybrid representation for better extracting and aggregating the point-wise feature. Inspired by the codebook learning in Hough Voting in 2D object detection [11, 44], VoteNet [32] pioneerly construct the codebook of voting supervision from points to object centers by sampling and grouping proposed in PointNet++ [35]. Based on the framework of VoteNet [32], H3DNet [53] incorporates the votes to additional 3D primitives such as centers of box edges and surfaces. BRNet [6] revisits the back-tracing operation in hough voting by querying the neighboring points around the object candidates. These methods enhance the feature representation of local elements by improving the voting mechanism itself. On the other hand, RGNet [10] models the relation of object proposals by graph structures. SPOT [8] takes the probabilistic voting into account by measuring the information entropy of different local patches. MLCVNet [48, 49] and PointFormer [28] incorporate multi-level attentional learning for object candidates and their contextual information. GroupFree3D [22] and 3DETR [24] introduce the classical Transformer [46] architectures to the detection framework and achieve state-of-the-art performance. These methods explore the relation between local elements such as object candidates, local patches, point coordinates and their clusters.

HyperNetworks in Deep Neural Networks: HyperNetworks [13] output the weights of the target network (called primary network) conditioned on specific input embedding. HyperNetworks have been embedded to replace

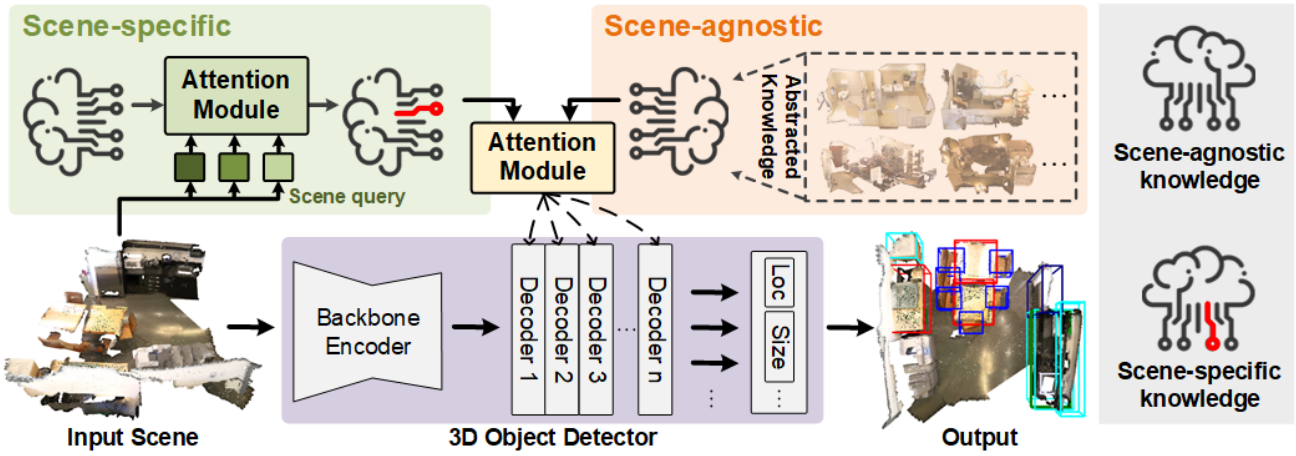

Figure 2. Illustration of the proposed method. For a detection network in the lower half, our HyperDet3D in the upper half attentionally learns both scene-specific and scene-agnostic knowledge. Such scene-conditioned knowledge is then aggregated with object-level features in the decoder layers of detection network, so that the 3D detector is dynamically adjusted in accordance with different input scenes. The scene-agnostic knowledge is the sharable abstract learned from various scenes. The scene-specific knowledge attentionally measures how well a specific scene matches the general embedding (or how much they differ) by using the current scene as query. (*Best viewed in color.*)

the convolution or linear layers in image recognition [13], semantic segmentation [27], neural architecture search [2] and natural language modeling [13]. In the field of 3D understanding, HyperCloud [43] and HyperCube [30] propose to produce a variety of shape representation for a single object by modifying the input to the hypernetwork. SDF-SRN [20] and MetaSDF [40] use hypernetworks to implicitly learn the object semantics within a category. More relevant to our work is HyperGrid [45] which designs the task-conditioned input embeddings of hypernetworks for a multi-task Transformer-based [46] language model. Our HyperDet3D instead implicitly constructs the scene-specific and scene-agnostic embeddings for 3D object detection and, to our knowledge, is the first to incorporate hypernetworks in this task.

3. Approach

In this section, we first briefly introduce the overall architecture and some preliminaries. Next, we elaborate our proposed method. Finally, we provide the implementation details of the proposed method.

3.1. Overview and Preliminaries

Figure 2 illustrates 3 key components in our proposed HyperDet3D, which are the backbone encoder, object decoder layer and detection head. Given an input point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, the backbone firstly downsamples the dense points into initial object candidates, as well as coarsely extracts their features through hierarchical architectures. For fair comparison, we consider PointNet++ [35] as the backbone network similar to previous works [22, 32, 53], which uses furthest point sampling (FPS) to uniformly cover the 3D space. Then the object decoder layers refine the candidate features by incorporating scene-conditioned prior knowledge into object-level representation (elaborated in Sec. 3.2). Finally the detection head regresses the bounding boxes from the location and refined features of those object candidates (elaborated in Sec. 3.3).

To enable HyperDet3D the awareness of scene-level meta information, we adopt HyperNetwork [13] which is a neural network used to parameterize learnable parameters for another network (called primary network). For a target layer in primary network, its learnable parameters $\boldsymbol{W}$ are usually generated by feeding a learnable embedding z or intermediate features x into a hypernetwork $\boldsymbol{H}$:

$$\boldsymbol{W} = \boldsymbol{H}(z) \quad \text{or} \quad \boldsymbol{W} = \boldsymbol{H}(x) \tag{1}$$

Unlike conventional deep neural networks that keep the layer fixed at test time, hypernetworks enable flexibility of learnable parameters by modifying its input.

In HyperDet3D, we propose to use a scene-conditioned hypernetwork to inject prior knowledge into the layer parameters in Transformer decoder, which dynamically adjusts the detection network in accordance with different input scenes.

3.2. Scene-Conditioned HyperNetworks

For the feature representation $\boldsymbol{o}$ of a set of object candidates produced by the backbone encoder, the goal of our scene-conditioned hypernetworks is to endow it with the prior knowledge parameterized by $\{\boldsymbol{W}, \boldsymbol{b}\}$:

$$\hat{\boldsymbol{o}} = \boldsymbol{W}\boldsymbol{o} + \boldsymbol{b} \tag{2}$$

where $\boldsymbol{W} \in \mathbb{R}^{C_{\text{out}}\times C_{\text{in}}}$ and $\boldsymbol{b} \in \mathbb{R}^{C_{\text{out}}}$ are weight and bias parameters in primary detection network. The parameters are

produced by our scene-conditioned hypernetworks, which can be categorized into scene-agnostic and scene-specific hypernetworks.

Scene-Agnostic HyperNetwork: Take the weight parameters $\boldsymbol{W}$ of primary network for example. For scene-agnostic knowledge, we firstly maintain a set of n scene-agnostic embedding vectors $\boldsymbol{Z}^a = \{z_j^a \in \mathbb{R}^{C_a}\}_{j=1}^n$. $\boldsymbol{Z}^a$ is then consumed by a scene-agnostic hypernetwork $\boldsymbol{h}_\theta^a$ which projects z_j^a into another $\mathbb{R}^{C_{ui}}$ space, and the output $\boldsymbol{W}^a$ parameterizes our scene-agnostic knowledge:

$$\boldsymbol{W}^a := \{w_j^a \in \mathbb{R}^{C_{ui}}\}_{j=1}^n,\ w_j^a = \boldsymbol{h}_\theta^a(z_j^a) \tag{3}$$

where C_{ui} is unit fan-in channel size, and satisfies:

$$\text{mod}\ (C_{\text{out}}, n) \equiv 0, \quad \text{mod}\ (C_{\text{in}}, C_{\text{ui}}) \equiv 0 \tag{4}$$

While the object features are iteratively refined by a series of decoder layers [22, 24], they can be consistently incorporated with the output of scene-agnostic hypernetwork which abstracts the prior knowledge of various 3D scenes. In this way, we not only maintain the general scene-conditioned knowledge throughout the decoder layers, but also save the computational cost by sharing the knowledge with rich feature hierarchies.

Scene-Specific HyperNetwork: For scene-specific knowledge, we also learn a set of embedding vectors $\boldsymbol{Z}^s = \{z_k^s \in \mathbb{R}^{C_s}\}_{k=1}^n$ similar to $\boldsymbol{Z}^a$. The difference is, to adapt $\boldsymbol{Z}^s$ to the input scene, our scene-specific hypernetwork $\boldsymbol{h}_\theta^s$ uses the input scene P^i as a scene-specific query. Inspired by the alignment [1] in language model, we measure how well $\boldsymbol{z}_w^s$ matches the input scene (or how much they differ) in the embedding space through attention mechanism:

$$\begin{aligned} \boldsymbol{W}^s &:= \{w_k^s \in \mathbb{R}^{C_{ui}}\}_{k=1}^n \\ w_k^s &= \boldsymbol{h}_\theta^s(z_k^s, \text{P}_d^i) = W_f(z_k^s || W_p \text{P}_d^i) \end{aligned} \tag{5}$$

where $\text{P}_d^i \in \mathbb{R}^{N_d \times 3}$, $W_p \in \mathbb{R}^{C_n \times N_d}$ are a subset of the current input scene, and transformation matrix which projects P_d^i into the embedding space of $\boldsymbol{Z}^s$. W_f represents the weight matrix with Tanh the activation function. As we intend to get responses from the latent embedding space, we use concatenation $(\cdot||\cdot)$ as coding of query points and embedding vectors similar to SDF query [29]. We adopt the downsampled representation P_d^i instead of P^i because hypernetworks, as suggested by the previous research [27], do not fully capture the high-resolution information.

From the set of scene-specific attentional scores $\boldsymbol{W}^s$ and scene-conditioned knowledge $\boldsymbol{W}^a$, now we can get unit block for $\boldsymbol{W}$:

$$\boldsymbol{W}^u = \boldsymbol{W}^s \odot \boldsymbol{W}^a \tag{6}$$

where $\odot$ denotes the element-wise multiplication.

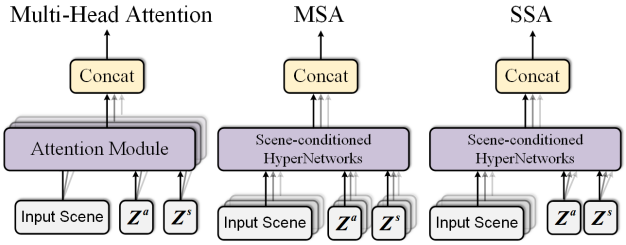

Figure 3. The comparison between Multi-Head Attention [46], our proposed Multi-Head Scene-Conditioned Attention (MSA) and Single-Head Scene-Conditioned Attention (SSA).

Multi-Head Scene-Conditioned Attention: For the i-th input scene P^i, the abovementioned process can be encapsulated into 2 scene-conditioned attention operations:

$$\boldsymbol{W}^u = \text{Att}_2(\{z_j^a\}, \text{Att}_1(\{z_k^s\}, \text{P}^i)) \tag{7}$$

where Att_1 and Att_2 correspond to the attention in (5) and (6) respectively. To fit the shape of target weights $\boldsymbol{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ for primary network, a simple solution is to repeat $\boldsymbol{W}^u$ by $\frac{C_{\text{out}}}{n} \times \frac{C_{\text{in}}}{C_{\text{ui}}}$ times and tile them along its 2 dimensions. The feasibility is guaranteed by (4). As $\boldsymbol{Z}^a$ and $\boldsymbol{Z}^s$ are initialized and consumed by hypernetworks only once, we name it Single-Head Scene-Conditioned Attention (SSA).

To allow the primary detector to jointly attend to the scene-conditioned knowledge in various sub-spaces, we further propose Multi-Head Scene-Conditioned Attention (MSA) based on SSA. The idea of multi-head attention is proposed in [46] which consumes the same set of input via parallel attention modules. However, as target weights $\boldsymbol{W}$ are conditioned on the input of hypernetworks in our case, we instead implement MSA by re-initializing $\boldsymbol{Z}^a$ and $\boldsymbol{Z}^s$ multiple times. Therefore, our MSA can be formulated as:

$$\boldsymbol{W} = \text{Concat}(\boldsymbol{W}_{(1)}^u, \boldsymbol{W}_{(2)}^u, \ldots, \boldsymbol{W}_{(\frac{C_{\text{out}}}{n} \times \frac{C_{\text{in}}}{C_{\text{ui}}})}^u) \tag{8}$$

where $\boldsymbol{W}_{(l)}^u$ denotes the result in (7) produced by the l-th initialization of $\boldsymbol{Z}^a$ and $\boldsymbol{Z}^s$. The Concat operation tiles the matrices along 2 dimensions similar to SSA.

In Figure 3, we illustrate the comparison between the original Multi-Head Attention [46], our Multi-Head Scene-Conditioned Attention (MSA) and Single-Head Scene-Conditioned Attention (SSA). The computation overhead for a single input sample in [46] is proportional to the number of parallel attention modules which define the attentional sub-spaces. Instead, the MSA network is shared between all training samples in our HyperDet3D. Moreover, as we mine the sub-spaces via hypernetwork structures, MSA exploits the flexibility of scene-conditioned knowledge via modifying the input in (1). In comparison, SSA consumes the same set of embedding vectors and is inferior to MSA in terms of expressiveness, which we verify in the ablation experiments.

The pipeline of obtaining the bias parameters $\boldsymbol{b}$ is similar to that of $\boldsymbol{W}$, which we display in the supplementary pages. $\boldsymbol{W}$ and $\boldsymbol{b}$ are aggregated with object features as in (2). The renewed representation $\hat{\boldsymbol{o}}$ is then consumed by the detection head to generate the detection results.

3.3. Disentangled Detection Head

Following [32], existing works locate the object center $\boldsymbol{c}_i$ via directly regressing an offset ($\Delta \boldsymbol{q}_i$) from the candidate location $\boldsymbol{q}_i$ by a detection head parameterized by $\boldsymbol{W}_c$:

$$\boldsymbol{c}_i = \boldsymbol{q}_i + \Delta \boldsymbol{q}_i,\ \Delta \boldsymbol{q}_i = \boldsymbol{W}_c \hat{\boldsymbol{o}}_i \tag{9}$$

Here we use a Disentangled variant of Detection Head (DDH) which factorizes the offset regression into 2 branches. Given a predicted $\Delta \boldsymbol{q}_i$, one branch regresses a scalar $r \in \mathbb{R}^1$ to modulate its length, and another regresses a 4-dim vector, $\boldsymbol{R} \in \mathbb{R}^4$, to modulate its orientation. Each branch contains a light-weighted regression head. $\boldsymbol{R}$ is regarded as the real part of a quaternion, which can be transformed into a rotation matrix to modulate the orientation of $\Delta \boldsymbol{q}_i$. Therefore, the final offset $\Delta \boldsymbol{q}_i^{'}$ is computed as follows:

$$\Delta \boldsymbol{q}_i^{'} = f_T(\boldsymbol{R}) * (r \Delta \boldsymbol{q}_i) \tag{10}$$

where $*$ denotes dot production. f_T is the transformation function defined in [39] which converts the quaternion into a 3x3 rotation matrix. Note that $\boldsymbol{R}$ is firstly L2-normalized when being transformed.

3.4. Implementation Details

The backbone network PointNet++ [35] in HyperDet3D contains 4 set abstraction layers which downsample the input scan into {2048,1024,512,256} points consecutively. The radius for ball query is {0.2m,0.4m,0.8m,1.2m}. Then 2 feature propagation layers recover them into 1024 points and produce the point-wise features. We use the KPS proposed in [22] to generate object candidates from the original locations of these 1024 points, as it saves the computational cost in $\mathrm{O}(N^2)$ search space of FPS [35].

To obtain $\boldsymbol{o}$ in each decoder layer, we follow [22, 24] to employ the standard multi-head attention layer to compute the self-attention of object candidates, followed by the cross-attention between object candidates and downsampled points produced by the backbone. The scene-agnostic hypernetwork in (3) contains 2 linear layers. The scene-specific hypernetwork in (5) contains 1 linear layer followed by Tanh activation function. Each linear layer is parameterized by a weight matrix and bias vector, initialized by Xavier [12] and zeros. For the scene query P_d^i of scene-specific hypernetwork, we use the off-the-shelf downsampled results of KPS.

As for detection head, each light-weighted regression head mainly contains a fully-connected (FC) layer to map $\boldsymbol{f}_i$ into r or $\boldsymbol{R}$. In r-head, the output of FC layer is processed by sigmoid function and further normalized into [0.9,1.1] to control the extent of adjustment. In $\boldsymbol{R}$-head, identity quaternion is added to $\boldsymbol{R}$ before transformation (f_T), which can simultaneously hold the possibility of identity rotation and control the rotation degree.

4. Experiment

In experiment section, we firstly introduce the datasets and evaluation metrics of the benchmark for 3D object detection (Sec. 4.1). We then display the thorough experimental results by comparing HyperDet3D with state-of-the-art approaches both quantitatively and qualitatively (Sec. 4.2). We also analyze the design choice and effectiveness of HyperDet3D by ablation studies and cross-dataset evaluation (Sec. 4.3). Finally we point out the limitation of our work (Sec. 4.4). More analysis and visualizations are provided in the supplementary pages.

4.1. Datasets and Settings

ScanNet V2: The ScanNet V2 dataset [7] includes 1,513 scanned and reconstructed indoor scenes, with axis-aligned bounding box labels for 18 object categories. The point cloud data are converted from reconstructed meshes. Following [32], we employ 1,201 scenes as the training set and the rest 312 validation scenes as the test set.

SUN RGB-D V1: The SUN RGB-D V1 dataset [41] contains 10k single-view indoor RGB-D images, 5,285 for training and 5,050 for testing. It's densely annotated with 64k oriented 3D bounding boxes. The whole dataset is categorized into 37 indoor object classes. For fair comparison, we follow the evaluation protocol in [32] which selects the 10 most common categories.

For both datasets, we only employ point cloud data as the input. No scene-level supervision is employed by HyperDet3D. Following [32], we report the detection performance on the validation sets by computing mean Average Precision (mAP) with 3D IoU threshold 0.25 (mAP@0.25) and 0.5 (mAP@0.5). Detection performance on individual categories and their average results are displayed.

As for the training strategy, in the first 100 epochs of both datasets, the detection head directly consumes $\boldsymbol{o}$ rather than $\hat{\boldsymbol{o}}$ in (2). Then the network was finetuned for 300 and 500 epochs on ScanNet and SUN RGB-D respectively, using $\hat{\boldsymbol{o}}$ instead. The strategy aims for the stability of loss curves when incorporated with hypernetworks. The finetuned network was used for inference at test time. The details of hyper-parameters for 2 datasets can be found in the supplementary material.

4.2. Main Results

Quantitative results: We compare our HyperDet3D quantitatively with a number of reference methods, which

Table 1. 3D object detection results on the ScanNet V2 validation set (left) and the SUN RGB-D V1 validation set (right). Evaluation metric is average precision with 3D IoU thresholds as 0.25 and 0.50. Results of H3DNet [53] are reported under 4 PointNet++ backbones settings. Results of 3DETR [24] are reported on its stronger 3DETR-m variant with inductive biases.

ScanNet V2	Input	mAP@0.25	mAP@0.50
DSS [42]	Geo + RGB	15.2	6.8
MRCNN [14]	Geo + RGB	17.3	10.5
F-PointNet [33]	Geo + RGB	19.8	10.8
GSPN [52]	Geo + RGB	30.6	17.7
3D-SIS [15]	Geo + 5 views	40.2	22.5
VoteNet [32]	Geo only	58.6	33.5
GCENet [21]	Geo only	60.7	-
HGNet [3]	Geo only	61.3	34.4
DOPS [25]	Geo only	63.7	38.2
H3DNet* [53]	Geo only	67.2	48.1
BRNet [6]	Geo only	66.1	50.9
VENet [47]	Geo only	67.7	-
RGNet [10]	Geo only	48.5	26.0
SPOT [8]	Geo only	59.8	40.4
MLCVNet [48]	Geo only	64.7	42.1
PointFormer [28]	Geo only	64.1	42.6
3DETR* [24]	Geo only	65.0	47.0
GF3D [22]	Geo only	69.1	52.8
Ours	Geo only	**70.9**	**57.2**

SUN RGB-D	Input	mAP@0.25	mAP@0.50
DSS [42]	Geo + RGB	42.1	-
2D-driven [16]	Geo + RGB	45.1	-
PointFusion [50]	Geo + RGB	45.4	-
COG [36]	Geo + RGB	47.6	-
F-PointNet [33]	Geo + RGB	54.0	-
VoteNet [32]	Geo only	57.7	32.9
H3DNet* [53]	Geo only	60.1	39.0
VENet [47]	Geo only	62.5	39.2
GCENet [21]	Geo only	60.8	40.1
HGNet [3]	Geo only	61.6	-
ImVoteNet [31]	Geo + RGB	63.4	-
BRNet [6]	Geo only	61.1	43.7
3DETR* [24]	Geo only	59.1	32.7
RGNet [10]	Geo only	59.2	-
MLCVNet [48]	Geo only	59.8	-
SPOT [8]	Geo only	60.4	36.3
PointFormer [28]	Geo only	61.1	36.6
GF3D [22]	Geo only	63.0	45.2
Ours	Geo only	**63.5**	**47.3**

can be divided into 3 categories: early approaches that require 2D guidance to locate 3D objects [15, 16, 33, 36, 42, 50, 52], voting-based approaches that explore optimal local representation to provide informative cues [3, 6, 21, 25, 31, 32, 47, 53], and relation-based approaches that explore the interaction between local elements such as objects or point clusters [8, 10, 22, 24, 28, 48]. The experimental results are shown in Table 1 and Table 2. Bold indicates the best results under the corresponding metrics.

From the comparison results in Table 1, we can observe that the state-of-the-art relation-based GF3D [22] outperforms all the other compared methods, except for ImvoteNet [31] which incorporates 2D image votes. However, thanks to the acquired scene-conditioned prior knowledge, our HyperDet3D stills achieves leading average on 2 metrics of both ScanNet V2 (+1.8% mAP@0.25, +4.4% mAP@0.5) and SUN RGB-D V1 (+0.5% mAP@0.25, +2.1% mAP@0.5) validation set. Note that compared with SUN RGB-D, ScanNet is annotated with 1.8x as many categories for 3D detection task. Therefore, the scene-level prior knowledge learned by HyperDet3D content is relatively richer in ScanNet than SUN RGB-D, and yields more significant mAP gain on the former dataset.

We then look into the per-category results of mAP@0.5 on ScanNet V2 validation set, which is the benchmark with more categories, more challenging threshold for evaluation, and more performance gain by our method. The detailed results are displayed in Table 2. For the categories largely conditioned on the scene prior (such as *bed* in bedroom, *fridge* in kitchen/canteen, *shower curtain/toilet/sink/bathtub* in bathroom), they consistently obtain notable AP gain compared with the baseline methods. This indicates the effectiveness of learned scene-conditioned knowledge by HyperDet3D. The performance drops on the *counter* category which is less conditioned on the scene-level semantics. We display the detailed results on SUN RGB-D in the supplementary pages.

In Table 3, we compare our method with the state-of-the-art GF3D [22] furtherly[1]. It can be seen that in a normal or light-weighted version of network configurations, our approach outperforms GF3D in both metrics while containing notably fewer learnable parameters. Therefore, HyperDet3D is likely to efficiently absorb the external data due to the mechanism of scene-conditioned hypernetworks and knowledge sharing in different layers.

Qualitative Results: In Figure 4, we illustrate the representative 3D object detection results of 4 scans in ScanNet V2 validation set. Taking groundtruth annotations (GT) and real image scans as reference, we compare our HyperDet3D with the state-of-the-art GF3D [22] which involves the dense interaction between object candidates. The first 3 scans highlight the ambiguity in the aspect of largely intersected bounding boxes, where the baseline module mistakes a refrigerator or washing machine for a cabinet, or detect a sink in an office. With the help of scene-conditioned prior knowledge, our HyperDet3D can obtain better detection results on these objects. The ambiguous detections also include the mistaken detections. For example, in the last scan, the baseline method mistakes a cabinet in a bedroom for a

[1]In Table 3, as suggested by [22], L denotes the number of decoders; O denotes the number of object candidates; and w2× denotes the feature dimension in backbone is expanded by 2 times.

Table 2. 3D object detection results on the ScanNet V2 validation dataset. We show per-category results of mean average precision (mAP) with 3D IoU threshold 0.5 as proposed in [41], and mean of AP across all semantic classes with 3D IoU threshold 0.5.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
Votenet [32]	8.1	76.1	67.2	68.8	42.4	15.3	6.4	28.0	1.3	9.5	37.5	11.6	27.8	10.0	86.5	16.8	78.9	11.7	33.5
DOPS [25]	25.2	70.2	75.8	54.8	41.2	27.8	12.1	21.4	**12.3**	9.5	39.4	24.4	33.7	17.3	80.6	35.7	71.0	35.0	38.2
MLCVNet [48]	16.6	83.3	78.1	74.7	55.1	28.1	17.0	51.7	3.7	13.9	47.7	28.6	36.3	13.4	70.9	25.6	85.7	27.5	42.1
PointFormer [28]	19.0	80.0	75.3	69.0	50.5	24.3	15.0	41.9	1.5	26.9	45.1	30.3	41.9	25.3	75.9	35.5	82.9	26.0	42.6
H3DNet [53]	20.5	79.7	80.1	79.6	56.2	29.0	21.3	45.5	4.2	33.5	50.6	37.3	41.4	37.0	89.1	35.1	90.2	35.4	48.1
BRNet [6]	28.7	80.6	81.9	80.6	60.8	35.5	22.2	48.0	7.5	**43.7**	54.8	39.1	51.8	35.9	88.9	38.7	84.4	33.0	50.9
GF3D [22]	26.0	81.3	82.9	70.7	**62.2**	41.7	26.5	**55.8**	7.8	34.7	67.2	43.9	44.3	44.1	92.8	37.4	89.7	40.6	52.8
Ours	**33.1**	**90.1**	**83.8**	**83.8**	60.3	**43.6**	**31.7**	52.2	4.2	20.9	**78.5**	**49.0**	**61.1**	**56.3**	**95.9**	**43.9**	**100**	**42.3**	**57.3**

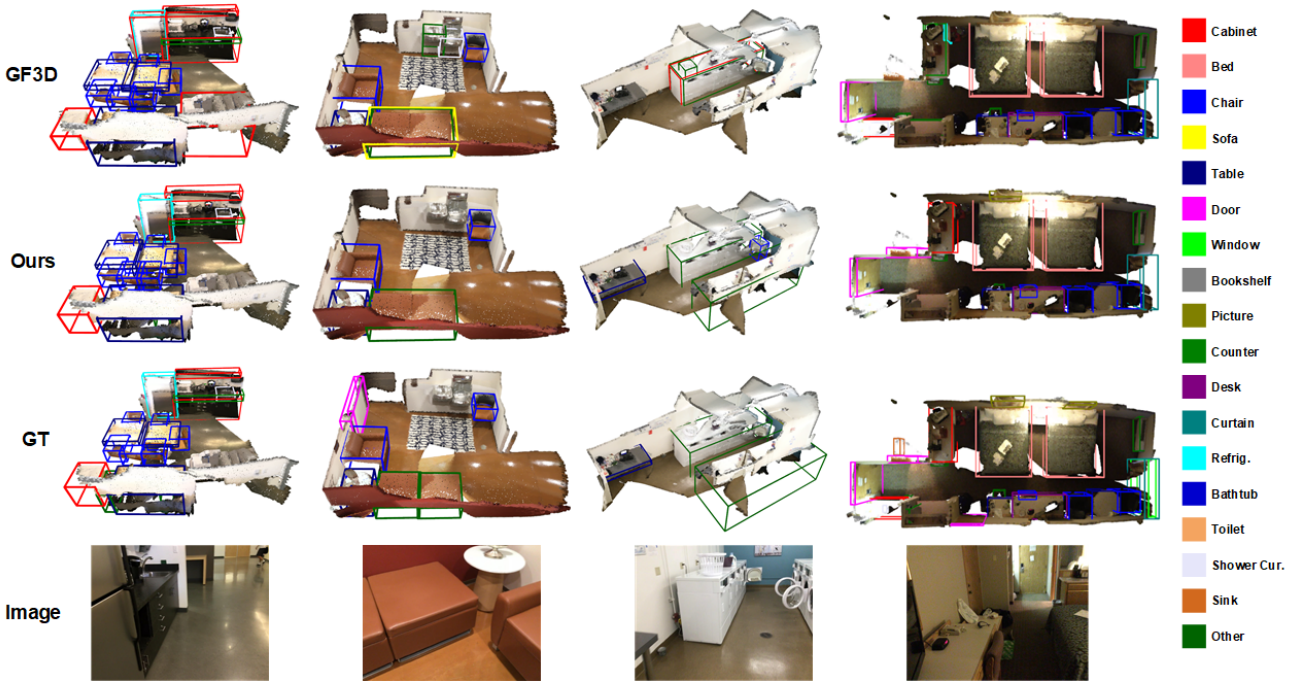

Figure 4. Qualitative comparisons between our approach and GF3D [22] baseline approach on the ScanNet V2 validation set. The groundtruth annotations (GT) and 2D image scans are taken as reference. Our method achieves favourable results compared to the baseline method. Fewer ambiguous detections are observed in our results. The point clouds are colored only for easy illustration, and not utilized in the compared method nor ours. (*Best viewed in color.*)

Table 3. Comparison with GroupFree-3D [22] (GF3D) with various configurations on the ScanNet V2 validation set. The upper section shows results for GF3D models reported in [22].

Model	backbone	#params	mAP@0.5
GF3D-(L6,O256)	PointNet++	14.5M	48.9
GF3D-(L12,O512)	PointNet++w2×	29.6M	52.8
Ours-(L6,O256)	PointNet++	11.1M	51.0
Ours-(L12,O512)	PointNet++w2×	22.6M	57.2

counter.

4.3. Ablation Study and Discussions

To analyze the importance of learned scene-conditioned knowledge in our HyperDet3D network, we conducted ablation experiments on various combinations of design choices. The quantitative results are shown in Table 4. The baseline model only uses disentangled detection head and we gray its corresponding row for clear comparison. Applying the SSA to learn scene-conditioned knowledge leads to improvement of mAP@0.25 by 1.2%, and mAP@0.5 by 1.2%. The multi-head variant (MSA) further brings +1.1% mAP@0.25 and +2.5% mAP@0.5. As expected, only learning scene-agnostic or scene-specific prior knowledge is inadequate for thorough scene-conditioned understanding. For the challenging mAP@0.5 metric, only learning scene-agnostic or scene-specific knowledge causes performance drop by -1.8% and -3.4% respectively. The removal of disentangled regression of center offsets leads to a performance drop by -0.6% mAP@0.25 and -0.4% mAP@0.5, which indicates that the exquisite regression of targets helps to leverage the learned scene-conditioned knowledge.

Cross-dataset evaluation: Since HyperDet3D learns

Table 4. Experimental results of ablation studies on the ScanNet V2 validation set. The baseline method only applies the disentangled detection head (DDH) on candidate features without scene-conditioned prior knowledge (the row colored in gray).

Scene-Conditioned		Attention		DDH	mAP@0.25	mAP@0.5
agnostic	specific	SSA	MSA			
				✓	68.6	53.5
✓	✓	✓		✓	69.8	54.7
	✓		✓	✓	70.6	55.4
✓			✓	✓	70.0	53.8
✓	✓		✓		70.3	56.8
✓	✓		✓	✓	**70.9**	**57.2**

Table 5. Cross-dataset evaluation results on the ScanNet V2 val dataset, which is pre-trained on the SUN RGB-D V1 val dataset. We show mAPs of 8 shared categories between ScanNet V2 and SUN RGB-D, and all 18 categories of ScanNet V2. The 3D IoU threshold of mAP is 0.5.

	bed	chair	sofa	tabl	bkshf	desk	toil	bath	mAP_8	mAP_{18}
VoteNet	30.3	21.7	12.4	8.3	**4.4**	4.4	21.7	33.4	17.1	8.4
GF3D	66.6	21.3	46.9	**17.8**	0.4	25.6	54.6	48.6	35.2	19.1
Ours	**78.9**	**22.7**	**58.0**	16.0	2.4	**40.1**	**58.9**	**71.4**	**43.6**	**22.2**

the scene-conditioned knowledge as prior, we infer such knowledge acquired by the detector still takes effect when faced with domain gaps. To validate this, we conducted cross-dataset evaluation in comparison with VoteNet [32] and GF3D [22] as the baseline detectors. We firstly pre-trained the baseline detectors and HyperDet3D on the SUN RGB-D V1 validation set then finetuned on the ScanNet V2 validation set. The backbone networks in all 3 approaches and the scene-conditioned hypernetworks in ours were frozen during finetuning.

In Table 5, we show the detection mAP of 8 shared categories between SUN RGB-D and ScanNet with IoU threshold set as 0.5, as well as average mAP over all 18 (mAP_8) categories in ScanNet or the shared 8 (mAP_{18}) categories. The observation is two-fold. Our HyperDet3D surpasses the baseline methods on both mAP_8 and mAP_{18}, especially on the shared categories between 2 datasets. This indicates that the scene-conditioned knowledge learned on the source dataset can be well transferred to the target dataset by our method. On the other hand, among the 8 shared categories, those more conditioned on the scene semantics are improved by an obvious margin similar to the results in Table 1. The exception is the *bookshelf* category partially due to the scarcity of library scenes (1.9%) in SUN-RGBD [41]. Moreover, the novel categories such as *refrigerator* and *sink* are improved by +14.9% and +11.1% respectively. The details can be found in supplementary pages.

Incorporation of scene labels: An interesting question is, what if we utilize the groundtruth scene label as an additional supervision? To this end, we added a classification(Cls.) branch to the bottleneck of the backbone in GF3D, and finetuned the whole network for 100 epochs based on the GF3D pretrained model. The ScanNet results

Table 6. Comparison with Multi-task classification (Cls.) baseline.

HyperDet3D		GF3D		GF3D+Cls.	
mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
70.9	**57.2**	69.1	52.8	69.4	54.3

in Table 6 suggest the additional branch with scene type labels improves the detection performance, but is still inferior to HyperDet3D. Note that our HyperDet3D achieves the best results without any supervision on the scene type classification. We expect better detection performance by training a unique detector for each type of scene. However, this may limit the generality of the method and is less computationally friendly for real-world applications.

4.4. Limitations

As we focus on the scene-level information, we can observe some failure cases on detailed local geometries. For example, in the second example of Figure 4, HyperDet3D misdetects 2 closely connected objects as a whole. A possible solution is to incorporate more detailed representation of scene query, which may require SDF [29] to exquisitely model the geometries in the scene. Moreover, another important work Mix3D [26] proposes to reduce the scene-level variation by enriching the scene-level data with object-level information, while HyperDet3D aims to utilize such scene-specific variation by endowing object representation with scene-prior knowledge. We expect a future solution might combine the advantages of both methods.

5. Conclusion

In this paper we have introduced HyperDet3D: a new framework to explore scene-conditioned prior for 3D object detection. Our HyperDet3D simultaneously learns the scene-agnostic knowledge which explores the sharable abstracts from various 3D scenes, and scene-specific knowledge which adapts the detector to the given scene. HyperDet3D achieves state-of-the-art results on the 3D object detection benchmark of 2 widely-used datasets, and demonstrates effectiveness when faced with domain gap.

Potential Impact: Our method aims to improve the researches on 3D object detection, which is critical for the safety of robotic systems. Similar to many deep learning methods, one potential negative impact is that it still lacks theoretical guarantees. To improve the applicability in this domain, the community might consider challenges of explainability and transparency.

Acknowledgements: This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603, Grant U1813218, in part by Beijing Academy of Artificial Intelligence (BAAI), and in part by a grant from the Institute for Guo Qiang, Tsinghua University.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 4

[2] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. In *ICLR*, 2018. 3

[3] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *CVPR*, pages 392–401, 2020. 6

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, pages 1907–1915, 2017. 2

[5] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *ICCV*, pages 9775–9784, 2019. 2

[6] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *CVPR*, pages 8963–8972, 2021. 2, 6, 7

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828—-5839, 2017. 2, 5

[8] Hongyuan Du, Linjun Li, Bo Liu, and Nuno Vasconcelos. Spot: Selective point cloud voting for better proposal in point cloud object detection. In *ECCV*, pages 230–247, 2020. 2, 6

[9] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *CVPR*, pages 949–958, 2019. 2

[10] Mingtao Feng, Syed Zulqarnain Gilani, Yaonan Wang, Liang Zhang, and Ajmal Mian. Relation graph network for 3d object detection in point clouds. *TIP*, 30:92–107, 2020. 2, 6

[11] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 33(11):2188–2202, 2011. 2

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTAS*, pages 249–256, 2010. 5

[13] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017. 2, 3

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 6

[15] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, pages 4421–4430, 2019. 6

[16] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object detection in rgb-d images. In *ICCV*, pages 4622–4630, 2017. 1, 6

[17] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2

[18] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *IROS*, pages 1513–1518, 2017. 2

[19] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*, pages 7546–7555, 2021. 2

[20] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. Sdf-srn: Learning signed distance 3d object reconstruction from static images. *NeurIPS*, 33:11453–11464, 2020. 3

[21] Xu Liu, Chengtao Li, Jian Wang, Jingbo Wang, Boxin Shi, and Xiaodong He. Group contextual encoding for 3d point clouds. *NeurIPS*, 33:13413–13422, 2020. 6

[22] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, pages 2949–2958, 2021. 2, 3, 4, 5, 6, 7, 8

[23] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, pages 922–928, 2015. 2

[24] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, pages 2906–2917, 2021. 2, 4, 5, 6

[25] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. Dops: Learning to detect 3d objects and predict their 3d shapes. In *CVPR*, pages 11913–11922, 2020. 6, 7

[26] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *3DV*, pages 116–125, 2021. 8

[27] Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *CVPR*, pages 4061–4070, 2021. 3, 4

[28] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *CVPR*, pages 7463–7472, 2021. 2, 6, 7

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 4, 8

[30] Magdalena Proszewska, Marcin Mazur, Tomasz Trzciński, and Przemysław Spurek. Hypercube: Implicit field representations of voxelized 3d models. *arXiv preprint arXiv:2110.05770*, 2021. 3

[31] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *CVPR*, pages 4404–4413, 2020. 6

[32] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 2, 3, 5, 6, 7, 8

[33] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, pages 918–927, 2018. 6

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 2, 3, 5

[36] Zhile Ren and Erik B Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, pages 1525–1533, 2016. 6

[37] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, pages 10529–10538, 2020. 2

[38] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 2

[39] Ken Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH*, pages 245–254, 1985. 5

[40] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *NeurIPS*, 33:10136–10147, 2020. 3

[41] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015. 2, 5, 7, 8

[42] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, pages 808–816, 2016. 1, 6

[43] Przemysław Spurek, Sebastian Winczowski, Jacek Tabor, Maciej Zamorski, Maciej Zieba, and Tomasz Trzcinski. Hypernetwork approach to generating point clouds. In *ICML*, pages 9099–9108, 2020. 3

[44] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, pages 658–671, 2010. 2

[45] Yi Tay, Zhe Zhao, Dara Bahri, Donald Metzler, and Da-Cheng Juan. Hypergrid transformers: Towards a single model for multiple tasks. In *ICLR*, 2021. 3

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 3, 4

[47] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Dening Lu, Mingqiang Wei, and Jun Wang. Venet: Voting enhancement network for 3d object detection. In *ICCV*, pages 3712–3721, 2021. 6

[48] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *CVPR*, pages 10447–10456, 2020. 2, 6, 7

[49] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Vote-based 3d object detection with context modeling and sob-3dnms. *IJCV*, 129(6):1857–1874, 2021. 2

[50] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *CVPR*, pages 244–253, 2018. 6

[51] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, pages 11040–11048, 2020. 2

[52] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, pages 3947–3956, 2019. 6

[53] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *ECCV*, pages 311–329, 2020. 2, 3, 6, 7

[54] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Sessd: Self-ensembling single-stage object detector from point cloud. In *CVPR*, pages 14494–14503, 2021. 2

[55] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2