iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** III **Month of publication:** March 2022

DOI: https://doi.org/10.22214/ijraset.2022.41001

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

Corporeality (BR): A Weakly-supervised Object Detection Approach with 3D Shape-guided Label Enhancement

Bidisha Mondal
Computer Science and Engineering Department, Sikkim Manipal Institute of Technology, Majhitar, Rangpo

Abstract: ***This paper, proposes a weakly-supervised approach for three-dimensional object detection, which makes it possible to train a strong three-dimensional detector with position-level based annotations (i.e. an- notations pertaining to the centre of an object). In an attempt to rectify this information loss from box annotations to object centres, the proposed method, named Corporeality (referred to as BR in short in image and tabular representations) makes use of synthetic three-dimensional shapes to convert weak labels into completely annotated virtual scenes as stronger supervision, and then in turn utilizes these perfect virtual labels to complement and refine the original set of labels. The process involves the assemblage of three-dimensional shapes into physically reasonable virtual scenes according to the coarse scene layout extracted from position-level based annotations previously. Then we go back to reality by applying a virtual-to-real do-main adaptation function, which refines the weak labels along with supervising the three-dimensional detector's training with the virtual scenes. This paper further proposes a more challenging benchmark for three-dimensional object detection with more diverse object sizes to better emphasize the potential of Corporeality. With an investment of a meagre 5% labelling labour, Corporeality was able to perform competitively when compared with some of the popular fully-supervised approaches out there, widely used with ScanNet datasets. Code is available at: https://github.com/mondalbidisha/Corporeality-BR.***

I. INTRODUCTION

Three-dimensional object detection is basically a fundamental scene comprehending problem, which aims to detect the three-dimensional bounding boxes and semantic labels from the point cloud of three-dimensional scene. Owing to the irregularities in the form of point clouds and complex contexts in three-dimensional scenes, most existing two dimensional approaches [33, 34, 52] cannot be directly implemented with three-dimensional object detection.

Fortunately, with the development of deep learning techniques on point cloud understanding [29,30] and recent works [13,22,27,37,53] proposing the efficacy of deep neural networks at directly detecting objects from point clouds has received intrigue as well as favourable critique.

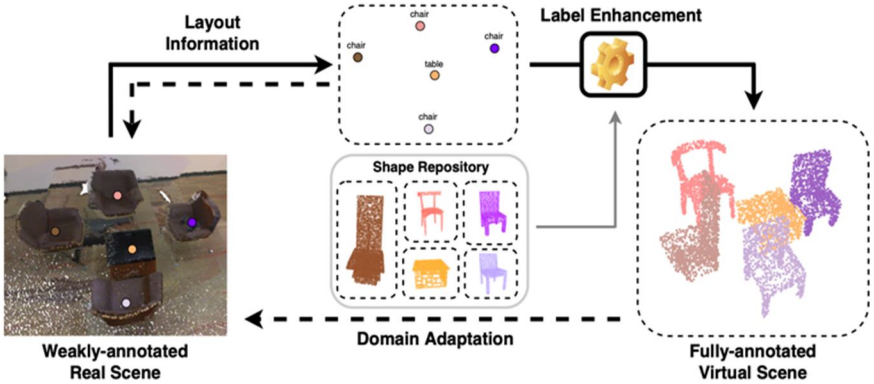


Figure 1. Demonstration of *Corporeality (BR).* The position-level based annotations are regarded as the coarse layout of these scenes, which is utilized to generate virtual scenes from a three-dimensional shape repository. Physical constraints are applied on these virtual scenes to remedy the information loss from box annotations to centres. Then a virtual-to-real domain adaptation method is presented to additionally supervise the real-scene three-dimensional object detection with the virtual scenes. Dashed arrows indicate supervision for training.

Despite the success of deep learning supervision at object detection on point cloud, a significantly large amount of labelled bounding boxes are essential for training the three-dimensional detector. This concern drastically inhibits the application of these methods, as labelling a three-dimensional box with precision takes more than 100 seconds even while being executed by an experienced annotator [38]. Therefore, three-dimensional object detection methods using sub-standard labels are desirable only for practical applications. Driven by this, an exceeding amount of attention has been paid to weakly-supervised three-dimensional object detection techniques, which can be classified into the following two categories basis the annotation forms :- scene-level [35] and position- level [23, 24] with {class tag} and {object centre, class tag} annotated for each object respectively. The two types of annotations only require less than 1% and 5% of the initial amount of time for a single instance as compared to labelling a bounding box, as shown in ***Table 1***.

Annotation	BBox [22]	S-L [35]	P-L [23]	P-L(BR)
Time(s per object)	110	1	5	5
mAP@0.25(%)	54.2	<20	32.4	47.0

Table 1. Annotating time and detection results of methods based on different types of annotation. The benchmark is detailed in Section 4. (BBox refers to bounding-box annotation. S-L and P-L indicate scene-level and position-level annotations respectively.)

While scene-level annotation is less time-consuming, it is hard for the detector to learn how to precisely locate each object in a scene due to the lack of position-level based annotation information, and thus the performance is far from satisfactory [35]. Considering the time accuracy trade-off, position-level annotation is a more practical approach. However, previous position-level based weakly-supervised three-dimensional detection models still require a certain number of precisely labelled boxes and can only comprehend sparse outdoor scenes [23, 24]. A Purely position-level based weakly-supervised approach for the complex indoor object detection is still under exploration.

This paper proposes a shape-guided label enhancement technique referred to as ***Corporeality (BR)*** for weakly-supervised three-dimensional object detection[1]. In an attempt to offset the effort and labour cost, the algorithm only labels the centre of each object in a three-dimensional space and erroneous labelling of centres is permitted[2]. While this significantly reduces the workload of labelling, the information loss is non-negligible compares to box annotations pertaining to a three-dimensional objects' centres. In order to solve this predicament, ***Corporeality (BR)*** converts these weak labels into virtual scenes which contain most of the lost information, and this in turn is utilized by the algorithm to incrementally supervise a real-scene training, as shown in ***Figure 1***. This approach is based on two motivations:

1) In three-dimensional vision, large-scale datasets of synthetic shapes are available. They contain detail-rich geometric information, which can serve well and assist in three-dimensional object detection

2) The position-level based annotations are not only a supervision for the training, but they also provide a coarse layout of the developing scene. Therefore, the algorithm assembles these three-dimensional shapes into a fully-annotated virtual scene in accordance to the coarse layout and apply physical constraints on them to remedy the information loss. Then a virtual-to-real domain-mapping adaptation method is presented to align the global features and object proposal features extracted by the detector between the real and virtual scenes. Moreover, this method can take advantage of the precise centre labels in a virtual scene and correct the centre error of position-level based annotations as well. This way the useful knowledge stored in virtual scenes is leveraged and consequently transferred back to reality. Experimental results on ScanNet [9] show the effectiveness of the proposed ***Corporeality (BR)*** method.

II. RELEVANT WORK

A. Three-dimensional - Shape to Scene

As it is much easier to obtain a large scale synthetic three-dimensional shape dataset than a real scene dataset, utilizing the shapes to assist scene comprehension is a promising technique. Existing approaches can be broadly classified into the following two categories: supervised [4, 5, 8, 42] and unsupervised [10, 21, 26, 32, 44]. In case of the supervised approach, a synthetic shape is usually used to complete the imperfections in the real scene scans. Given a set of CAD models and a real scan, a network is trained to predict how to place the CAD models in the scene and replace the partial and noisy real objects [4,5,8,42]. Human-annotated pairs of raw scans and object-aligned scans are used in the training process. Since supervised approaches need extra human effort, this drawback might limit the maximised utilization of three-dimensional shape datasets. Unsupervised methods are often used for data augmentation or dataset expansion.

Three-dimensional CAD models are placed in a random manner by following the basic physical constraints, in order to generate mixed reality scenes [10, 44] or virtual scenes [21, 26]. Recently, RandomRooms [32] proposed the use ShapeNet dataset for unsupervised pre-training of their three-dimensional detectors. This approach also leverages three-dimensional shapes to assist with object detection in an unsupervised manner. The aim here is to employ synthetic shapes to enhance the weak labels and obtain stronger supervision in position-level based weakly-supervised detection tasks.

B. Three-dimensional - Shape to Scene

Early three-dimensional object detection approaches primarily involved template-based functions [18,20,25] and the sliding-window approach [39, 40]. Deep learning-based three-dimensional object detection frameworks for point clouds began to emerge thanks to the advent of PointNet/PointNet++ [29, 30]. However, [6, 7, 17, 28] relying on generating two-dimensional proposals followed by projecting them onto a three-dimensional space is fruitless as it gets tedious to handle scenes with heavy occlusion. More recently, networks that directly consume point clouds have been proposed [13, 22, 27, 37, 53]. While the development of three-dimensional object detection techniques is fast and evolving, the application is still restricted partially because of limited labelled data availability. In order to reduce the human annotation efforts, weakly-supervised methods [23, 24, 31, 35], semi-supervised methods [43, 51] and unsupervised pre-training methods [14, 32, 47, 49] have been proposed recently. However, pre-training methods rely on a large amount of computing resources for the training of these networks in a contrastive learning manner. Semi-supervised methods follow an approach similar to their two-dimensional counterparts [41] and do not entirely explore the characteristics of three-dimensional data. Therefore, a weakly-supervised approach was built and tailored for this three-dimensional object detection task.

III. APPROACH

Figure 2 illustrates this approach. Given real scenes with position-level based annotations, this algorithm utilizes three-dimensional shapes to convert weak labels into virtual scenes, which are then utilized to provide additional supervision for the training of the three-dimensional detector. In this section, the initial discussion pertains to a weakly-supervised setting and then demonstrates the steps involved in *Corporeality (BR).*

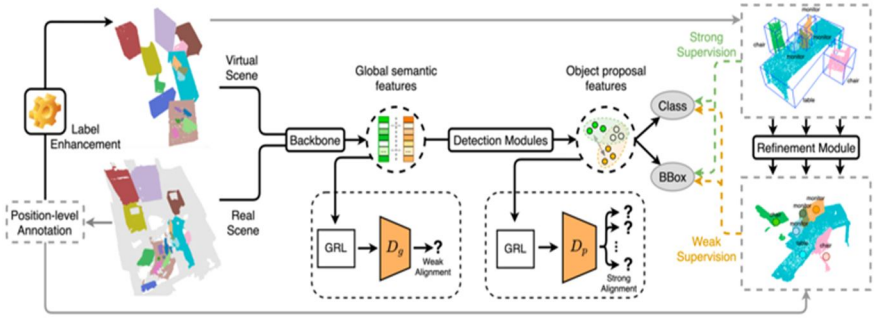


Figure 2. The framework of the *Corporeality (BR)* approach. Given real scenes with position-level based annotations, first the weak labels are enhanced to get fully-annotated virtual scenes. Then the real scenes and virtual scenes are fed into the detector, trained with weakly-supervised and fully-supervised detection loss respectively. During training the precise object centres are used in a virtual scene to refine the imprecise centres in real scenes. Strong-weak adversarial domain adaptation method is utilized to align the distributions of features from both domains. The global discriminator outputs judgments for each scene, and the proposal discriminator outputs judgments for each object proposal. (Here GRL refers to gradient reversal layer; Dg and Dp stand for the global and proposal discriminators respectively.)

A. Position-level Based Annotations

Since selection a point in three-dimensional space is difficult to operate, the algorithm divides the labelling process into two steps: first the centres of an object in a proper two-dimensional view of the scene is labelled, then the line that goes through this centre is calculated along with the focal point of the camera as per the camera parameters of the two-dimensional view. Secondly a point on the line is chosen to determine the object's centre in a three-dimensional space. This strategy requires less than 5 seconds to label an instance, and the labelling error margin can be controlled within 10% of the instance size.

B. Shape-guided Label Enhancement

While position-level based annotations require far less labelling time, its information loss however is severe, which is manifested in two aspects:

1) The information of the objects' sizes is lost

2) The object centres are imprecise.

Despite this, position-level based annotations can provide a coarse layout of the scenes. By assembling synthetic three-dimensional shapes according to the layout, weak labels are enhanced and generate accurately-annotated virtual scenes where sizes are available and centres are precise. The label enhancement method has two-step:

a) Calculation of the basic properties of three-dimensional shapes

b) Placement of these shapes to generate physically reasonable virtual scenes from the labels.

- *Definition of Shape Properties*: Given a synthetic three-dimensional shape, which is represented as $O \in R^{N\times 3}$, its safe to assume that its axis-aligned and normalized into a unit sphere. The length, width and height of O is defined as l, w and h. Then these shapes are categorised into the following three classes: supporter, stander and supportee. Supporters and standers are objects that can only be supported from the ground, with the only difference being that standers are not likely to support other things. The third category is supportees. So, if a shape belongs to supporter, the following three properties are calculated: minimum-area enclosing rectangle (MER^*), supporting surface height (SSH^*) and compactness of the supporter surface (CSS^*). The MER is computed in XY plane, which is the minimum rectangle enclosing all the points of the shape. The SSH is the height of the highest surface on which other objects can stand. CSS is a boolean value, indicating whether the supporting surface can be approximated by the MER.
- *Virtual Scene Generation*: This algorithm utilizes a three-stage approach to construct the virtual scenes, which is equivalent to generating the position of each shape stage by stage:

➢ First the coarse layout provided is refined by using position-level based annotations and then the initial positions are generated

➢ Then the gravity-aware positions are generated by restoring the supporting relationships between the objects

➢ Finally generate collision-aware positions are generated to make the virtual scenes physically reasonable. This pipeline has been illustrated in ***Figure 3***.

To generate the *initial positions*, this model needs to recover a more precise layout based on the geometric information available from the scenes. Given a scene is in a mesh format, first the meshes are over-segmented using a normal-based graph cut method [11, 15]. The result is a segment graph, where the nodes indicate segments and the edges denote adjacent relations. Then for horizontal* segments where the area* is larger than 0.1 m^2 and height* is larger than 0.1 m, the model iteratively merges its neighbours into them if the height difference between the horizontal segment and the neighbouring segment is smaller than 0.02 m.

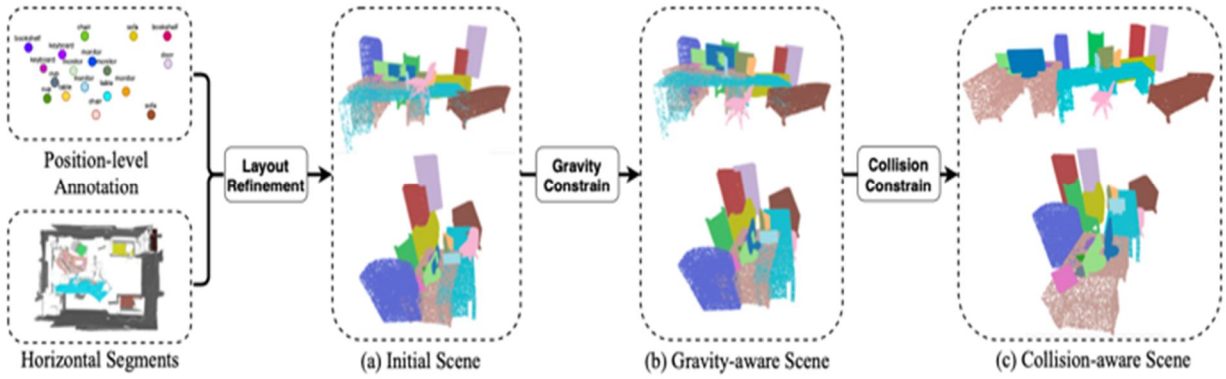


Figure 3. The pipeline of our three-stage virtual scene generation method. We first extract horizontal segments from the mesh data and use them to refine the coarse layout provided by position-level annotations. Then synthetic 3D shapes are placed in virtual scenes according to the new layout to construct initial virtual scenes. After that we apply gravity and collision constraints on the virtual scenes to restore the lost physical relationships between objects and make the scenes more realistic.

Once merged, the segments are considered as a whole and the height of the newly merged segment is set to be the same as the original horizontal segments height. Post merge, each horizontal segment is represented by its MER. If only one supporter's centre falls in a MER, the MER is assigned to that particular supporter. When the centres of multiple supporters fall under the same MER, a K-means clustering of the horizontal segments is performed pertaining to all these centres and the MER for each supporter is calculated respectively.

Then the three-dimensional shapes are placed within their corresponding categories based on the centre given by position-level based annotations and then the horizontal segments are utilized to refine the layout. The initial positions of these shapes is represented by a dictionary, where the key is an instance index and the value is a list:

$[(x, y, z), (s_x, s_y, s_z), O, \theta, S, M, H]$……(1)

where the instance index is an integer ranging from 1 to the total number of objects in the scene.

(x, y, z) denotes the centre coordinates.

(s_x , s_y , s_z) indicates the scales in three-dimensions. **θ** is the rotation angle of the shape.

S tells whether the shape is a supporter.

M and H indicate the MER and SSH of supporter. They are set to None when S is false.

If the shape has been assigned a horizontal segment, then the MER of that segment is used to initialize the above parameters. That is, a supporter whose CSS is True is chosen and made the MER of this supporters overlap with the horizontal segments. Otherwise a random initialization is conducted. If only point cloud data is available, a simple random initialization is performed and the subsequent stages are the similar.

Next, the initial positions to generate *gravity aware positions* is traversed. In this process change to z and SSH in the position dictionary is necessary. For supporters and standers, their bottoms are directly aligned with the ground (i.e. the XY plane). For a supportee, if its (x, y) falls anywhere within the supporter's MER, it is assigned to the nearest supporter and its bottoms are also aligned with the supporting surface. Otherwise, it is aligned to the ground.

The shapes are then moved to acquire *collision-aware positions*. During this stage only x and y coordinates in the position dictionary will be modified. First the objects on the ground are moved, then the supported ones on which these shapes will be moving together (if there are any) are moved. Then for each supporter, the corresponding supportees are moved until there is no more overlap. Note that these three generation stages can not only make the virtual scenes more realistic, but also weaken the impact of imprecise centre labels. Thus the virtual scene generation method is robust at labelling and pointing out errors.

Finally, the collision-aware positions are converted to point clouds with proper density. As larger surfaces are more likely to be captured by the sensor, the use of **(ls_x)(wsy), (wsy)(hs_z)** and **(ls_x)(hs_z)** is maximised to approximate the surface area of these shapes. Then the number of points for each object is set proportional to their surface areas using uniform sampling, the largest one remaining N points.

C. Virtual2Real Domain Adaptation

Although the label enhancement approach is able to generate physically reasonable and fully-annotated virtual scenes, there is still a huge domain gap between them and the real scenes (e.g. backgrounds like walls are missed in the virtual scenes). Therefore, mining useful knowledge is necessary in these perfect virtual labels to make up for the information loss of position-level based annotations, rather than just relying on the virtual scenes.

The virtual scenes and real scenes are referred as source domain and target domains respectively. A virtual-to-real adversarial domain adaptation method is utilized to solve the above problem, where the overall objective is:

$\max\min J = L_{sup}(O) - L_{adv}(O, D)$ DO

$= (L_1 + L_2 + L_3) - (L_4 + L_5)$ …… (2)

where **O** refers to the object detection network (detector) **D** indicates the discriminators used for adversarial feature alignment.

L_{sup} aims to minimize the differences between the predicted bounding boxes and the annotations, which can be further divided into the loss for centre refinement module (L_1), fully-supervised detection loss on source domain (L_2) and weakly-supervised detection loss on target domain (L_3). The objective of L_{adv} is to align the features from source domain and target domain, which aims to utilize the knowledge learned from source domain to assist object detection in target domain. L_{adv} can be divided into global feature alignment loss (L_4) and proposal feature alignment loss (L_5). The following explains these loss functions and the network in detail.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

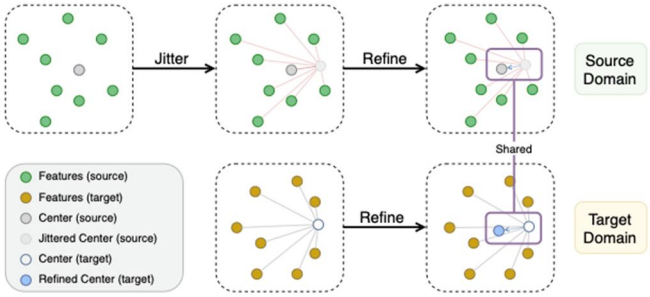


Figure 4. Demonstration of our centre refinement method. We first jitter the centre labels in source domain, and utilize a PointNet-like module to predict the centre offset from the local graph of the jittered centres. This module can be directly utilized to predict the centre error in target domain as the global semantic features from the two domains have been aligned.

As shown in ***Figure 2***, the detector is divided into three blocks: a backbone which extracts global semantic features from the scene, a detection module which generates object proposals from the semantic features, and a prediction head which predicts the semantic label and bounding box from each object proposal feature.

During training, the imprecise centre labels in target domain are jointly refined and the predictions of the detector are then supervised. As shown in ***Figure 4***, the centre labels in source domain are jittered by adding noise within 10% of the objects' sizes to imitate the labelling error in target domain. Then for each jittered centre, its k nearest neighbors in the three-dimensional euclidean space are queried from the global semantic features to construct a local graph, and predict the centre offset through a PointNet-like module:

$p(c) = MLP_2 \{ \max i \in N(c) \{ MLP_1[f_i; c_i - c] \} \}$ ……. (3)

where **p** denotes the PointNet-like module, **c** indicates the jittered centre label, **N(c)** is the index set of the k nearest neighbours of c, $\mathbf{f_i}$ is the global semantic feature, whose coordinate is $\mathbf{c_i}$, and ***max*** refers to the channel-wise max-pooling.

L_1 is set as the mean square error between the ground-truth centre offset and p(c). Then for a fully-supervised training, the detection loss L_2 is the same as the loss utilized in the original method. For weakly-supervised training, p is utilised to predict the centre error in target domain and acquire refined centre labels. L_3 is set as a simpler version of L_2 which ignores the supervision for box sizes. More details about L_3 can be found in supplemental. Then the $L_{adv}(O,D)$ is analysed. A feature alignment is conducted in an adversarial manner: the discriminator predicts which domain the feature belongs to, and the detector aims to generate features that are difficult to discriminate. The sign of gradients is reversed by a gradient reversal layer [12]. As the virtual scenes and real scenes are processed by the same network, it is presumed that L_3 helps the network learn how to locate each object in real scenes, and L_2 compensates for the information loss of centres and sizes. However, due to the domain gap, L_2 will introduce domain-specific knowledge of the virtual scenes, which impair the influence introduced by L_3. Besides, the centre refinement module is trained only on source domains, which may not perform satisfactorily on target domains. Therefore, the global semantic features and object proposal features are aligned with L_4 and L_5 respectively. Inspired by [36], the features are aligned with different intensities at different stages. For global semantic features, PointNet is used to predict the domain label. Focal loss [19, 36] is utilized to apply weak alignment:

$$L_4 = -\sum_{i=1}^{B} (1-p_i)^{\gamma} \log(p_i), \gamma>1 \quad \text{……(4)}$$

where **B** is the batch size, and

$\mathbf{p_i}$ refers to the probability of the global discriminator's predictions on the corresponding domain.

Features with high p are easy to judge, which means they are domain-specific features and forcing invariance on them can negatively impact their performance. So a small weight is used to reduce their impact on the training dataset. For object proposal features, they will be directly taken to predict the properties for bounding boxes. As the properties are domain-invariant and have real physical meaning, we strongly align this stage of features using an objectness weighted L2 loss:

$$L_5 = \sum_{i=1}^{B} \sum_{j=1}^{N} s_{ij} (1-p_{ij})^2 \quad \text{……..…. (5)}$$

where **B** is the batch size,

N is the number of proposals,
sij refers to the objects label and
pij is the probability of the proposal discriminator's predictions on the corresponding domain.
Details of the architecture of centre refinement module and discriminators are presented supplementarily.

IV. EXPERIMENT

This section, specifies the details of conducted experiments that show the effectiveness of the ***Corporeality (BR)*** approach. The first section described the datasets and their respective experimental settings. Then the generated virtual scenes are evaluated and the detection results are reported by the method. This section also demonstrates the design experiments that showcase the robustness of the virtual scene generation method and advocate the practicality of this approach.

	Property	Bath-tub	Bed	Bench	Book-shelf	Bottle	Chair	Cup	Cur-tain	Desk	Door	Dresser	Key-board	Lamp	Laptop	Monitor	Night-stand	Plant	Sofa	Stool	Table	Toilet	Ward-robe
# train	Object Number	113	308	58	786	234	4357	132	408	551	2028	174	193	376	86	574	190	293	406	315	1526	201	98
# validate		31	81	21	234	41	1368	34	95	127	467	43	53	83	25	191	34	50	97	51	407	58	19
# real	Point Number	2941	3905	1015	2679	101	726	66	2919	1525	1110	1274	74	272	173	370	700	792	2718	525	1282	1445	2762
# virtual		6891	8683	4097	6258	162	2135	91	5495	5004	6048	2703	480	609	343	939	1088	1249	7250	1391	5421	3716	6105

Table 2. Number of objects in each category in the training set and validation set of ScanNet, and average number of points of objects in each category in the real scenes and the virtual scenes.

	Setting	batht.	bed	bench	bsf.	bot.	chair	cup	curt.	desk	door	dres.	keyb.	lamp	lapt.	monit.	n.s.	plant	sofa	stool	table	toil.	ward.	mAP@0.25
VoteNet	FSB [27]	66.8	86.2	24.4	**55.6**	0.0	88.3	0.0	48.5	62.8	45.8	24.1	0.1	47.2	5.2	62.1	73.2	13.4	88.7	35.1	62.6	94.6	7.8	45.1
	WSB	21.9	46.9	0.3	2.3	0.0	53.7	0.0	0.9	32.1	1.0	6.6	0.1	0.2	0.1	1.8	53.6	0.1	57.0	4.6	6.4	19.7	0.0	14.1
	WS3D† [23]	22.0	58.5	10.3	5.8	0.0	60.4	0.0	4.1	26.7	3.2	1.6	0.0	14.0	0.6	18.6	46.3	0.4	32.7	11.8	23.5	65.0	0.0	18.4
	$WSBP_P$	43.2	58.0	2.4	16.1	0.0	75.1	0.7	7.9	54.2	6.4	7.1	2.3	35.2	18.4	12.8	64.0	4.4	68.5	20.2	22.0	71.6	5.2	27.1
	$WSBP_M$	45.0	49.6	5.5	18.5	0.0	62.7	2.9	11.4	49.6	6.9	2.5	1.0	30.0	7.6	21.4	64.8	7.3	79.6	23.1	35.2	80.9	2.2	27.6
	BR_P(Ours)	51.2	73.0	16.4	27.1	0.1	70.3	0.0	8.3	44.5	7.3	16.0	1.5	40.2	7.7	42.1	50.8	7.4	67.1	10.7	39.0	88.4	18.1	31.2
	BR_M(Ours)	57.1	80.4	14.3	31.7	0.0	77.4	0.0	13.2	49.7	11.3	14.8	1.0	43.5	6.0	56.5	65.0	10.6	80.2	26.9	44.2	91.4	6.5	35.5
GroupFree3D	FSB [22]	**86.2**	87.5	16.3	49.6	0.6	**92.5**	0.0	**70.9**	**78.5**	**53.5**	**56.0**	6.4	**68.2**	11.5	**81.5**	**88.5**	15.2	88.2	**45.6**	**65.0**	**99.7**	**31.2**	**54.2**
	WSB	75.0	75.7	4.3	17.2	0.0	81.4	0.0	3.5	34.0	4.7	3.2	2.1	46.6	3.3	45.8	52.8	8.3	71.0	15.7	18.1	90.8	0.7	29.7
	WS3D† [23]	71.9	78.3	0.9	20.2	0.8	79.2	1.0	2.9	47.6	7.7	10.6	19.2	41.6	13.5	65.6	41.2	0.8	74.6	17.7	26.3	88.9	1.7	32.4
	$WSBP_P$	71.9	77.1	7.7	25.2	**3.0**	80.6	0.4	3.2	50.1	10.5	36.3	17.0	52.9	**30.3**	59.9	63.8	9.6	78.2	28.4	25.3	93.3	14.4	38.2
	$WSBP_M$	81.8	82.6	0.0	35.0	0.0	77.5	0.4	27.1	38.4	7.6	22.3	9.7	44.3	24.4	65.4	76.5	5.5	62.4	34.7	28.7	**99.7**	5.4	37.7
	BR_P(Ours)	72.3	73.5	**45.8**	27.7	0.0	77.2	**8.2**	30.8	35.0	17.8	51.7	0.3	64.2	25.0	63.5	66.6	23.8	86.7	33.9	37.6	98.3	5.2	43.0
	BR_M(Ours)	85.3	**90.9**	8.8	34.3	1.9	80.0	7.7	24.7	58.0	20.8	45.4	**31.3**	64.4	25.8	67.5	76.7	**27.3**	**91.4**	43.3	46.7	94.8	8.3	47.1

Table 3. The class-specific detection results (mAP@0.25) of different weakly-supervised methods on ScanNetV2 validation set. (FSB is the fully-supervised baseline. † indicates the method requires a small proportion of bounding boxes to refine the prediction. Other methods only use position-level annotations as supervision. We set best scores in bold, runner-ups underlined.)

A. *Experiments Setup*

1) *Datasets:* ModelNet40 [45] was chosen as the dataset for synthetic three-dimensional shapes. ModelNet40 harbours 12,311 synthetic CAD models from 40 different categories, split into 9,843 for training and 2,468 for testing. Experiments were performed on ScanNetV2 [9] dataset. ScanNet is a richly annotated dataset comprising of indoor scenes with 1201 training scenes and 312 validation scenes. For each object appearing in those scenes, ScanNetV2 officially provides the objects corresponding class in ModelNet40. Therefore 22 categories of Model- Net40 were chosen which comprise of more than 50 objects in the ScanNetV2 training set and 20 in the validation set, and reports the detection performance of each. Since ScanNet does not provide human-labelled bounding boxes, axis-aligned bounding boxes were predicted and used to evaluate the predictions returned by the validation set as in [22, 27, 46, 50]. This named as the benchmark ScanNet-md40. Compared to the 18-category setting in previous attempts [22, 27, 46], the ScanNet-md40 benchmark is actually more challenging. Along with involving large object categories (e.g. desks and bathtubs), the end goal is to also effectively detect relatively small objects, such as laptops, keyboards and monitors. It is expected that the benchmark is capable of better evaluating the performance of both detectors as well as the weakly-supervised learning methods.

2) *Compared Methods:* To illustrate the effect of the Corporeality approach, the popular VoteNet [27] and state-of-the-art GroupFree3D [22] were chosen as detectors. Corporeality was then compared with the following settings:

a) *FSB:* a fully-supervised baseline, which serves as the upper bound for weakly-supervised methods;

b) *WSB:* a weakly-supervised baseline, which trains the detector on real scenes by using L_3 only;

c) *WS3D:* a position-level based weakly-supervised approach proposed in [23], which makes use of a number of precisely annotated bounding boxes;

d) *WSBP:* WSB pretrained on the virtual scenes.

For settings which require virtual scenes, experiments were conducted on two versions of virtual scenes (from points/meshes), which are distinguishable basis the subscripts M and P respectively.

3) *Implementation Details:* The following values were set - $N = 10000$, $k = 16$ and $\gamma = 3$. During training, as real scenes become more complicated, the convergence of L_3 is much slower than L_2. Hence multiplied L_2 by 0.1 to slow down the training based on virtual scenes and this helps stabilize the process of feature alignment. In order to effectively train the centre refinement module, the global semantic features should not change rapidly. Therefore Corporeality is first trained without L_1 until convergence, and then used with the net information loss function to fine-tune the network. In case of GroupFree3D, there are several decoders and each individual decoder outputs a stage of proposal features, and consequently feature alignment is performed on the last stage only. Different compared to previous works [22, 27], this setting needs to be able to detect small objects, such as bottles, cups and keyboards. As it is difficult for the network to extract high-quality features of these objects, an augmentation strategy is employed to alleviate the problem, which is similar to [16]. Please refer to the supplementary attached herewith for additional details.

B. *Results and Analysis*

1) *Virtual Scene Evaluation:* First the statistics of the generated virtual scenes are evaluated by calculating the average number of points of each individual objects in each category exist in the real scenes and virtual scenes. As the input point clouds are sampled down to a given number before being fed into the network, only the ratio of the average point numbers of objects in each category are if interest as the numbers can be controlled by the sampling down the scale. Results are presented in *Table 2*. It shows that the ratio in the virtual scenes is similar to that of the real scenes, which therefore implies that the statistics of the virtual scenes are reasonable enough. Qualitative visualizations are shown to demonstrate the scene generation method in *Figure 5*. The virtual scenes generated with mesh information are named mesh-version virtual scenes.

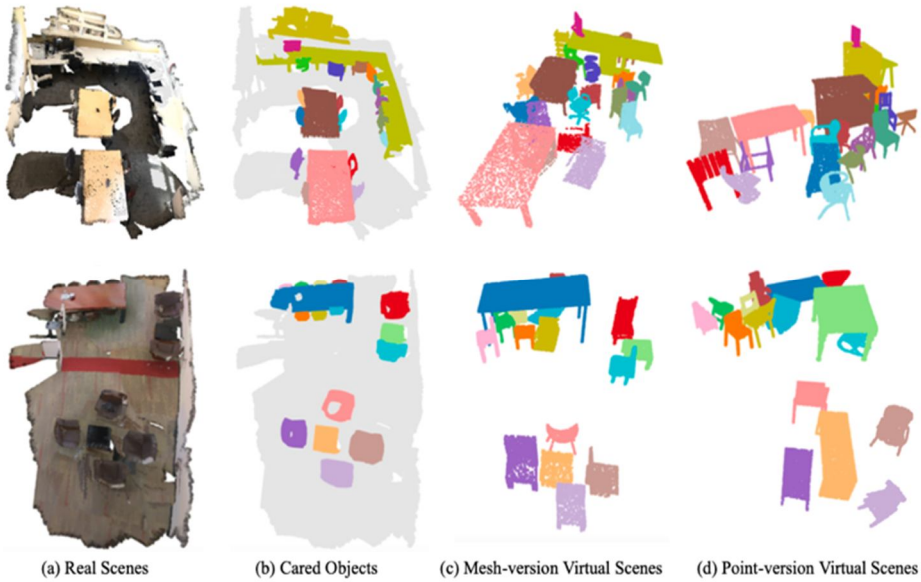


Figure 5. The qualitative visualization results of our virtual scene generation. In (b), (c) and (d), the same colour indicates the same object. Gray points are floors, walls and objects that we do not care. It can be seen that the virtual scenes preserve the coarse scene context and the supporting relationships between objects.

Otherwise they are named point-version virtual scenes. It is shown that a mesh-version virtual scene can largely preserve the layout of the real scene, and the point-version ones can successfully combine the individual three-dimensional shapes in a meaningful way.

Three-dimensional Object Detection Results: As shown in *Table 7*, with position-level based annotations only, WSB reduces the detection accuracy by a large margin in terms of mAP@0.25 compared to FSB. That's mainly because WSB fails to learn the ability of predicting precise the centres and sizes of the bounding boxes according to the scene context. WS3D makes use of some box annotations to a achieve better performance. However, as it is specially designed for outdoor three-dimensional object detection, WS3D is still far from satisfactory when coping with the complicated indoor scenes. With pretraining on the virtual scenes, WSBP has more than 8% improvement over the WSB. That shows the ability of predicting precise bounding boxes learned in the source domain has been successfully transferred to the target domain. With our domain adaptation method to conduct better transfer- ring, the improvement over the WSB is boosted to a higher level. The above results shows each step in *Corporeality (BR)* is necessary: the virtual scenes are helpful to boost the detection performance, and the domain adaptation method can further explore the potential of the virtual scenes. Interestingly, as the virtual scenes become more realistic (from point-version to mesh-version), the performance of *Corporeality (BR)* improves a lot while WSBP has little change, which indicates that layout may not be that important in pretraining as in domain adaptation.

In terms of class-specific results, on some categories the mAP@0.25 of the BRM (for GroupFree3D) is even the highest among all the methods including the FSB. However, all methods fail to precisely detect cup and bottle, which shows current three-dimensional detectors still face huge challenges in small object detection. More detection results (mAP@0.5) can be found in supplementary.

Robustness for Labeling Error: In our labelling strategy, the centre error is within 10%, which we define as the error rate, of the object's size. To show the robustness of our approach, we gradually increase this rate from 10% to 50% by randomly jittering the centres according to the box sizes, and report the detection results of WSB and BRM (for GroupFree3D) in terms of mAP@0.25. As shown in *Table 4*, with the increasing of error rate, the performance of *Corporeality (BR)* degrades more slowly than WSB. Even if the error rate is 50%, which allows us to label the centres in a more time-saving strategy, *Corporeality (BR)* can still achieve satisfactory results (higher than 0.41 in terms of mAP@0.25).

Visualization Results: We visualize the detection results of WSB and BRM (for GroupFree3D) on ScanNetV2. As shown in *Figure 6*, *Corporeality (BR)* can produce more accurate detection results with less false positives. The visual results further confirm the effectiveness of the proposed method.

C. Ablation Study

Further design ablation experiments were made to study the influences of each scene generation step and each domain adaptation loss to the performance of our ***Corporeality (BR)*** approach. In this section, **VoteNet** is adopted as the detector and use point-version virtual scenes for universality.

Table 5, illustrates that in the virtual scene generation pipeline, the physical constraints and density control are effective.

As the virtual scenes become more realistic, the performance of our *Corporeality (BR)* approach is getting better.

Table 4. The detection results (mAP@0.25) of *Corporeality (BR)* represented in the table below under different error rate for centre labelling on ScanNetV2. GroupFree3D was adopted as the detector and utilize mesh-version virtual scenes for *Corporeality (BR).*

Method	Error Rate				
	10%	20%	30%	40%	50%
WSB	29.7	26.8	25.0	22.3	19.7
BR_M (Ours)	**47.1**	**46.0**	**43.9**	**43.1**	**41.2**

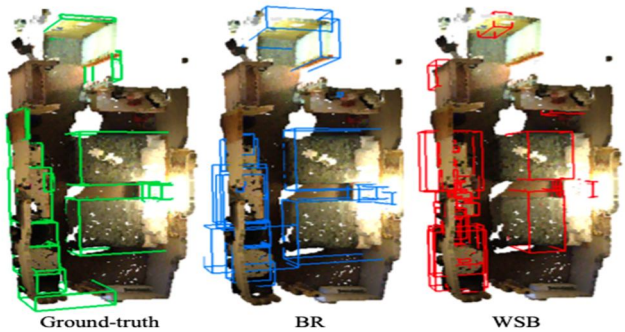


Figure 6. Visual Results on ScanNetV2.

We compare Corporeality (BR) and WSB with the ground-truth bounding boxes.

As shown in *Table 6*, we show the effect of each domain adaptation module and the centre refine module. It can be seen that with global alignment or object proposal alignment, the detection performance can be boosted by 3.5% and 2.2% respectively. By combining the two kinds of feature alignments, we are able achieve higher detection accuracy. Then after applying the centre refinement method, the performance is further boosted by 1.0%.

Gravity Constrain	Collision Constrain	Density Control	mAP@0.25
			26.3
✓			27.2
✓	✓		28.5
✓	✓	✓	**31.2**

Table 5. The detection results (mAP@0.25) of BR with virtual scenes at different generation stages on ScanNetV2. Here the detector is VoteNet and the virtual scenes are point-version.

Global Alignment	Proposal Alignment	Center Refinement	mAP@0.25
			24.2
✓			28.7
	✓		27.4
✓	✓		30.2
✓	✓	✓	**31.2**

Table 6. The detection results (mAP@0.25) of BR with different domain adaptation modules on ScanNetV2. Here the detector is VoteNet and the virtual scenes are point-version.

D. Limitation

Due to the limited number of categories in Model- Net40, we selectively evaluate the performance of ***Corporeality (BR)*** on 22 classes. However, as online repositories of user-generated three-dimensional shapes, such as the three-dimensional Warehouse repository [3], contain three-dimensional shapes in almost any category, ***Corporeality (BR)*** can be easily extended to three-dimensional object detection on more classes once these online synthetic shapes are organized into a standard dataset. Therefore, ideally we can leverage a larger synthetic three-dimensional shape dataset, which covers all objects that may appear in indoor scenes. This dataset can promote more researches on three-dimensional scene understanding with synthetic shapes, which we leave for future work.

V. CONCLUSION

In this paper, we have proposed a new label enhancement approach, namely ***Corporeality (BR),*** for three-dimensional object detection trained using only object centres and class tags as supervision. To fully explore the information contained in the position-level based annotations, we regard them as the coarse layout of scenes, which is utilized to assemble three-dimensional shapes into fully-annotated virtual scenes. We apply physical constraints on the generated virtual scenes to make sure the relationship between objects is reasonable. Then in order to make use of the virtual scenes to remedy the information loss from box annotations to centres, we present a virtual-to-real domain adaptation method, which transfers the useful knowledge learned from the virtual scenes to real-scene three-dimensional object detection. Experimental results on **ScanNet** dataset show the effectiveness of our ***Corporeality (BR)*** approach.

REFERENCES

[1] Open3d: A modern library for 3d data processing. [EB/OL]. http://www.open3d.org/. 11
[2] Opencv. [EB/OL]. https://opencv.org/. 10
[3] Trimble 3d warehouse. [EB/OL]. http : / / 3dwarehouse.sketchup.com/. 8
[4] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In CVPR, pages 2614–2623, 2019. 2
[5] Armen Avetisyan, Angela Dai, and Matthias Nießner. End- to-end cad model retrieval and 9dof alignment in 3d scans. In ICCV, pages 2551–2560, 2019. 2

[6] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object de- tection for autonomous driving. In CVPR, pages 2147–2156, 2016. 2
[7] 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
[8] Manuel Dahnert, Angela Dai, Leonidas J Guibas, and Matthias Niessner. Joint embedding of 3d scan and cad ob- jects. In ICCV, pages 8749–8758, 2019. 2
[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Hal- ber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In CVPR, pages 5828—-5839, 2017. 2, 6, 13
[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In ICCV, pages 2758–2766, 2015. 2
[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Effi- cient graph-based image segmentation. IJCV, 59(2):167– 181, 2004. 3
[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In ICML, pages 1180–1189, 2015. 5
[13] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d seman- tic instance segmentation of rgb-d scans. In CVPR, pages 4421–4430, 2019. 1, 2
[14] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In CVPR, pages 15587–15597, 2021. 2
[15] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. Object discovery in 3d scenes via shape analysis. In ICRA, pages 2088–2095, 2013. 3
[16] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small ob- ject detection. arXiv preprint arXiv:1902.07296, 2019. 6, 12
[17] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object de- tection in rgb-d images. In ICCV, pages 4622–4630, 2017. 2
[18] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In CGF, volume 34, pages 435–446, 2015. 2
[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dolla´r. Focal loss for dense object detection. In ICCV, pages 2980–2988, 2017. 5
[20] Or Litany, Tal Remez, Daniel Freedman, Lior Shapira, Alex Bronstein, and Ran Gal. Asist: automatic semantically in- variant scene transformation. CVIU, 157:284–299, 2017. 2
[21] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In CVPR, pages 529–537, 2019. 2
[22] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. arXiv preprint arXiv:2104.00678, 2021. 1, 2, 6, 11, 13
[23] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In ECCV, pages 515–531, 2020. 1, 2, 6, 12, 13
[24] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Yunde Jia, and Luc Van Gool. Towards a weakly su- pervised framework for 3d point cloud object detection and annotation. TPAMI, 2021. 1, 2
[25] LiangliangNan,KeXie,andAndreiSharf.Asearch-classify approach for cluttered indoor scene understanding. TOG, 31(6):1–10, 2012. 2
[26] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In ECCV, pages 523–540, 2020. 2
[27] 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. 1, 2, 6, 11, 13
[28] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and LeonidasJ Guibas. Frustum pointnets for 3d object detection from rgb-d data. In CVPR, pages 918–927, 2018. 2
[29] 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. 1, 2
[30] 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. 1, 2, 11
[31] Zengyi Qin, Jinglu Wang, and Yan Lu. Weakly supervised 3d object detection from point clouds. In ACM MM, pages 4144–4152, 2020. 2
[32] Yongming Rao, Benlin Liu, Yi Wei, Jiwen Lu, Cho-Jui Hsieh, and Jie Zhou. Randomrooms: Unsupervised pre- training from synthetic shapes and randomized layouts for 3d object detection. In ICCV, pages 3283–3292, 2021. 2
[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In CVPR, pages 779–788, 2016. 1
[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. TPAMI, 39(6):1137–1149, 2016. 1
[35] Zhongzheng Ren, Ishan Misra, Alexander G Schwing, and Rohit Girdhar. 3d spatial recognition without spatially labelled 3d. In CVPR, pages 13204–13213, 2021. 1, 2
[36] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In CVPR, pages 6956–6965, 2019. 5
[37] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointr- cnn: 3d object proposal generation and detection from point cloud. In CVPR, pages 770–779, 2019. 1, 2
[38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In CVPR, pages 567–576, 2015. 1
[39] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In ECCV, pages 634–651, 2014. 2
[40] Shuran Song and Jianxiong Xiao. Deep sliding shapes for a model 3d object detection in rgb-d images. In CVPR, pages 808–816, 2016. 2
[41] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets im- prove semi-supervised deep learning results. arXiv preprint arXiv:1703.01780, 2017. 2
[42] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-aware 3d model embedding and retrieval. In ECCV, pages 397–413, 2020. 2
[43] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi- supervised 3d object detection. In CVPR, pages 14615– 14624, 2021. 2
[44] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In CVPR, pages 2642–2651, 2019. 2

[45] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lin- guang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, pages 1912–1920, 2015. 6
[46] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level con- text votenet for 3d object detection. In CVPR, pages 10447– 10456, 2020. 6
[47] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre- training for 3d point cloud understanding. In ECCV, pages 574–591, 2020. 2
[48] Ning Xu, Yun-Peng Liu, and Xin Geng. Label enhancement for label distribution learning. TKDE, 2019. 2
[49] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. arXiv preprint arXiv:2101.02691, 2021. 2
[50] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primi- tives. In ECCV, pages 311–329, 2020. 6
[51] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self- ensembling semi-supervised 3d object detection. In CVPR, pages 11079–11087, 2020. 2
[52] Xingyi Zhou, Dequan Wang, and Philipp Kra ̈henbu ̈hl. Ob- jects as points. arXiv preprint arXiv:1904.07850, 2019. 1
[53] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In CVPR, pages 4490–4499, 2018. 1, 2

SUPPLEMENTARY MATERIAL

A. Overview

This supplementary material[4] is organized as follows:

** Section 1 details the Approach section in the main paper.
** Section 2 shows the implementation detail of WS3D.
** Section 3 details our augmentation strategy for small objects during training.
** Section 4 shows more experimental results.

B. Approach Details

In this section, we show the details in our approach, which is divided into shape-guided label enhancement and virtual2real domain adaptation.

B.1. Label Enhancement

We show the exact definitions of some concepts appeared in Section 3.2 of the main paper as below.

Shape Properties: The MER is computed in XY plane, which is the minimum rectangle enclosing all the points of the object template. The SSH is the height of the largest surface on which other objects can stand. The CSS is a boolean value, indicating whether the supporting surface is similar with the MER (i.e. we can use the MER to approximate the supporting surface if CSS is true).

In order to calculate MER, we use the OpenCV [2] tool- box to calculate the MER of two-dimensional point set. As OpenCV cannot be directly utilized to process point clouds, we first project the object templates to XY plane to acquire two-dimensional point sets. Then we calculate the MER of a point set

$S = \{ (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \}$ as below :

$(x, y, l, w, \theta) = \text{minAreaRect}(1000 * S)$ ……(6)

$\text{MER} = (x, y, l/1000, w/1000, \theta)$ …… (7)

where minAreaRect is a function in OpenCV, which takes integer 2D point set as input and returns a rectangle, and rectangle is represented by a quintuple (x, y, length, width, θ), which indicates the centre coordinates, length, width and rotation angle of a rectangle.

1000 * S means that we multiply all the coordinates in S by 1000 and then convert the coordinates from float to integer, which can reduce the rounding error.

To compute SSH, we first utilize Open3D [1] to get the normal values of each point from point cloud. Then if the normal of a point is almost vertical (i.e. the normal values length along Z-axis is greater than 0.88), we record the coordinate of this point. After traversing all the points, we have recorded a list of coordinates. We sort the list according to the Z coordinate in ascending order, and the list of sorted Z coordinate is named as l_z.Then get a slice of l_z from index$\lfloor \frac{4}{5} len_z \rfloor$ to $\lfloor \frac{9}{} len_z \rfloor$, where len_z denotes the length of l_z. SSH can be 10 calculated by averaging this slice. Note that this algorithm suppose the supporter has a large supporting surface on its top, and it can tolerate 10% points higher than this surface.

To calculate CSS , we collect points which satisfy SSH– ^{1} h<z<SSH+ ^{1} h from the given object template, where h is the height of this object template. Then we project these points to XY plane and name them supporter points PS. If PS can almost fill the MER, the CSS is set to be T rue. To analyse the compactness, we use K-means algorithm to divide PS into 2 clusters: PS1 and PS2. Then we calculate the area of convex hull of PS1 and PS2. The area is computed by using OpenCV:

A = contourArea (convexHull (1000 ∗ P))/ 1000000
……. (8)
where contourArea and convexHull are functions in OpenCV,Pisa2DpointsetandAistheareaofP.The areas for PS1 and PS2 are A1 and A2 respectively. So we can compute CSS as below:

$$CSS = \begin{cases} True, & A_1 + A_2 > 0.9 * l * w \\ False, & otherwise \end{cases} \qquad (9)$$

where l and w are the length and width of the MER of this object template.
Segment Properties: Next we provide the definitions of horizontal segment, the area of segment and the height of segment. For a segment, we define z as the Z coordinate of all the points on it. Then if $|maximum(z) - median(z)| < 0.2$ or $|minimum(z) - median(z)| < 0.2$, we consider this segment is horizontal. To calculate the area of segment, we directly utilize (8) and take all points on the segment as input (ignore the Z coordinates of points). To compute the height of a segment, we follow the same procedure as computing SSH : we first calculate the normal values and pick out points with normal values that are almost vertical, and then we pick out the Z coordinates of these points and acquire a list l_Z . The segment's height is defined as the mean of l_Z .
B.2. Domain Adaptation
We first provide detailed definition of L3. Then we show the architectures of our centre refinement module and the two discriminators.
For weakly-supervised training, as only objects' centres and semantic classes are available, we set **L3** as a simpler version of **L2**:
L3 = Lf + Li, Lf = Ls + Lo + Lc ……(10)
Lf is used to supervise the final prediction, where Ls and Lo are the cross entropy losses for semantic labels and objectness scores, and Lc is defined as:

$$L_c = \sum_i max(||C_{gi} - C_i||_2 - \lambda S_{gi}, 0) \qquad (11)$$

which denotes the hinge loss for centres. Ci is the i-th predicted centre, Cgi is the nearest ground-truth centre to Ci, and Sgi indicates the average size for the semantic class of this object. We set λ = 0.05 to approximate the labelling error of centres. For Li, we only make use of the centre coordinates to weakly supervise the intermediate process of training. For example, in VoteNet [27], the detection module predicts votes from the semantic features and aggregate them to generate object proposals, in which voting coordinates are the intermediate variables need to be super- vised. Here we utilize the Chamfer Distance between the voting coordinates and the ground-truth centre coordinates to supervise the voting. In GroupFree3D [22], the detection module utilize KPS to sample the semantic features and generate initial object proposals, where the sampled points require supervision. Originally the KPS operation requires us to sample the nearest k points to the object centre from the point cloud belong to this object. However, we weaken this requirement and sample the nearest k points without any constraints.
For the centre refinement module, we adopt the Set Abstraction (SA) layer [30] to extract features from the local KNN graph. Then an MLP is utilized to predict centre offset from the feature. The SA layer first concatenates the relative coordinates between the centre and its neighbours to the features of the neighbours, which is followed by a shared –
MLP (MLP (256, 128) ^{5}) and a channel-wise max-pooling layer. The pooled feature contains the local information of the centre, which is then concatenated with the one-hot vector of the centre's semantic class (we name the feature after concatenation as centre feature). We utilize another MLP (MLP (64, 3)) to predict the centre offset from the centre feature. For the global and proposal discriminators, we show their architectures in Figure 7.

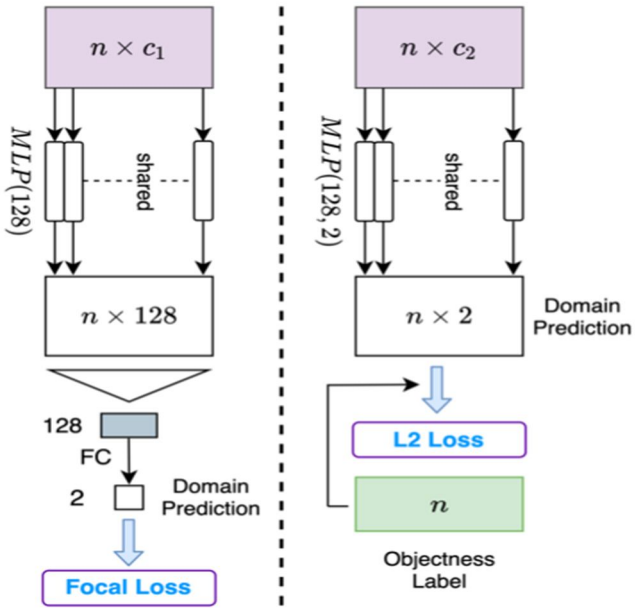


Figure 7. Architecture of the global and proposal discriminators. (Global on the left, proposal on the right.)

C. Implementation Detail of WS3D

In this section, we show how we implement WS3D [23] to adapt to indoor 3D object detection task.

C.1. Introduction of WS3D

Here is a simple summary of WS3D: The authors annotate the object centres in the bird's eye view (BEV) maps, which takes 2.5s per object. Then they utilize a two-stage approach to detect a specific category of objects (the author focus on Car in their paper), which can be divided into proposal and refinement stages. At the proposal stage, WS3D creates cylindrical proposals from the labelled centres, whose radius and height are fixed since the sizes of cars are close. Therefore the probability of a car being wrapped in a cylindrical proposal is high. Then a network (Net1) is trained to generate proposals from a point cloud scene. At the refinement stage, another network (Net2) is trained to take in the cylindrical proposal and output the bounding box of the car contained in the proposal, where around 3% well-labelled instances are used for supervision.

C.2. Proposal Stage

Since the indoor scenes in **ScanNetV2** are more complicated, the size and height of each object is different, even for objects in the same class. Therefore we annotate the object centres in three-dimensional space rather than in the **BEV** map, which is the same labelling strategy with us and takes 5s per object, to provide stronger supervision for WS3D. Instead of using a simple fixed-size cylinder as the proposal, we utilize a cuboid instead, whose size (length, width and height) is 1.5 times the average size of the object's category. In this way we are able to generate a more reasonable proposal.

During this stage, we can adopt different detectors as Net1. Net1 is trained with position-level annotations and used to predict the centres and semantic labels of objects (we adopt VoteNet and GroupFree3D as Net1 in our experiments). Then we generate cuboid proposals from the predicted centres and classes.

	Setting	batht.	bed	bench	bsf.	bot.	chair	cup	curt.	desk	door	dres.	keyb.	lamp	lapt.	monit.	n.s.	plant	sofa	stool	table	toil.	ward.	mAP@0.5
VoteNet	FSB [27]	69.8	76.9	6.7	26.0	0.0	67.6	0.0	10.2	30.0	13.3	21.1	0.0	15.5	0.0	19.6	47.9	3.1	70.4	10.1	38.9	85.0	2.7	28.0
	WSB	0.0	11.5	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.2	0.7	0.0	0.2	0.0	0.1	2.5	0.0	0.8
	WS3D† [23]	0.0	22.7	0.0	0.0	0.0	12.2	0.1	0.0	0.3	0.0	0.0	0.0	1.1	0.0	1.3	11.3	0.0	0.1	0.2	1.4	16.4	0.0	3.1
	$WSBP_P$	0.0	3.7	0.0	0.1	0.0	28.4	0.0	0.0	1.1	0.5	0.0	0.0	1.2	0.0	0.0	26.1	0.0	0.9	4.4	0.8	7.6	0.6	3.4
	$WSBP_M$	12.3	1.3	0.0	0.3	0.0	16.5	0.0	0.0	4.1	0.1	0.0	0.4	5.9	0.0	0.1	26.9	0.4	3.3	5.4	0.8	4.8	0.1	3.8
	BR_P(Ours)	36.8	15.2	1.2	6.9	0.0	42.7	0.0	0.0	4.4	1.3	2.1	0.0	9.0	0.0	2.7	31.4	1.3	14.4	4.1	8.3	51.6	0.0	10.6
	BR_M(Ours)	9.6	59.2	0.2	12.8	0.0	37.9	0.0	0.0	22.1	1.0	6.2	0.0	10.6	0.0	2.1	44.6	2.7	33.0	2.0	25.3	57.0	0.1	14.8
GroupFree3D	FSB [22]	75.7	75.6	4.5	28.4	0.0	75.3	0.0	20.3	47.4	24.7	29.5	0.3	20.4	0.0	37.5	61.4	3.7	74.6	37.1	51.1	96.2	11.7	35.2
	WSB	1.9	24.7	0.0	0.1	0.0	31.2	0.0	0.0	0.1	0.1	0.0	0.0	6.5	0.0	2.1	1.5	0.1	2.6	2.0	0.5	54.3	0.0	5.8
	WS3D† [23]	3.8	25.7	0.0	0.1	0.0	36.4	0.0	0.0	2.1	0.0	0.3	0.3	10.2	0.0	7.5	16.4	0.2	2.7	4.5	0.4	68.3	0.0	8.1
	$WSBP_P$	1.9	5.2	0.0	1.3	0.0	31.8	0.0	11.3	1.1	0.1	0.0	0.0	18.7	4.4	1.0	48.1	1.3	1.3	1.3	0.6	62.0	1.8	8.3
	$WSBP_M$	4.9	16.7	0.0	0.5	0.0	34.1	0.0	0.1	5.6	0.2	0.5	0.1	9.0	4.6	8.9	48.5	0.9	9.9	12.3	3.4	51.9	0.0	9.6
	BR_P(Ours)	83.6	79.1	0.0	10.8	0.0	53.5	0.0	0.0	0.0	1.6	3.7	0.0	19.6	50.0	6.5	60.0	16.7	21.1	5.7	14.6	90.1	0.0	23.5
	BR_M(Ours)	83.3	65.0	0.0	4.1	0.0	56.2	0.0	0.5	11.8	2.1	16.7	1.2	23.8	12.5	16.0	80.0	17.5	42.2	28.6	28.0	99.2	0.0	26.8

Table 7. The class-specific detection results (mAP@0.5) of different weakly-supervised methods on ScanNetV2 validation set. (FSB is the fully-supervised baseline. † indicates the method requires a small proportion of bounding boxes to refine the prediction. Other methods only use position-level annotations as supervision.)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

C.3. Refinement Stage

We find 3% well-labelled bounding boxes are not enough to train the Net2, as there 22 categories in our benchmark and the size of each object is very different, so we use around 15% bounding boxes instead. The proposals generated from the previous stage are post-processed by a 3D NMS module with an IoU threshold of 0.25, and then re- fined into precise bounding boxes by Net2.

We adopt a PointNet++-like module as Net2, whose in- put is the point cloud inside the cuboid proposal and output is the refined centre coordinate, box size and box orientation.

D. Augmentation Strategy

As the number of scenes which contain small objects[6] and the probability of small objects being sampled are relatively smaller than others, it is difficult for the detector to learn how to locate small objects in complex scenes. There- fore we utilize an augmentation strategy similar to [16] to handle the problem.

During training, we oversample the virtual scenes which contain small objects twice in each epoch. We further copy- paste small objects to the oversampled virtual scenes: for each small object, we copy it with a probability of 0.75 and paste it randomly in the scene (the pasted centre must be in the axis-aligned bounding box of the whole scene). Then we apply gravity and collision constraints and control the densities of these added small objects as mentioned in the virtual scene generation method.

Apart from small objects, we also consider the scarce objects[7], as the number of them is relatively small and thus the detector is not sufficiently trained on these categories. We add the scarce objects to the oversampled virtual scenes to expand the number of them. We first decide how many objects of each scarce category we should add according to ***Table 2*** in the main paper, where we set 40, 70, 15, 55 and 50 for bathtub, bench, dresser, laptop and wardrobe respectively. Then we choose scenes which are suitable for adding these objects by calculating the value of correlation between scenes and scarce categories as below:

$$Corr(s,c) = \sum_{i=1}^{22} l_{s_i}(v_{c_i} - r) \qquad (12)$$

where s indicates a scene and c denotes a scarce category. $\mathbf{l_S}$ is a 22-dimensional boolean vector where $\mathbf{l_{Si}}$ indicates whether there is an object of the i-th category in **s**. $\mathbf{v_C}$ is a 22-dimensional vector which indicates the correlation be- tween c and other categories:

$$v_{c_i} = \begin{cases} \frac{Num(i, Index(c))}{Num(Index(c))}, & i \neq Index(c) \\ 0, & i = Index(c) \end{cases} \qquad (13)$$

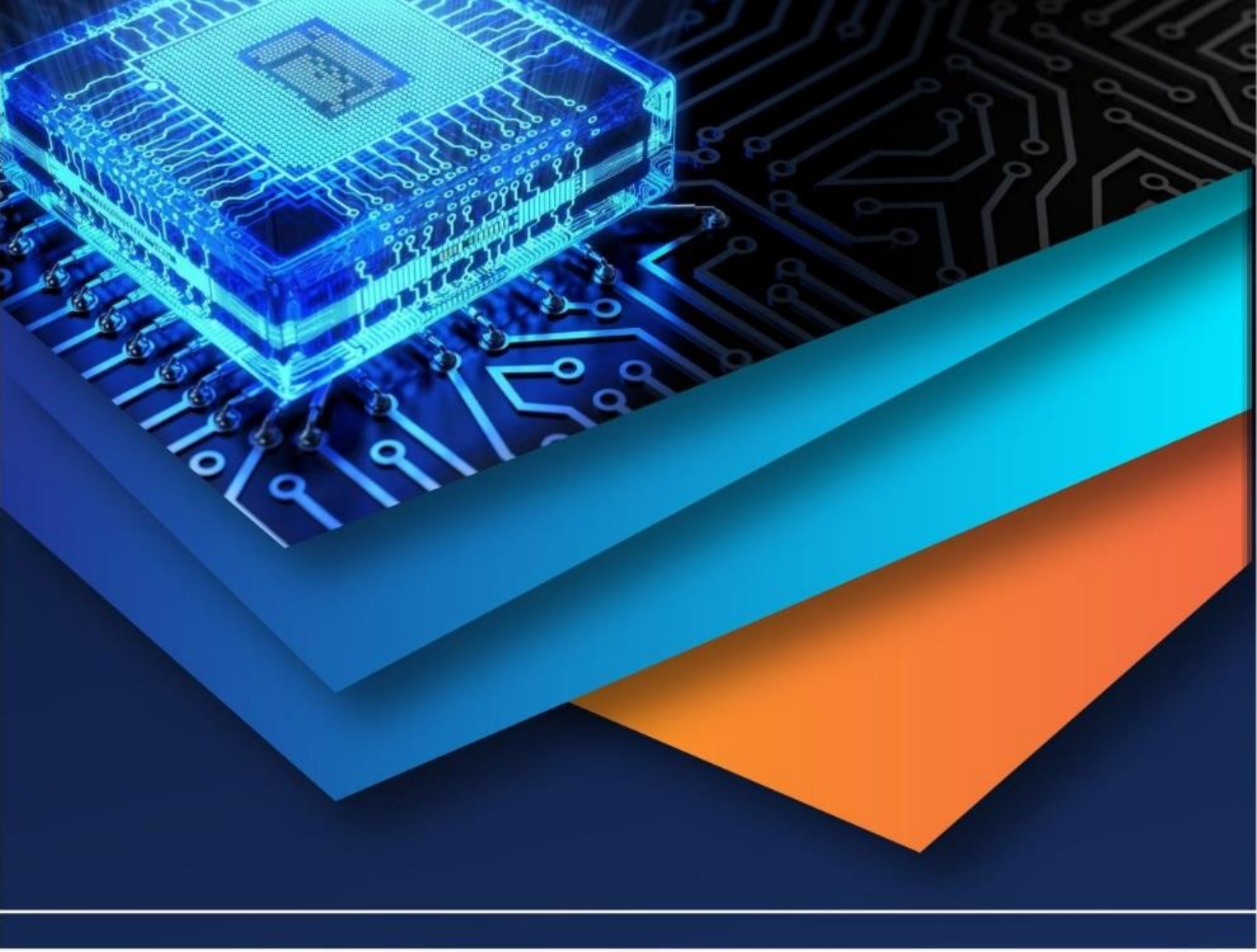
where **N um(...)** is a function, whose input is a set of indices of categories and the output is the number of scenes which contain objects in all the input categories. The larger $\mathbf{v_{Ci}}$, the stronger the correlation between **c** and the i'th category. It is hoped that the highly correlated scenes for **c** do not contain too many categories with low $\mathbf{v_{Ci}}$, a penalty term **r** was therefore introduced to reduce the value of **Corr(s, c)** when there are a large number of categories weakly correlated to **c** in **s**. r is set to 0.25 in our experiments.

E. More Detection Results

Three-dimensional object detection results are shown (mAP@0.5) to differ basis weakly-supervised methods employed on the ScanNetV2 [9] validation set in ***Table 7***.

Consistent with the results on **mAP@0.25**, our ***Corporeality (BR)*** approach achieves the best performance among all the weakly-supervised approaches. Under a more strict metric, the performances of most weakly-supervised approaches fail to surpass 10% in terms of **mAP@0.5**, that shows it is really hard to precisely detect the objects in a complicated indoor scene for a detector trained with only position- level annotations. However, the performance of **BRM** (for **GroupFree3D**) still achieves 26.8% in terms of **mAP@0.5**, which is comparable to the performance of fully-supervised **VoteNet**.

It also appears that the ***Corporeality (BR)*** approach works better on GroupFree3D than on **VoteNet** (the gap between **FSB** and ***Corporeality (BR)*** is smaller). This may be due to the features extracted by stronger detector has better generalization ability and thus our virtual2real domain adaptation method can transfer more useful knowledge contained in the virtual scenes to real-scene training.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY