Back to Reality: Learning Data-Efficient 3D Object Detector with Shape Guidance

Xiuwei Xu, *Student Member, IEEE*, Ziwei Wang, *Student Member, IEEE*, Jie Zhou, *Senior Member, IEEE*, and Jiwen Lu, *Senior Member, IEEE*

Abstract—In this paper, we propose a weakly-supervised approach for 3D object detection, which makes it possible to train a strong 3D detector with position-level annotations (i.e. annotations of object centers and categories). In order to remedy the information loss from box annotations to centers, our method makes use of synthetic 3D shapes to convert the position-level annotations into virtual scenes with box-level annotations, and in turn utilizes the fully-annotated virtual scenes to complement the real labels. Specifically, we first present a shape-guided label-enhancement method, which assembles 3D shapes into physically reasonable virtual scenes according to the coarse scene layout extracted from position-level annotations. Then we transfer the information contained in the virtual scenes back to real ones by applying a virtual-to-real domain adaptation method, which refines the annotated object centers and additionally supervises the training of detector with the virtual scenes. Since the shape-guided label enhancement method generates virtual scenes by human-heuristic physical constraints, the layout of the fixed virtual scenes may be unreasonable with varied object combinations. To address this, we further present differentiable label enhancement to optimize the virtual scenes including object scales, orientations and locations in a data-driven manner. Moreover, we further propose a label-assisted self-training strategy to fully exploit the capability of detector. By reusing the position-level annotations and virtual scenes, we fuse the information from both domains and generate box-level pseudo labels on the real scenes, which enables us to directly train a detector in fully-supervised manner. Extensive experiments on the widely used ScanNet and Matterport3D datasets show that our approach surpasses current weakly-supervised and semi-supervised methods by a large margin, and achieves comparable detection performance with some popular fully-supervised methods with less than 5% of the labeling labor.

Index Terms—3D Object Detection, Weakly-supervised Learning, Label Enhancement.

1 Introduction

Three-dimensional object detection is a fundamental scene understanding problem, which aims to detect 3D bounding boxes and semantic labels from a point cloud of 3D scene. Due to the irregular form of point clouds and complex contexts in 3D scenes, most existing 2D methods [1], [2], [3] cannot be directly applied to 3D object detection. Fortunately, with the development of deep learning techniques on point cloud understanding [4], [5], recent works [6], [7], [8], [9] have employed deep neural networks to directly detect objects from point clouds and achieved favorable performance.

Despite the successes in deep learning based object detection on point clouds, massive amounts of labeled bounding boxes significantly limits the applications of these methods, as labeling a precise 3D box takes around 120s even by an experienced annotator [10]. Therefore, 3D object detection methods using cheap labels are desirable for practical applications. Motivated by this, increasing attention has been paid to weakly-supervised 3D object detection methods, which replace the box-level annotations with simpler ones to largely reduce the annotating time. Mainstream weakly-supervised methods for 3D object detection can be divided into two categories: scene-level [11] and position-level [12], [13] where only the class tag (about 1s per object) and both object center and class (about 5s per object) are annotated for each instance respectively. While scene-level annotation is more time-saving, it is hard for the detector to learn how to precisely locate each object in a scene due to the lack of position information, and thus the performance is far from satisfactory [11]. Considering the time-accuracy tradeoff, position-level annotation is a more practical solution. However, previous position-level weakly-supervised 3D detection methods still require a number of precisely labeled boxes and can only cope with sparse outdoor scenes [12], [13]. We study purely position-level weakly-supervised 3D object detection for the complicated indoor scenes, where the spatial relation between objects are complicated and the size variance of objects is large.

In this paper, we propose a data-efficient training paradigm called ***BackToReality*** (BR) for weakly-supervised 3D object detection. To reduce the labor cost, we only label the center and category of each object in the 3D space and the labeling error of centers is allowed[1]. While largely reducing the workload of labeling, the information loss is non-negligible from box annotations to centers. To address these, BR converts the weak labels into virtual scenes which complements object size information and in turn utilizes them to additionally supervise real-scene training, as shown in Fig. 1. Our approach is based on two motivations: 1) in 3D vision, large-scale datasets of synthetic shapes are available. They contain rich geometry information, which can serve as strong priors to assist 3D object detection; 2) the position-level annotations are not only supervision for training, but they also provide coarse layout of the scene. Therefore, we first present a shape-guided label-enhancement method[2], which

• *The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: xxw21@@mails.tsinghua.edu.cn, wang-zw18@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn, lujiwen@tsinghua.edu.cn.*

1. We show the detailed labeling strategy in Section 3.1.

2. Label enhancement (LE) is a technique to recover label distributions from logical labels, as defined in [14]. Here we extend the concept of LE to denote the process of recovering the lost information for weak labels.

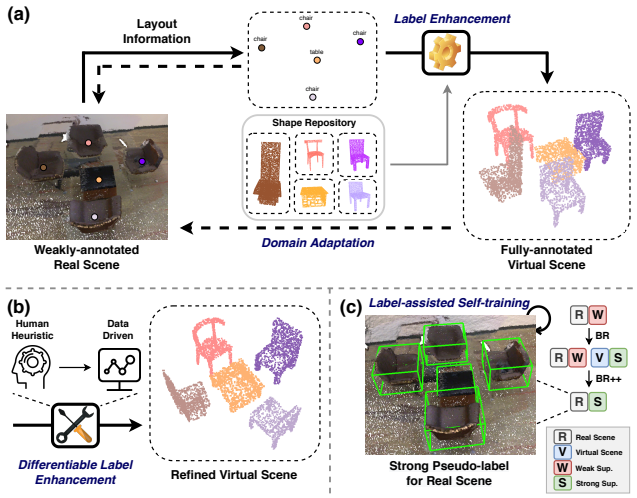

Fig. 1. Demonstration of *BR* and *BR++*. (a) shows the proposed training paradigm of BR. We consider position-level annotations as the coarse layout of the scenes, which is utilized to generate virtual scenes from a 3D shape repository by our *shape-guided label enhancement* method. Physical constraints are applied to construct the virtual scenes for remedying the information loss from box annotations to centers. Then a *virtual-to-real domain adaptation* method is presented to additionally supervise the training of real-scene 3D object detector with the virtual scenes. Dashed arrows indicate supervision for training. (b) demonstrates the presented *differentiable label enhancement* module of BR++. In order to improve the unreasonable scene arrangement caused by human-heuristic generation of the virtual scenes, we conduct label enhancement in a data-driven manner and optimize this module end-to-end with the detector. (c) illustrates the *label-assisted self-training* method of BR++. Aiming to fully exploit the capability of the detector, we further utilize position-level annotations and virtual scenes to assist the post-processing of the predictions and generate high-quality box-level pseudo labels on the real scenes.

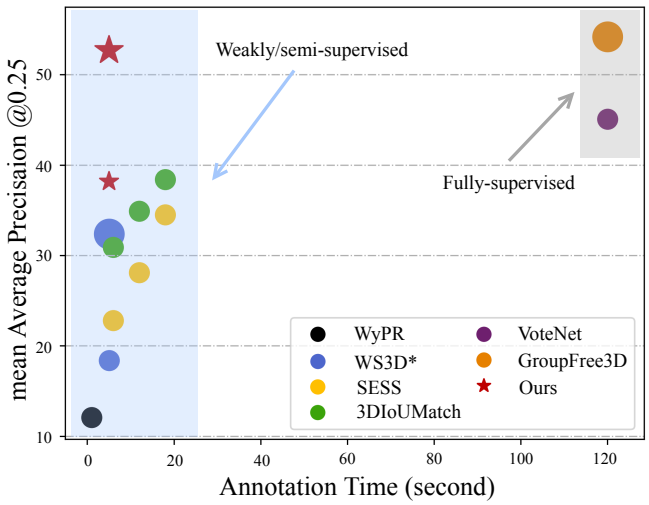

Fig. 2. Performance (mAP@0.25) *vs.* annotation cost (second per object), tested on ScanNet-md40 benchmark. Compared with current fully-supervised detectors including VoteNet [8] and GroupFree3D [9], semi-supervised methods like SESS [16] and 3DIoUMatch [17] and weakly-supervised methods such as WyPR [11] and WS3D [12], [13], our methods achieve the best time-accuracy tradeoff. * means that WS3D additionally requires a small proportion of bounding boxes to refine the predictions. Markers of different sizes represent different detectors: the bigger one is GroupFree3D and the smaller one is VoteNet. For semi-supervised methods, we report performance under different ratios of annotated scenes (5%, 10% and 15%).

assembles the 3D shapes into fully-annotated virtual scenes with physical constraints according to the coarse layout to remedy the information loss. Then a virtual-to-real domain adaptation method is presented to align the global features and object proposal features between the real and virtual scenes. Moreover, our method can take advantage of the precise center labels in virtual scenes to correct the center error of position-level annotations. In this way the useful knowledge contained in virtual scenes is transferred back to reality.

In fact, the human-heuristic physical constraints cannot ensure that the layout of virtual scenes is reasonable, which may lead to large domain gap between the virtual and real scenes. Moreover, the feature alignment for domain adaptation ignores the architecture design of the detector and thus cannot fully exploit the network capability. Based on those observations, we further propose ***BackToReality++*** (BR++) with differentiable label enhancement and label-assisted self-training. More specifically, we maintain a set of learnable pose parameters which assign the 3D shapes to proper positions in the scenes during training. This enables us to optimize the virtual scenes end-to-end with the detector in a data-driven manner. After training the detector, we reuse the position-level annotations and virtual scenes to filter out predicted bounding boxes with false categories, imprecise centers and unreasonable sizes. In this way, we acquire pseudo box-level annotations on real scenes to train the detector in a fully-supervised manner. Extensive experiments on two mainstream indoor datasets ScanNet and Matterport3D demonstrate the superior performance of our method and show position-level annotations can be very competitive alternatives of box annotations for training a strong 3D object detector. Under the same annotating time and network architecture, our method surpasses previous state-of-the-art weakly/semi-supervised methods by more than 12.4% in terms of mAP@0.25. Moreover, we achieve comparable results with the fully-supervised baseline on both datasets (within 2% mAP@0.25) while only requires less than 5% labeling labor. Code is available at: https://github.com/xuxw98/BackToReality.

This paper is an extended version of our conference paper [15], where we make the following new contributions: 1) we extend the shape-guided label enhancement method in the conference version to a differentiable one by optimizing the pose parameters of synthetic shapes end-to-end with the detector in a data-driven manner, so that the layouts of virtual scenes become more reasonable and the domain gap between real and virtual scenes is narrowed; 2) we further design a label-assisted self-training paradigm for weakly-supervised domain adaptation which generates pseudo box-level labels on the real scenes, enabling us to train a detector with explicit supervision and thus fully exploit the network capability; 3) we conduct extensive experiments on the more challenging Matterport3D dataset to verify the generality of our approach. Moreover, we further compare our method with semi-supervised ones for more comprehensive evaluation.

2 Related Work

We briefly review three related topics: 1) 3D object detection, 2) 3D scene generation with synthesis shapes and 3) 3D domain adaptation.

3D Object Detection. In terms of fully-supervised 3D object detection, early methods mainly include template-based methods [18], [19] and sliding-window methods [20], [21]. Deep

learning-based 3D detection methods for point clouds began to emerge thanks to PointNet/PointNet++ [4], [5]. However, methods in [22], [23], [24], [25] rely on generating 2D proposals and then project them into the 3D space, which is hard to handle scenes with heavy occlusion. More recently, networks that directly consume point clouds have been proposed [6], [7], [8], [9], [26]. PointRCNN [6] is a two-stage detector which first generates 3D proposals directly from the whole point clouds and then adopts region pooling to extract features for each object proposal. VoteNet [8] proposes a one-stage detector using the hough voting strategy for better object feature grouping. While the development of 3D object detection methods is rapid, the application is still restricted partially due to the limited labeled data. To reduce the labor of human annotation, semi-supervised methods [16], [17] and weakly-supervised methods [11], [12], [13], [27] have been proposed recently. Semi-supervised methods only label a part of the scenes. Following the similar procedure as their 2D counterparts [28], they leverage a mutual learning framework composed of an EMA teacher and a student, which are fed with data under asymmetric augmentation. Different kinds of consistency losses between the teacher and student outputs are applied during training. Although semi-supervised methods bring significant improvement compared to simply training on labeled scenes, they still require more than 10% labeled scenes to achieve a satisfactory performance, as shown in Fig. 2, while weakly-supervised methods only need less than 5% labeling time [11], [12]. However, existing weakly-supervised methods still face huge challenges in the complicated indoor 3D object detection task, which is the problem we aim to solve in this work.

3D Scene Generation with Synthesis Shapes. Since it is much easier to obtain a large scale synthetic 3D shape dataset than a real scene dataset, utilizing 3D shapes to assist scene understanding is a promising idea. Existing approaches can be divided into two categories: supervised [29], [30], [31], [32], [33], [34] and unsupervised [35], [36], [37], [38], [39]. For supervised methods, there are two main application scenarios: 3D reconstruction [18], [29], [30], [31], [40] and indoor scene synthesis [32], [33], [34]. In terms of 3D reconstruction, the synthetic shapes are usually used to complete the imperfect 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 [29], [30], [31]. Human-annotated pairs of raw scans and object-aligned scans are used in the training process. With respect to indoor scene synthesis, recent works use GAN [41] or CNN to recursively generate room layouts, based on which synthetic shapes are placed into the scene [32], [33], [42], [43]. In this way, given an empty or partial scene, multiple completions of the scene can be acquired. Fully-annotated and complete scenes are required during training. As supervised methods need extra human labor, that may limit the full utilization of 3D shape datasets. Unsupervised methods are usually used for data augmentation or dataset expansion. 3D shapes are placed in a random manner following the basic physical constraints, in order to generate mixed reality scenes [35], [36] or virtual scenes [37], [38], [39]. Recently, MetaSim [44] proposes a generative model of synthetic scenes, which produces images as well as its corresponding ground-truth via a graphics engine. MetaSim2 [45] further enhances this framework by learning the scene structure in addition to parameters. Our approach also utilizes 3D shapes to assist object detection in an unsupervised manner. Differently, we aim to make use of synthetic shapes to enhance the weak label and gain stronger supervision in position-level weakly-supervised detection.

3D Domain Adaptation. Domain adaptation aims to generalize the model trained on source domain to target domains where data or label is limited. While extensive researches have been conducted on domain adaptation tasks with 2D image data [46], [47], [48], [49], [50], there are only a small number of literature in the field of 3D point cloud domain adaptation. PointDAN [51] proposes to jointly align local and global features using maximum discrepancy and adversarial training for shape classification. Jaritz et al. [52] projects point cloud to 2D images and train models with multi-modal information. Yi et al. [53] presents a sparse voxel network to perform point cloud completion for domain adaptive semantic segmentation. For object detection, Wang et al. [54] propose SN to normalize the object size of the source domain leveraging the statistics of the target domain to close the size-level domain gap. SF-UDA [55] computes motion coherence over consecutive frames to select the best scale for the target domain. SRDAN [56] employes scale-aware and range-aware feature alignment to match the distribution between two domains. MLC-Net [57] adopts the meanteacher paradigm to address the geometric mismatch between point clouds from source and target domains. While these works have made great progress in domain adaptive outdoor 3D object detection, domain adaptation for indoor scene understanding tasks are still under exploration.

3 Approach

In this section, we first discuss our position-level annotating strategy. Then we introduce the weakly-supervised training paradigm for 3D object detector called BackToReality. Finally, we present BackToReality++ with differentiable label enhancement and label-assisted self-training, which enables efficient end-to-end training and further improves the detection performance.

3.1 Position-level Annotation

As choosing a point in the 3D space is hard, we divide the labeling process into two steps: firstly we label the center of an object in a proper 2D view of the scene, and compute the line that goes through this center and the focus point of the camera according to the camera parameters of the 2D view. Secondly we choose a point on the line to determine the object's center in the 3D space. This strategy requires less than 5s to label an instance, and the labeling error can be controlled within 10% of the instance size. More details can be found in supplementary material.

The input of our approach is a 3D shape repository and point cloud scenes with position-level annotations, and output is a 3D detector which is able to predict precise bounding boxes. When the 3D scene is scanned, in many cases we can acquire **mesh** data which contains more geometric information than point clouds. Therefore our method also take consideration of the situation when mesh data is available in the training scenes.

3.2 BackToReality

Fig. 3 illustrates the framework of BackToReality. Given real scenes with position-level annotations, we utilize 3D shapes to convert the weak labels into virtual scenes, which are utilized to provide additional supervision for the training of the detector. Then we propose a virtual-to-real domain adaptation method to transfer the knowledge contained in the fully-annotated virtual scenes back to reality.

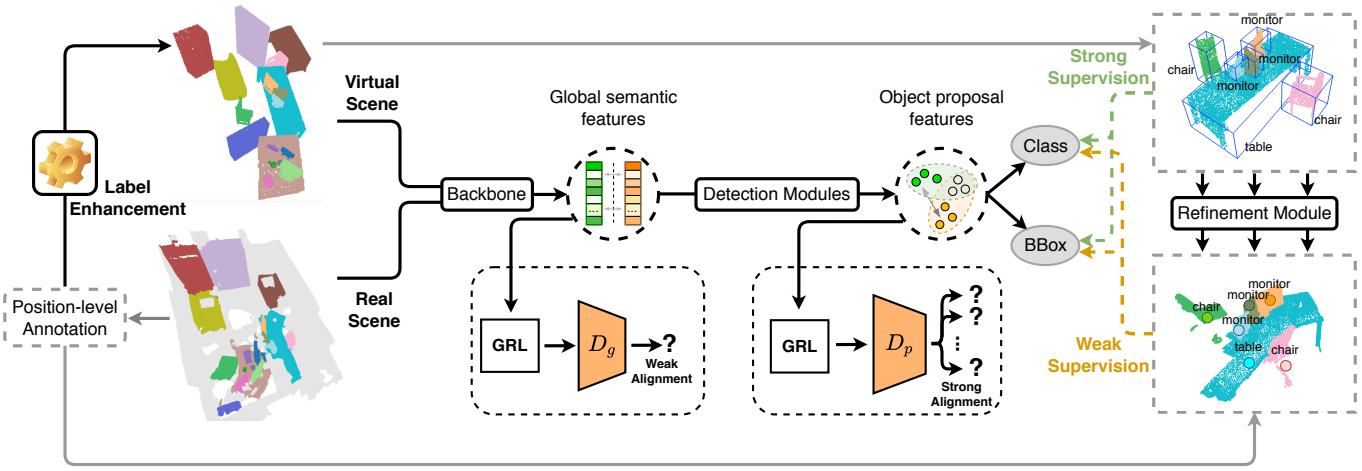

Fig. 3. The framework of our *BR* approach. Given real scenes with position-level annotations, we first enhance the weak labels to get fully-annotated virtual scenes. Then the real scenes and virtual scenes are fed into the detector, trained with weakly-supervised and fully-supervised detection loss respectively. During training we use the precise object centers in virtual scenes to refine the imprecise centers 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; D_g and D_p stand for the global and proposal discriminators respectively.)

3.2.1 Shape-guided Label Enhancement

Due to lost object size information and imprecise object centers, it is challenging to train a 3D detector which predicts accurate bounding boxes. In spite of this, position-level annotations can provide a coarse layout of the scenes. By assembling synthetic 3D shapes according to the layout, we are able to enhance the weak labels and generate accurately annotated virtual scenes where sizes are available and centers are precise. Our label enhancement method includes two steps: 1) first we compute some basic properties of 3D shapes; 2) then we place these shapes to generate physically self-consistent virtual scenes from the labels.

Definition of Shape Properties: Given a synthetic 3D shape, which is represented as $O \in R^{N\times 3}$, we assume it is axis-aligned and normalized into a unit sphere. The length, width and height of O is defined as l, w and h. Then we divide the categories of shapes into three classes: supporter, stander and supportee. Supporters and standers are objects that can only be supported by ground, with the difference that standers are not likely to support other things. Other categories are supportees.

If a shape belongs to supporter, 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. The CSS is a boolean value, indicating whether the supporting surface can be approximated by the MER.

Virtual Scene Generation: We utilize a three-stage approach to construct the virtual scenes, which is equivalent to generate the position of each shape stage by stage: 1) we first refine the coarse layout provided by position-level annotations and generate the initial positions; 2) then we generate gravity-aware positions by restoring the supporting relationships between objects; 3) lastly we generate collision-aware positions to make the virtual scenes physically self-consistent. The pipeline is shown in Fig. 4.

To generate *initial positions*, we need to recover a more precise layout from the geometric information of the scenes. Given a scene in mesh format, we first oversegment the meshes using a normal-based graph cut method [58]. The result is a segment graph, where the nodes indicating segments and the edges denoting adjacency relations. Then for horizontal segments whose area is larger than A_{min} and height is larger than H_{min}, we iteratively merge their neighbors into them if the height difference between the horizontal segment and the neighbor segment is smaller than Δ_h. Once merged, the segments are considered as a whole and the height of the new merged segment is set to be same as the original horizontal segments. After merging, each horizontal segment is represented by its MER. If only one supporter's center falls in a MER, we assign this MER to the supporter. When the centers of multiple supporters fall in the same MER, we perform K-means clustering of the horizontal segment according to these centers and calculate MER for each supporter respectively.

Then we place the 3D shapes of corresponding categories on the centers given by position-level annotations and utilize the horizontal segments to refine the layout. The initial positions of the shapes are represented by a dictionary, whose key is the instance index and value is a list:

$$[(x, y, z), (s_x, s_y, s_z), \theta, O] \tag{1}$$

where the instance index is integer ranging from 1 to the number of objects in the scene. (x, y, z) denotes the center coordinates. (s_x, s_y, s_z) indicates the scales in three dimensions. θ and O represent the rotation angle and the original point cloud of the shape. If the shape belongs to supporters and has been assigned a horizontal segment, we use the MER of that segment to initialize the above parameters. That is, we choose a supporter whose CSS is True and make the MER of this supporter overlap with the horizontal segment. Otherwise we initialize the above list randomly. If only point cloud data is available, we simply perform random initialization and the following stages are the same. As the random initialization of scale should be constrained in a reasonable range, we first estimate the average size (X, Y, Z) of objects in each category. Then we uniformly sample s_x, s_y and s_z in $[0.8X, 1.3X]$, $[0.8Y, 1.3Y]$ and $[0.8Z, 1.3Z]$ respectively.

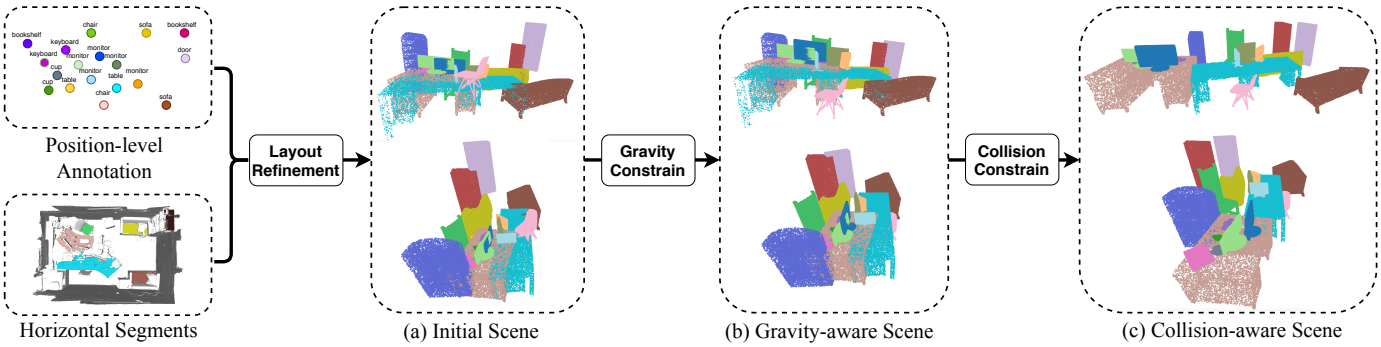

Fig. 4. The pipeline of our three-stage virtual scene generation method. We first extract horizontal segments from the mesh data and use them to refine the coarse layout provided by position-level annotations. Then synthetic 3D shapes are placed in virtual scenes according to the new layout to construct 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.

(X, Y, Z) can be coarsely computed from several fully-annotated objects.

Next we traverse the initial positions to generate *gravity-aware positions*. In this process we only need to change z in the position dictionary. For supporters and standers, we directly align their bottoms with the ground (i.e. the XY plane). For a supportee, if its (x, y) fall in any supporter's MER, we assign it to the nearest supporter and align its bottoms with the supporting surface. Otherwise, it is aligned to the ground.

After that we move the shapes to acquire *collision-aware positions*. This stage only x and y in the position dictionary will be changed. First we sort the key objects by their horizontal distance to $(0, 0)$. Nearer object template are processed more first, which is likely to move fewer objects. Then we compute the moving vector as below:

$$\boldsymbol{v} = \sum_{i \in nearer} \frac{\boldsymbol{d_i}}{||\boldsymbol{d_i}||^2}, \ \boldsymbol{d_i} = (x - x_i, y - y_i) \tag{2}$$

where (x, y) is the coordinates on X-Y plane. When moving the supporters, the supported objects are moved together. To judge collision, we compute the distance of the nearest point pair of two shapes. If the distance is less than $0.02m$, the two are considered colliding. For the supportees, we handle collision in a similar way with two differences: (1) we sort supportees on a same supporter by their horizontal distance to the center of the supporter. Further supportees are processed more first, to avoid objects falling off the supporter; (2) the moving operation only performs on individuals. Note that the three generation stages can not only make the virtual scenes more realistic, but also weaken the impact of imprecise center labels. Thus the virtual scene generation method is robust to labeling errors.

Finally, we convert the collision-aware 3D shapes to point clouds with proper density. As larger surfaces are more likely to be captured by the sensor, we use the maximum of $(ls_x)(ws_y)$, $(ws_y)(hs_z)$ and $(ls_x)(hs_z)$ to approximate the surface area of 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.

3.2.2 Virtual-to-Real Domain Adaptation

While the label enhancement approach is able to generate physically self-consistent 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, we need to mine useful knowledge in the perfect virtual labels to make up for the information loss of position-level annotations, rather than just relying on the virtual scenes.

We refer to the virtual scenes and real scenes as source domain and target domain respectively. A virtual-to-real adversarial domain adaptation method is utilized to solve the above problem, whose overall objective is:

$$\begin{aligned} \max_D \min_G J &= L_{sup}(G) - L_{adv}(G, D) \\ &= (\lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3) - (\lambda_4 L_4 + \lambda_5 L_5) \end{aligned} \tag{3}$$

where G refers to the object detection network (detector) and 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 center 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). Below we detail these loss functions and the network.

Firstly we elaborate on $L_{sup}(G)$. As shown in Fig. 3, we divide the detector 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, we jointly refine the imprecise center labels in target domain and supervise the predictions of the detector. As shown in Fig. 5, we jitter the center labels in source domain by adding noise within 10% of the objects' sizes to imitate the labeling error in target domain. Then for each jittered center, we query its k nearest neighbors in 3D euclidean space from the global semantic features to construct a local graph, and predict the center offset through a PointNet-like module:

$$PN(c) = \text{MLP}_2 \left\{ \max_{i \in N(c)} \{\text{MLP}_1[f_i; c_i - c]\} \right\} \tag{4}$$

where PN denotes the PointNet-like module, c indicates the jittered center label, $N(c)$ is the index set of the k nearest neighbors of c, f_i is the global semantic feature, whose coordinate is c_i, and *max* refers to the channel-wise max-pooling. We set $\boldsymbol{L_1}$ as the mean square error between the ground-truth center offset

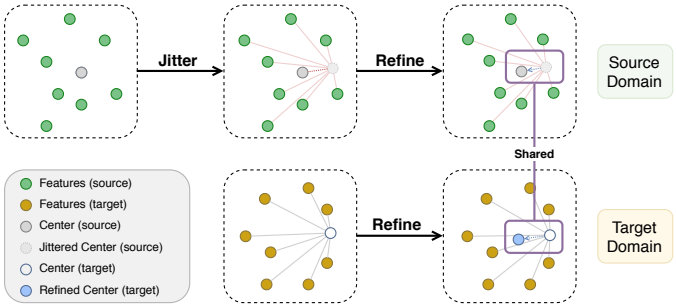

Fig. 5. Demonstration of our center refinement method. We first jitter the center labels in source domain to imitate the labeling error in target domain, and utilize a PointNet-like module to predict the center offset from the local graph of the jittered centers. This module can be directly utilized to predict the center error in target domain as the global semantic features from the two domains have been aligned.

and $PN(c)$. Then for fully-supervised training, the detection loss $\boldsymbol{L_2}$ is the same as the loss utilized in the original method. For weakly-supervised training, we utilize p to predict the center error in target domain and acquire refined center labels. We set $\boldsymbol{L_3}$ as a simpler version of L_2 which ignores the supervision for box sizes and orientations. More details about L_3 can be found in supplementary.

Secondly we analyze $L_{adv}(G, D)$. We conduct feature alignment in an adversarial manner: the discriminator predicts which domain the features belong to, and the detector aims to generate features that are hard to discriminate. The sign of gradients is flipped by a gradient reversal layer [59].

As the virtual scenes and real scenes are processed by the same network, we hope L_3 helps the network learn how to locate each object in real scenes, and L_2 compensates for the information loss of centers and sizes. However, due to the domain gap, L_2 will introduce domain-specific knowledge of the virtual scenes, which impairs the influence of L_3. Besides, the center refinement module is trained only on source domain, which may not perform well on target domain. Therefore, we align the global semantic features and object proposal features with L_4 and L_5 respectively. Inspired by [60], the features are aligned with different intensities at different stages. For global semantic features, we use a PointNet to predict the domain label. Focal loss [60], [61] is utilized to apply weak alignment:

$$\boldsymbol{L_4} = -\sum_{i=1}^{B}(1-p_i)^{\gamma} log(p_i),\ \gamma > 1 \tag{5}$$

where B is the batch size, and p_i refers to the probability of the global discriminator's predictions on the corresponding domain. Features with high p is easy to judge, which means they are domain-specific features and forcing invariance to them can hurt performance. So a small weight is used to reduce their impact on training. 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:

$$\boldsymbol{L_5} = \sum_{i=1}^{B}\sum_{j=1}^{M} s_{ij}(1-p_{ij})^2 \tag{6}$$

where B is the batch size, M is the number of proposals, s_{ij} refers to the objectness label and p_{ij} is the probability of the proposal discriminator's predictions on the corresponding domain. We detail the architectures of center refinement module and discriminators in supplementary.

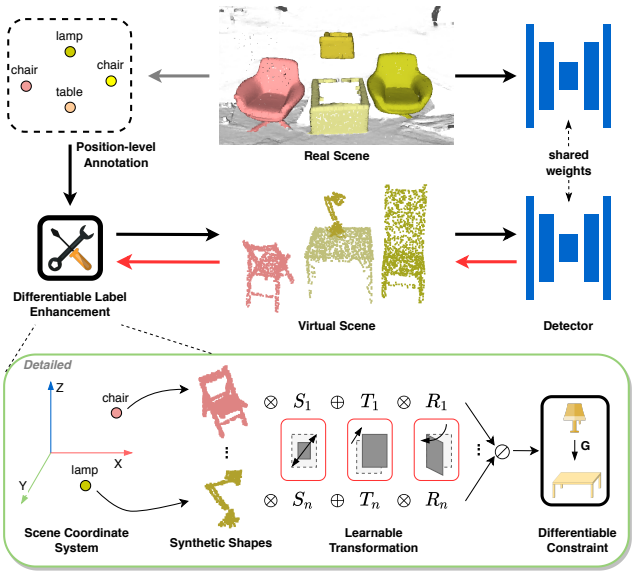

Fig. 6. Illustration of differentiable label enhancement. The black line, the red line, $\otimes$, $\oplus$ and $\oslash$ represent the forward propagation, the backward propagation, matrix multiplication, summation and concatenatation respectively. Given position-level annotations, we follow the training paradigm of BR to generate virtual scenes and feed both real and virtual inputs to a shared detector for further processing. However, here we utilize a differentiable module to generate the virtual scenes with gradient and train this module end-to-end with the detector. Specifically, for each virtual object, we maintain a set of learnable pose parameters, which transform the 3D shapes to proper positions in the scenes. To restrict the optimization space, we further apply a gravity-aware differentiable constraint on all objects to get the final virtual scenes.

3.3 BackToReality++

We first propose an end-to-end training framework to optimize the virtual scenes for more reasonable layout. Then we present a self-training strategy to generate pseudo box-level annotations for better exploiting the network capability.

3.3.1 *Differentiable Label Enhancement*

The human-heuristic physical constraints applied in shape-guided label enhancement do not consider the fine-grained layout of virtual scenes. Therefore the generated virtual scenes may still be unreasonable and deviate a lot from the real ones, which leads to large domain gap. In order to address these limitations, we further propose differentiable label enhancement to enable end-to-end training and adjust the virtual scenes in a data-driven manner, as shown in Fig. 6. We will demonstrate the differentiable framework in two steps: 1) first we define the learnable variables and generate virtual point cloud scenes with gradient given position-level annotations; 2) then we show how to jointly optimize the virtual scenes and the detector.

Generation: Given position-level annotations, we first assign a synthetic shape O and maintain a set of pose parameters for each of them, which can be written as an extension of Equation (1):

$$[(x, y, z), (s_x, s_y, s_z), (\Delta x, \Delta y, \Delta z), \theta, O] \tag{7}$$

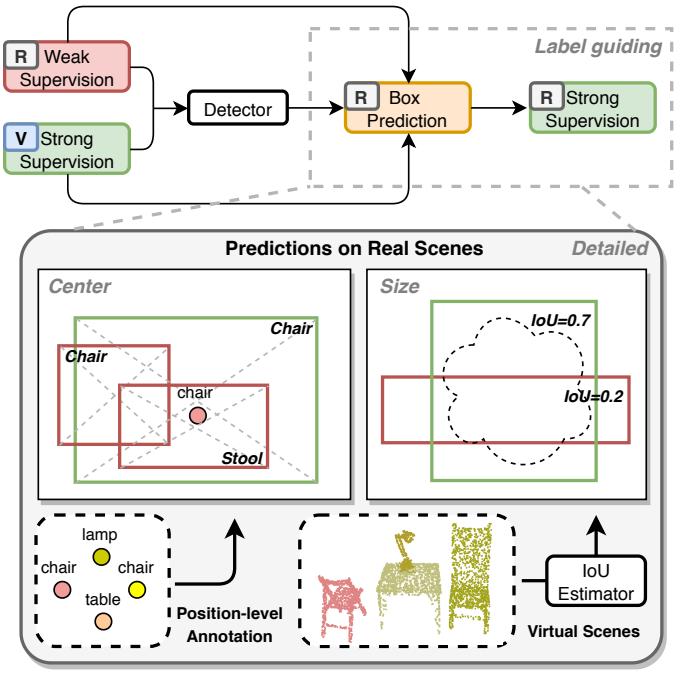

Fig. 7. Demonstration of label-assisted self-training. To fully exploit the information from both domains, we devise a new training paradigm for weakly-supervised domain adaptation. After the training phase of BR, position-level annotations and virtual scenes are utilized again to assist the process of Non-Maximum Suppression (NMS) and result in pseudo box-level annotations on real scenes, which serve as explicit supervision for the detector. To be specific, given box predictions, we utilize the position-level annotations to filter out predictions with false categories or imprecise centers, and train an IoU estimator to filter out predictions with unreasonable box sizes.

where (x, y, z) is the annotated center. (s_x, s_y, s_z), $(\Delta x, \Delta y, \Delta z)$ and θ are learnable variables and they represent scaling, translation and rotation respectively. We randomly choose the synthetic shapes and initialize the learnable scalings and rotations, while the learnable translations are set to 0 at beginning. We can convert these parameters to point cloud P by:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}, \; R = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$T = (x + \Delta x, y + \Delta y, z + \Delta z), \; \boldsymbol{P} = (O \cdot S + T) \cdot R$$

Then we concatenate the point cloud of each object to generate the virtual scenes. In this way, the gradient can be backpropagated from the detector to the generated point clouds, and then to the learnable pose parameters, which enables end-to-end training of the differentiable label enhancement module. Note that in BR, the virtual scenes are generated offline as the collision-aware constraint requires $O(N^2)$ time complexity to compute the nearest distance between two objects. While this differentiable module can generate virtual scenes in linear time with efficient matrix operations, which enables online generation.

To restrict the optimization space, we further propose a gravity-aware differentiable constraint to reduce the degrees of freedom of the learnable parameters. For each object, apart from the properties defined in Equation (7), we additionally consider SSH for supporters and bottom B for all objects (B is the minimum Z-coordinate of O). We follow the method introduced

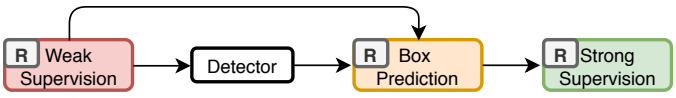

Fig. 8. Case when label-assisted self-training does not work. As the label guiding strategy is unable to bring additional size information, if we apply it in weakly-supervised training, the generated box-level pseudo labels are essentially the same as position-level annotations. Note that the essence of label-assisted self-training strategy is fusing the information from two domains into a whole and acquiring explicit supervision for the detector, which only works well for WDA problem.

in Section 3.2.1 to assign supportees to their corresponding supporters. Assume the height of floor is H, then the translation T for objects stand on the floor can be recomputed as:

$$T = (x + \Delta x, y + \Delta y, H - s_z \cdot B) \tag{9}$$

and that for supportees on others' supporting surfaces is:

$$T = (x + \Delta x, y + \Delta y, H - \underline{s_z} \cdot (\underline{B} - \underline{SSH}) - s_z \cdot B) \tag{10}$$

where the underlined properties belong to the corresponding supporters. As the gravity constraint can be computed in linear time, that will not affect the efficiency of online generation of the virtual scenes.

Optimization: We adopt two optimizers to update the weights of detector and the pose parameters alternatingly in each batch. In terms of loss function, L_2, L_4 and L_5 can backpropagate gradient to the differentiable label enhancement module. However, we find those functions are not suitable for learning proper poses for the virtual objects: the fully-supervised loss L_2 will guide the virtual scenes to change according to the predictions of detector, and the domain adaptation losses L_4 and L_5 only implicitly make features in real and virtual scenes similar by deceiving the discriminator, both of which lack clear supervision for the virtual scenes on how the real scenes look like. Therefore, we add a MMD loss term to minimize the distribution distance between the object proposal features of two domains, which clearly guides the virtual scenes to be closer to the real scenes:

$$L_D = \frac{1}{B^2 M^2} || \sum_{i=1}^{B} \sum_{j=1}^{M} \phi(A(f_{ij}^S)) - \phi(A(f_{ij}^T)) ||_H^2 \tag{11}$$

where B is the batch size, M is the number of proposals, ϕ is the radial basis function (RBF) kernel, A is a spatial self-attention layer, f^S and f^T represent the proposal features of source and target domain respectively. We use object poses from virtual scenes generated by shape-guided label enhancement method to initialize the learnable pose parameters.

3.3.2 Label-assisted Self-Training

In BR, we transfer the knowledge of object size contained in the virtual scenes to real-scene training by feature alignment, which can be considered as an implicit supervision for the predicted box sizes on real scenes. However, we find the implicit supervision is not the optimal training paradigm for this weakly-supervised domain adaptation (WDA) problem, which ignores the architecture design of the detector and cannot fully exploit the network capability (we will discuss this issue later in detail). Therefore, we present a label-assisted self-training method tailored for WDA to supervise the predicted box sizes explicitly with pseudo box-level annotations. As shown in Fig. 7, after the training phase of

BR, position-level annotations and virtual scenes can be utilized again to assist the process of Non-Maximum Suppression (NMS) and generate accurate pseudo labels on real scenes. In this way, the information contained in both domains are fused together, enabling us to directly train a detector in fully-supervised manner.

We acquire pseudo labels from the output of the detector, which usually applies Non-Maximum Suppression (NMS) based on objectness score to remove duplicated bounding boxes during inference time. As the inference is conducted on the training set, where position-level annotations are available, we leak the weak labels to filter out predictions with false categories or imprecise centers. Predictions that meet any of the following conditions:

$$p_{class} \neq g_{class} \tag{12}$$

$$||p_{center} - g_{center}||_2 > r \cdot p_{size} \tag{13}$$

will not be included in NMS. Where p_{center}, p_{size} and p_{class} refer to predicted center, size and category, g_{center} and g_{class} refer to ground-truth center and category. For each predicted box, we assign it a position-level annotation with the nearest center distance. Moreover, the objectness score cannot accurately reflects the reliability of the predicted boxes. Inspired by 3DIoUMatch [17], we train an IoU estimator on the virtual scenes, whose inputs are the global semantic features and a bounding box and output is the IoU between this box and the ground-truth. Thanks to feature alignment, the IoU estimator can be directly utilized on real scenes. Therefore, we filter out predictions with low IoU and conduct NMS based on the product of objectness score and IoU.

The final objective of BR++ can be divided into two stages. For the first stage (differentiable label enhancement and domain adaptation), the loss function is:

$$\begin{aligned}\max_{D}\min_{G,S} J &= L_{sup}(G) - L_{adv}(G,D) + L_D(G,S) \\ &= (\lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3) - (\lambda_4 L_4 + \lambda_5 L_5) \\ &+ \lambda_6 L_D\end{aligned} \tag{14}$$

where S refers to the learnable pose parameters of each synthetic shape. The loss function for the second stage (self-training) is:

$$\min_{G} J = L_2 \tag{15}$$

which is a fully-supervised loss on the real scenes with pseudo box-level annotations.

Discussion: We discuss two questions below: 1) Why reusing the position-level annotations for further training is a non-trivial solution? 2) Why the explicit box supervision is better than the implicit feature supervision?

The essence of label-assisted self-training strategy is not bringing additional information for training, but fusing the information from two domains into a whole. In this way, the implicit feature supervision is replaced by explicit box supervision, which benefits the training of detector (we will explain why in the next question). Fig. 8 shows a toy example when label-assisted self-training will be a trivial solution. If we first train a detector with only position-level annotations and then use the weak labels again to guide the post-processing of box predictions, the resulted box-level pseudo labels will do no good to the training of detector. That is because no size information is introduced during the whole training process, thus the box-level pseudo labels are essentially the same as position-level annotations.

Compared to the implicit feature supervision, the explicit box supervision has two advantages. First, it makes full use of the network architecture designs. Some advanced 3D detectors are designed for better exploiting the information of bounding box annotations. For example, H3D-Net utilizes a geometric primitive module to extract information from centers, edges and faces of the box annotations, and all voting-based 3D detectors require bounding boxes to generate the ground-truth for voting. On the contrary, with only implicit feature supervision, these architecture designs with strong inductive bias will not be fully utilized. Second, the box supervision avoids complex hyperparameter-tuning and shares nearly the same hyperparameters as the fully-supervised baseline. While in the feature alignment manner, there are additional hyperparameters and the training paradigm is very different from the fully-supervised one.

4 Experiment

In this section, we conduct experiments to show the effectiveness of our approach. We first describe the datasets and experimental settings. Then we evaluate the generated virtual scenes and present the detection results of our method. Finally we design several ablation studies to show the influence of each component or step in BR and BR++.

4.1 Experiments Setup

Datasets: We choose ModelNet40 [62] as the dataset of synthetic 3D shapes. ModelNet40 contains 12,311 synthetic CAD models from 40 categories, split into 9,843 for training and 2,468 for testing. We perform experiments on ScanNet [63] and Matterport3D [64] dataset with two proposed challenging benchmarks.

ScanNet is a richly annotated dataset of indoor scenes with 1201 training scenes and 312 validation scenes. For each object appeared in the scenes, ScanNet officially provides its corresponding class in ModelNet40. Therefore we choose 22 categories of ModelNet40 which have more than 50 objects in ScanNet training set, and report detection performance on them. Since ScanNet does not provide human-labeled bounding boxes, we predict axis-aligned bounding boxes and evaluate the prediction on validation set as in [8], [9], [65], [66]. We name this benchmark *ScanNet-md40*. Compared to the 18-category setting in previous works [8], [9], [65], our ScanNet-md40 benchmark is more challenging. Apart from the categories of big objects (e.g. desk and bathtub), we also aim to detect relatively small objects, such as laptop, keyboard and monitor.

Matterport3D is a more diverse indoor scene dataset with 1554 and 234 splits for training and validation respectively. Collected from 90 buildings, Matterport3D covers three times as many rooms as in ScanNet with far more room types including even outdoor space such as garden and rooftop. Similar to ScanNet, we select all 13 shared categories from Matterport3D and ModelNet40, each containing more than 80 object instances in Matterport3D training set, and report detection performance on them. In order to test the performance of 3D detector in predicting object orientations, we follow [67] to extract oriented bounding box annotations and evaluate the predictions on validation set. This benchmark is called *Matterport3D-md40*.

Compared Methods: To illustrate the effectiveness of our approach, the popular VoteNet [8] and state-of-the-art GroupFree3D [9] are selected as our detectors. We compare BR and BR++ with the following settings: 1) FSB: fully-supervised

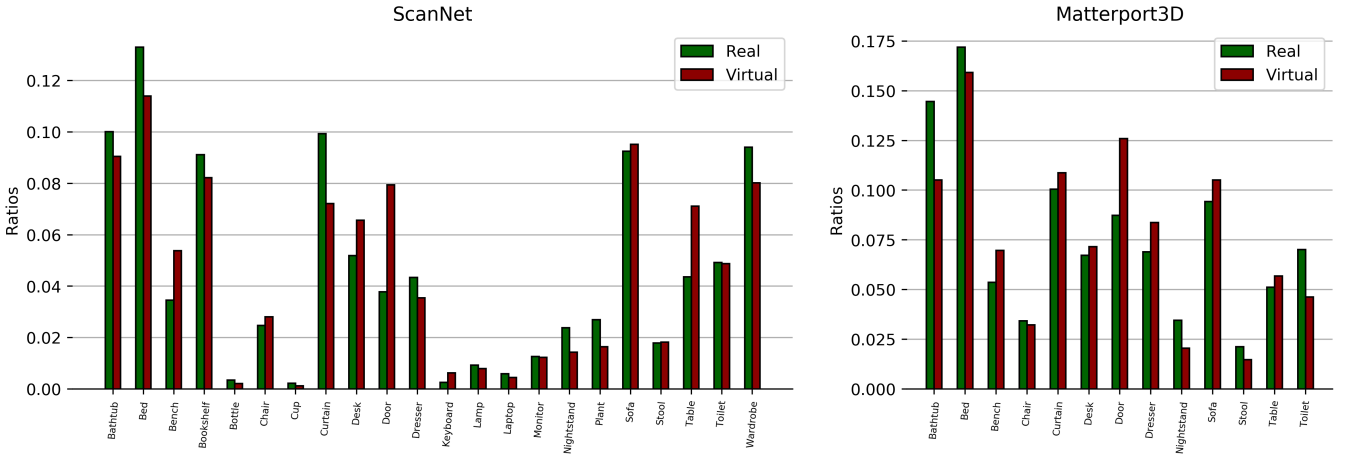

Fig. 9. Average point numbers of objects in each category in the real scenes and the virtual scenes for ScanNet and Matterport3D dataset. For better comparison between real and virtual scenes, we normalize the average point number of each category to ratios by dividing the summation of average point numbers among all categories. It can be seen that the ratios of each category in the real and virtual scenes are similar, which shows the effectiveness of our density control method.

baseline, which serves as the upper bound of weakly-supervised methods; 2) WSB: weakly-supervised baseline, which trains the detector on real scenes by using L_3 only and predicts the average size for each category; 3) WS3D: another position-level weakly-supervised approach proposed in [12], which additionally makes use of a number of precisely annotated bounding boxes; 4) WSBP: WSB pretrained on the virtual scenes; 5) SESS/3DIoUMatch: top-performance semi-supervised methods proposed in [16], [17], which annotate bounding boxes for a small proportion of scenes.

For settings which require the virtual scenes, we conduct experiments on two versions of virtual scenes (from meshes/points), which are distinguished by subscripts M and P respectively. The virtual scenes generated with mesh information are named as mesh-version virtual scenes. Otherwise they are named as point-version virtual scenes.

Implementation Details: We set $N = 10000$, $A_{min} = 0.1m^2$, $H_{min} = 0.1m$, $\Delta_h = 0.02m$, $k = 16$, $\gamma = 3$, $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, $\lambda_3 = 1.0$, $\lambda_4 = 0.5$, $\lambda_5 = 0.5$, $\lambda_6 = 1.0$ and $r = 0.3$. The training hyperparameters (including max epochs, learning rates, weight decays, etc.) for WSB, WS3D, WSBP, BR and BR++ are set just the same as FSB for fair comparison.

During training of BR, as real scenes are more complicated, the converging of L_3 is much slower than L_2. Therefore we multiple L_2 by 0.1 to slow down the training on virtual scenes and stabilize the process of feature alignment. To better train our center refinement module, the global semantic features should not change rapidly. Therefore we first train the detector without L_1 until convergence, and then use the whole loss function to fine-tune the network. For GroupFree3D which has several decoders and each one outputs a stage of proposal features, we conduct feature alignment only for the last stage.

For differentiable label enhancement, we use Adam [68] with a stepwise scheduler to optimize the pose parameters. The initial learning rate is set to $1e-3$, with decay rate 0.9 and decay step every 10 epochs. For label-assisted self-training, the IoU threshold is 0.25 and the objectness threshold is set to 0.9 and 0.3 for VoteNet and GroupFree3D respectively. The student network is initialized with the parameters of WSBP. We do not initialize it with implicitly trained models to ensure it receives fully explicit supervision.

Different from previous works [8], [9], in ScanNet-md40 we need to detect small objects, such as bottle, cup and keyboard. As it is difficult for the network to extract high-quality features of these objects, we utilize a small object augmentation strategy to alleviate the problem, which is similar to [69]. Please refer to the supplementary for more details.

4.2 Results and Analysis

4.2.1 *Virtual Scene Evaluation*

We first evaluate the statistics of the generated virtual scenes by computing the average number of points of objects in each category in real scenes and virtual scenes. As the input point clouds are downsampled to a given number before fed into the network, we only care about the ratio of average point numbers of objects in each category as the numbers can be controlled by the downsampling scale. We demonstrate the results for ScanNet and Matterport3D in Fig. 9. It shows that the ratio in our virtual scenes is similar with that in the real scenes, which indicates the statistics of the virtual scenes are reasonable.

We also show qualitative visualizations to demonstrate our scene generation method in Fig. 10. We compare the mesh-version virtual scenes and the optimized ones generated by differentiable label enhancement with the real scenes. It is shown that the virtual scenes can largely preserve the layout of the real scenes, and the differentiable strategy makes the virtual scenes more reasonable and similar to the real scenes after the end-to-end optimization. Note that the virtual scenes are not optimized to imitate the real scenes everywhere, but to be similar with the real scenes in a general view and thus make the layouts more reasonable. For example, there are three lamps in the scene of the first row, one complete and two partial. After optimization, two of the lamps in the virtual scenes are scaled down to proper size. Although the correspondence is not one-to-one, the virtual scenes become more similar to the real ones overall.

4.2.2 *3D Object Detection Results*

Results on ScanNet: As shown in Table 1, with position-level 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 centers

(a) Real Scenes (b) Cared Objects (c) Virtual Scenes (d) Optimized Virtual Scenes

Fig. 10. The qualitative visualization results of our virtual scene generation. (c) and (d) are generated by shape-guided label enhancement and differentiable label enhancement methods respectively. In (b), (c) and (d), the same color 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, and the differentiable label enhancement method makes the virtual scenes more reasonable and similar to the real scenes after the end-to-end optimization.

and sizes of bounding boxes according to the scene context. WS3D makes use of some box annotations and achieve better performance. However, as it is specially designed for outdoor 3D 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 transferring, the improvement over the WSB is boosted to a higher level by BR. The above results shows each step in 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 BR improves a lot while WSBP has little change, which indicates that layout may not be that important in pretraining as in domain adaptation.

With the differentiable label enhancement and label-assisted self-training methods, BR++ further achieves significant improvement over BR and the final performance gap between FSB and BR++ (for GroupFree3D) is within 2%. The performance gap between mesh-version and point-version virtual scenes is also reduced, which shows the end-to-end optimization of virtual scenes and the detector closes the domain gap between real and virtual scenes and reduces the requirement for layout initialization.

In terms of class-specific results, on some categories the mAP@0.25 of the BR and BR++ (for GroupFree3D) even achieves the highest performance among all the methods including the FSB. However, all methods fail to precisely detect cup and bottle, which shows current 3D detectors still face huge challenges in small object detection. We also note the per-category performance of GroupFree3D is not stable, this is because GroupFree3D has 6 decode heads and we report the highest performance among all heads for each method.

Results on Matterport3D: As a more challenging dataset, we find advanced detectors which achieve high performance on ScanNet such as MLCVNet [65] and GroupFree3D [9] may fail to perform well on Matterport3D. Therefore we only conduct experiments based on VoteNet detector, as shown in Table 2.

Although FSB gets lower performance than that on ScanNet-md40 due to the more complex scene context, we find WSB and WS3D perform better, which is because the distribution of object size in Matterport3D has smaller variance and thus is closer to the average size. As the layouts of scenes are diverse, mesh-version virtual scenes do not perform significantly better than point-version ones. Note that though WSBP outperforms BR in terms of mAP@0.25, its mAP@0.5 is much lower according to Table 3 and thus BR is still a better solution. It can be seen that the final performance gap between FSB and BR++ is within 2% again in terms of mAP@0.25, which shows the effectiveness and generalization ability of our proposed methods.

Comprehensive Comparison: We further compare our methods with state-of-the-art semi-supervised methods for indoor 3D object detection, as shown in Table 3. Under the same level of annotating time (5s for weakly-supervised methods, 6s for semi-

TABLE 1
The class-specific detection results (mAP@0.25) of different weakly-supervised methods on ScanNet 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.)

	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 [8]	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† [12]	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
	$BR{+}{+}_P$(Ours)	51.4	82.8	11.7	41.3	0.0	80.5	0.1	25.0	58.2	20.3	17.8	0.7	34.7	0.9	56.4	67.6	3.3	86.1	18.8	50.5	94.8	1.6	36.6
	$BR{+}{+}_M$(Ours)	49.1	82.3	35.0	42.2	0.0	81.4	0.6	29.9	57.4	22.4	20.7	1.3	30.6	8.4	51.8	80.1	5.5	83.5	12.1	52.1	91.9	5.5	38.3
GroupFree3D	FSB [9]	**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† [12]	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
	$BR{+}{+}_P$(Ours)	81.0	87.5	16.9	37.4	0.0	82.8	0.0	48.6	50.2	52.4	60.2	**34.4**	53.5	**55.8**	80.2	76.6	5.7	90.2	41.5	45.8	97.2	8.6	50.3
	$BR{+}{+}_M$(Ours)	83.3	**99.9**	0.9	37.1	0.0	82.7	0.5	56.1	58.6	53.0	**72.9**	15.6	**69.1**	50.0	**83.4**	**92.5**	5.6	89.1	19.0	37.5	97.6	**53.6**	52.6

Fig. 11. Visual results on ScanNet. We compare the predictions of WSB, BR and BR++ after NMS with the ground-truth bounding boxes. BR can produce more accurate detection results with higher confidence, so after NMS the false positives are far less than WSB. BR++ detects all objects in the scene with more accurate box sizes than BR.

supervised ones), our approach surpasses all other methods by a large margin in terms of mAP@0.25 and mAP@0.5 on both benchmarks. In terms of mAP@0.25, BR++ even outperforms current semi-supervised methods with 3.6× less annotating time.

4.2.3 *Training Cost:*

We further analyze the training cost of BR and BR++. As there are two stages in BR++, we divide BR++ into BRDLE (differentiable label enhancement) and BRST (label-assisted self-training) and compare the training time of them with BR respectively. We train BR and BR++ with VoteNet as backbone and report the training time in Table 4. It can be seen that BRDLE is slower than BR due to there are additional pose parameters of objects to be optimized. BRST is faster than BR, as the teacher network is fixed and student network only needs to process real scenes. In terms of overall training time (on a 2-GPU machine), BR and BR++ take 11.2 and 16.2 hours respectively, which is acceptable. Moreover, BR++ provides several choices for the user. If training time is limited, user can only train BRDLE or first train BR then train BRST, which can still achieve great improvement upon BR.

4.2.4 *Visualization Results:*

We visualize the detection results of WSB, BR_M and $BR{+}{+}_M$ (for VoteNet) after NMS on ScanNet. As shown in Fig. 11, BR produces more accurate detection results with higher confidence, so after NMS the false positives are far less than WSB. BR++ detects all objects in the scene with more accurate box sizes than BR. The visualization further confirms the effectiveness of the proposed methods.

4.3 Ablation Study

In this section, we adopt VoteNet as the detector and conduct ablation experiments on the ScanNet dataset. Point-version virtual scenes and initialization are used for BR and BR++ respectively.

4.3.1 *Ablation Study for BR*

We design ablation experiments to study the influences of each scene generation step and each domain adaptation loss to the performance of our BR approach. Moreover, we further conduct robustness test to show how labeling error will influence the performance.

TABLE 2
The class-specific detection results (mAP@0.25) of different weakly-supervised methods on Matterport3D 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.)

	Setting	batht.	bed	bench	chair	curt.	desk	door	dres.	n.s.	sofa	stool	table	toil.	mAP@0.25
VoteNet	FSB [8]	91.6	**93.3**	5.9	64.0	**10.6**	18.8	**18.4**	4.3	**71.6**	67.7	10.6	**34.3**	66.6	**42.9**
	WSB	65.2	58.1	0.5	52.7	0.3	8.6	3.7	1.0	11.5	54.1	10.9	13.1	61.4	26.2
	WS3D† [12]	70.8	64.1	0.2	51.1	0.3	5.2	3.2	7.4	14.1	53.7	8.8	13.6	70.9	28.0
	WSBP_P	87.4	83.8	2.5	63.4	5.2	16.1	3.8	**9.4**	55.1	**70.4**	25.9	18.0	66.8	39.0
	WSBP_M	84.4	83.3	4.1	61.0	7.2	3.6	3.5	1.3	71.3	64.5	21.8	22.0	**73.1**	38.6
	BR_P(Ours)	83.9	72.4	2.7	58.6	1.8	9.9	10.1	1.6	61.2	64.7	14.6	24.6	69.2	36.6
	BR_M(Ours)	80.1	82.5	5.3	58.2	5.1	9.0	8.5	0.8	64.0	59.0	12.5	24.8	70.2	36.9
	BR++_P(Ours)	94.6	90.9	2.6	64.5	2.3	13.7	10.3	2.7	63.3	61.2	**26.5**	30.8	68.5	40.9
	BR++_M(Ours)	**96.8**	90.2	**6.0**	**65.1**	2.2	**23.1**	10.6	2.1	58.0	62.0	22.3	29.4	67.0	41.1

TABLE 3
The detection results (mAP@0.25 and mAP@0.5) and average annotating time per object of different methods. Annotating time 6s, 12s and 18s stand for labeling 5%, 10% and 15% scenes for semi-supervised methods. We adopt VoteNet as the detector and utilize mesh-version virtual scenes for BR and BR++.

	Method	Annotating Time	mAP @0.25	mAP @0.5
ScanNet	FSB [8]	120s	45.1	28.0
	WS3D [12]	5s	18.4	0.8
	SESS [16]	6s	22.8	9.2
		12s	28.1	12.8
		18s	34.5	20.3
	3DIoUMatch [17]	6s	30.9	16.5
		12s	34.9	22.3
		18s	38.2	25.2
	WSBP_M	5s	27.6	3.8
	BR_M(Ours)	5s	35.5	14.8
	BR++_M(Ours)	**5s**	**38.3**	**18.9**
Matterport3D	FSB [8]	120s	42.9	16.4
	WS3D [12]	5s	28.0	0.7
	SESS [16]	6s	21.1	3.5
		12s	26.4	7.3
		18s	31.8	12.1
	3DIoUMatch [17]	6s	26.2	8.7
		12s	32.7	13.9
		18s	34.6	14.8
	WSBP_M	5s	38.6	6.2
	BR_M(Ours)	5s	36.9	10.1
	BR++_M(Ours)	**5s**	**41.1**	**12.1**

TABLE 4
Total number of trianing epoches and training time of each epoch for different methods. $+60$ refers to the finetuning stage for training the center refinement module. We train all methods on two RTX 3090 GPU.

Methods	Stage	Epoch	Time (s)
BR	–	$180 + 60$	168.72
BR++	BRDLE	$180 + 60$	198.67
	BRST	90	116.52

In Table 5, we illustrate that in our virtual scene generation pipeline, the physical constraints and density control are effective. As the virtual scenes become more realistic, the performance of our BR approach is getting better. In Table 6, we show the effect of each domain adaptation module and the center 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 achieve higher detection accuracy. With the center refinement method, the performance is further boosted by 1.0%.

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

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

TABLE 6
The detection results (mAP@0.25) of BR with different domain adaptation modules on ScanNet. 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**

In our labeling strategy, the center error is within 10% of the object's size, which is defined as error rate. To show the robustness of our approach, we gradually increase this rate from 10% to 50% by randomly jittering the centers according to the box sizes, and report the detection results of WSB and BR in terms of mAP@0.25. As shown in Table 7, with the increasing of error rate, the performance of BR degrades more slowly than WSB. Even if the error rate is 50%, which allows us to label the centers in a more time-saving strategy, BR can still achieve satisfactory results (3.6% lower in terms of mAP@0.25).

4.3.2 *Ablation Study for BR++*

We design ablation experiments to study the effects of each component of differentiable label enhancement and label-assisted self-training on the performance of BR++.

TABLE 7
The detection results (mAP@0.25) of BR under different error rate for center labeling on ScanNet. We adopt VoteNet as the detector and utilize point-version virtual scenes for BR.

Method	Error Rate				
	10%	20%	30%	40%	50%
WSB	14.1	12.2	9.9	8.4	6.8
BR_P(Ours)	**31.2**	**30.3**	**29.4**	**28.7**	**27.6**

TABLE 8
The detection results (mAP@0.25) of BR++ (w/o label-assisted self-training) with all ablated versions of differentiable label enhancement on ScanNet. Here the detector is VoteNet and we initialize the pose parameters with point-version virtual scenes.

Method	mAP@0.25
Without optimization (remove L_D)	32.1
Optimize with $L_4 + L_5$ (instead of L_D)	31.8
Optimize with L_2 (instead of L_D)	27.9
Remove differentiable constraint	32.8
Without initialization	32.3
The full differentiable label enhancement	**33.6**

TABLE 9
The detection results (mAP@0.25) of BR++ with all ablated versions of label-assisted self-training on ScanNet. Here the detector is VoteNet and the teacher network is trained with point-version initialization.

Center Filter	Category Filter	IoU-based NMS	mAP@0.25
			34.2
✓			35.2
	✓		34.9
✓	✓		36.0
		✓	35.0
✓	✓	✓	**36.6**

To directly show the effects of each component in differentiable label enhancement, we report the detection results without label-assisted self-training in Table 8. The first row indicates that the online generation of virtual scenes is better than the offline generation in BR, which is because online generation provides one-to-one correspondence between real and virtual scenes and stabilizes the feature alignment. The second and third rows show the necessity of L_D, as L_2, L_4 and L_5 lack clear supervision for the virtual scenes. Moreover, L_2 significantly degrades the performance, that is because this loss term will hinder the detector to update parameters from false predictions. The fourth and fifth rows verify the effectiveness of the differentiable gravity constraint and the initialization of pose parameters.

In Table 9, we show the effect of each technique used in label-assisted self-training. The first row shows that even without the proposed techniques, the performance of the student network is still high than the teacher network (34.2 vs 33.6), which demonstrates the explicit box supervision is better than the implicit feature supervision. The following rows show that reusing the position-level annotations and virtual scenes for further filtering imprecise predictions is effective.

5 Conclusion and Discussion

In this paper, we have proposed a new training paradigm, namely Back to Reality (BR), for 3D object detection trained using only object centers and class tags as supervision. To fully exploit the information contained in the position-level annotations, we consider them as the coarse layout of scenes, which is utilized to assemble 3D shapes into fully-annotated virtual scenes. We apply physical constraints on the generated virtual scenes to make sure the relationship between objects is reasonable. In order to make use of the virtual scenes to remedy the information loss from box annotations to centers, we present a virtual-to-real domain adaptation method, which transfers the useful knowledge learned from the virtual scenes to real-scene 3D object detection. We have also presented BR++ with differentiable label enhancement and label-assisted self-training, which optimizes the virtual scenes end-to-end with the detector to close the domain gap in a data-driven manner, and reuses the position-level annotations to generate box-level pseudo labels as explicit supervision for the detector. Extensive experiments on ScanNet and Matterport3D dataset have shown that the performance of our approach surpasses the state-of-the-art weakly-supervised and semi-supervised methods for 3D object detection by a large margin, and we achieve comparable detection performance with some popular fully-supervised methods with less than 5% of the labeling labor.

Despite of the high data utilization efficiency, we are not able to provide more choice for the tradeoff of time and accuracy as semi-supervised methods do. In future work, combining position-level weakly-supervised methods and semi-supervised methods may be a promising way to better exploit the information from limited annotations. Moreover, the number of object categories in ModelNet40 is limited, which makes it hard to apply our method on a wider range of scenarios. Recently, OpenShape [70] proposes a text-3D model which is able to retrieve objects of any category from a Internet-level 3D shape database [71] given the text desciption. By further combining BR and OpenShape, we can acquire synthetic 3D shapes in any category and thus perform weakly-supervised 3D object detection on more classes.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62125603, and Grant U1813218, and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, Jian, "Faster r-cnn: towards real-time object detection with region proposal networks," *TPAMI*, vol. 39, no. 6, pp. 1137–1149, 2016. 1
[2] Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788. 1
[3] Zhou, Xingyi and Wang, Dequan and Krähenbühl, Philipp, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019. 1
[4] Qi, Charles R and Su, Hao and Mo, Kaichun and Guibas, Leonidas J, "Pointnet: Deep learning on point sets for 3d classification and segmentation." 1, 3
[5] Qi, Charles Ruizhongtai and Yi, Li and Su, Hao and Guibas, Leonidas J, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." 1, 3
[6] Shi, Shaoshuai and Wang, Xiaogang and Li, Hongsheng, "Pointrcnn: 3d object proposal generation and detection from point cloud." 1, 3
[7] Hou, Ji and Dai, Angela and Nießner, Matthias, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *CVPR*, 2019, pp. 4421–4430. 1, 3
[8] Qi, Charles R. and Litany, Or and He, Kaiming and Guibas, Leonidas J., "Deep hough voting for 3d object detection in point clouds," in *ICCV*, 2019, pp. 9277–9286. 1, 2, 3, 8, 9, 11, 12
[9] Liu, Ze and Zhang, Zheng and Cao, Yue and Hu, Han and Tong, Xin, "Group-free 3d object detection via transformers," *arXiv preprint arXiv:2104.00678*, 2021. 1, 2, 3, 8, 9, 10, 11
[10] Song, Shuran and Lichtenberg, Samuel P and Xiao, Jianxiong, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *CVPR*, 2015, pp. 567–576. 1
[11] Ren, Zhongzheng and Misra, Ishan and Schwing, Alexander G and Girdhar, Rohit, "3d spatial recognition without spatially labeled 3d," in *CVPR*, 2021, pp. 13 204–13 213. 1, 2, 3
[12] "Qinghao Meng and Wenguan Wang and Tianfei Zhou and Jianbing Shen and Luc Van Gool and Dengxin Dai", "Weakly supervised 3d object detection from lidar point cloud," in *ECCV*, 2020, pp. 515–531. 1, 2, 3, 9, 11, 12

[13] Meng, Qinghao and Wang, Wenguan and Zhou, Tianfei and Shen, Jianbing and Jia, Yunde and Van Gool, Luc, "Towards a weakly supervised framework for 3d point cloud object detection and annotation," *TPAMI*, 2021. 1, 2, 3

[14] Xu, Ning and Liu, Yun-Peng and Geng, Xin, "Label enhancement for label distribution learning," *TKDE*, 2019. 1

[15] Xu, Xiuwei and Wang, Yifan and Zheng, Yu and Rao, Yongming and Zhou, Jie and Lu, Jiwen, "Back to reality: Weakly-supervised 3d object detection with shape-guided label enhancement," in *CVPR*, 2022. 2

[16] Zhao, Na and Chua, Tat-Seng and Lee, Gim Hee, "Sess: Self-ensembling semi-supervised 3d object detection," in *CVPR*, 2020, pp. 11 079–11 087. 2, 3, 9, 12

[17] Wang, He and Cong, Yezhen and Litany, Or and Gao, Yue and Guibas, Leonidas J, "3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection," in *CVPR*, 2021, pp. 14 615–14 624. 2, 3, 8, 9, 12

[18] Li, Yangyan and Dai, Angela and Guibas, Leonidas and Nießner, Matthias, "Database-assisted object retrieval for real-time 3d reconstruction," in *CGF*, vol. 34, no. 2, 2015, pp. 435–446. 2, 3

[19] Litany, Or and Remez, Tal and Freedman, Daniel and Shapira, Lior and Bronstein, Alex and Gal, Ran, "Asist: automatic semantically invariant scene transformation," *CVIU*, vol. 157, pp. 284–299, 2017. 2

[20] Song, Shuran and Xiao, Jianxiong, "Sliding shapes for 3d object detection in depth images," in *ECCV*, 2014, pp. 634–651. 2

[21] ——, "Deep sliding shapes for amodal 3d object detection in rgb-d images," in *CVPR*, 2016, pp. 808–816. 2

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

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

[24] Chen, Xiaozhi and Kundu, Kaustav and Zhang, Ziyu and Ma, Huimin and Fidler, Sanja and Urtasun, Raquel, "Monocular 3d object detection for autonomous driving," in *CVPR*, 2016, pp. 2147–2156. 3

[25] Lahoud, Jean and Ghanem, Bernard, "2d-driven 3d object detection in rgb-d images," in *ICCV*, 2017, pp. 4622–4630. 3

[26] J. Gwak, C. Choy, and S. Savarese, "Generative sparse detection networks for 3d single-shot object detection," in *ECCV*. Springer, 2020, pp. 297–313. 3

[27] Qin, Zengyi and Wang, Jinglu and Lu, Yan, "Weakly supervised 3d object detection from point clouds," in *ACM MM*, 2020, pp. 4144–4152. 3

[28] Tarvainen, Antti and Valpola, Harri, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017. 3

[29] Avetisyan, Armen and Dahnert, Manuel and Dai, Angela and Savva, Manolis and Chang, Angel X and Nießner, Matthias, "Scan2cad: Learning cad model alignment in rgb-d scans," in *CVPR*, 2019, pp. 2614–2623. 3

[30] Avetisyan, Armen and Dai, Angela and Nießner, Matthias, "End-to-end cad model retrieval and 9dof alignment in 3d scans," in *ICCV*, 2019, pp. 2551–2560. 3

[31] Dahnert, Manuel and Dai, Angela and Guibas, Leonidas J and Niessner, Matthias, "Joint embedding of 3d scan and cad objects," in *ICCV*, 2019, pp. 8749–8758. 3

[32] Ritchie, Daniel and Wang, Kai and Lin, Yu-an, "Fast and flexible indoor scene synthesis via deep convolutional generative models," in *CVPR*, 2019, pp. 6182–6190. 3

[33] Wang, Kai and Savva, Manolis and Chang, Angel X and Ritchie, Daniel, "Deep convolutional priors for indoor scene synthesis," *TOG*, vol. 37, no. 4, pp. 1–14, 2018. 3

[34] Qi, Siyuan and Zhu, Yixin and Huang, Siyuan and Jiang, Chenfanfu and Zhu, Song-Chun, "Human-centric indoor scene synthesis using stochastic grammar," in *CVPR*, 2018, pp. 5899–5908. 3

[35] Dosovitskiy, Alexey and Fischer, Philipp and Ilg, Eddy and Hausser, Philip and Hazirbas, Caner and Golkov, Vladimir and Van Der Smagt, Patrick and Cremers, Daniel and Brox, Thomas, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766. 3

[36] Wang, He and Sridhar, Srinath and Huang, Jingwei and Valentin, Julien and Song, Shuran and Guibas, Leonidas J, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *CVPR*, 2019, pp. 2642–2651. 3

[37] Liu, Xingyu and Qi, Charles R. and Guibas, Leonidas J., "Flownet3d: Learning scene flow in 3d point clouds," in *CVPR*, 2019, pp. 529–537. 3

[38] Peng, Songyou and Niemeyer, Michael and Mescheder, Lars and Pollefeys, Marc and Geiger, Andreas, "Convolutional occupancy networks," in *ECCV*, 2020, pp. 523–540. 3

[39] Rao, Yongming and Liu, Benlin and Wei, Yi and Lu, Jiwen and Hsieh, Cho-Jui and Zhou, Jie, "Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection," in *ICCV*, 2021, pp. 3283–3292. 3

[40] Dai, Angela and Nießner, Matthias and Zollhöfer, Michael and Izadi, Shahram and Theobalt, Christian, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *TOG*, vol. 36, no. 4, p. 1, 2017. 3

[41] Goodfellow, Ian J and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014. 3

[42] Li, Manyi and Patil, Akshay Gadi and Xu, Kai and Chaudhuri, Siddhartha and Khan, Owais and Shamir, Ariel and Tu, Changhe and Chen, Baoquan and Cohen-Or, Daniel and Zhang, Hao, "Grains: Generative recursive autoencoders for indoor scenes," *TOG*, vol. 38, no. 2, pp. 1–16, 2019. 3

[43] Wang, Kai and Lin, Yu-An and Weissmann, Ben and Savva, Manolis and Chang, Angel X and Ritchie, Daniel, "Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks," *TOG*, vol. 38, no. 4, pp. 1–15, 2019. 3

[44] A. Kar, A. Prakash, M.-Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler, "Meta-sim: Learning to generate synthetic datasets," in *ICCV*, 2019. 3

[45] J. Devaranjan, A. Kar, and S. Fidler, "Meta-sim2: Learning to generate synthetic datasets," in *ECCV*, 2020. 3

[46] Long, Mingsheng and Cao, Yue and Cao, Zhangjie and Wang, Jianmin and Jordan, Michael I, "Transferable representation learning with deep adaptation networks," *TPAMI*, vol. 41, no. 12, pp. 3071–3085, 2018. 3

[47] Kang, Guoliang and Jiang, Lu and Yang, Yi and Hauptmann, Alexander G, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019, pp. 4893–4902. 3

[48] Rozantsev, Artem and Salzmann, Mathieu and Fua, Pascal, "Beyond sharing weights for deep domain adaptation," *TPAMI*, vol. 41, no. 4, pp. 801–814, 2018. 3

[49] Xie, Shaoan and Zheng, Zibin and Chen, Liang and Chen, Chuan, "Learning semantic representations for unsupervised domain adaptation," in *ICML*. PMLR, 2018, pp. 5423–5432. 3

[50] Pan, Yingwei and Yao, Ting and Li, Yehao and Wang, Yu and Ngo, Chong-Wah and Mei, Tao, "Transferrable prototypical networks for unsupervised domain adaptation," in *CVPR*, 2019, pp. 2239–2247. 3

[51] Qin, Can and You, Haoxuan and Wang, Lichen and Kuo, C-C Jay and Fu, Yun, "Pointdan: A multi-scale 3d domain adaption network for point cloud representation," in *Advances in Neural Information Processing Systems*, 2019, pp. 7190–7201. 3

[52] Jaritz, Maximilian and Vu, Tuan-Hung and Charette, Raoul de and Wirbel, Emilie and Pérez, Patrick, "xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation," in *CVPR*, 2020, pp. 12 605–12 614. 3

[53] Yi, Li and Gong, Boqing and Funkhouser, Thomas, "Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds," in *CVPR*, 2021, pp. 15 363–15 373. 3

[54] Wang, Yan and Chen, Xiangyu and You, Yurong and Li, Li Erran and Hariharan, Bharath and Campbell, Mark and Weinberger, Kilian Q and Chao, Wei-Lun, "Train in germany, test in the usa: Making 3d object detectors generalize," in *CVPR*, 2020, pp. 11 713–11 723. 3

[55] Saltori, Cristiano and Lathuiliére, Stéphane and Sebe, Nicu and Ricci, Elisa and Galasso, Fabio, "Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection," in *3DV*. IEEE, 2020, pp. 771–780. 3

[56] Zhang, Weichen and Li, Wen and Xu, Dong, "Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection," in *CVPR*, 2021, pp. 6769–6779. 3

[57] Luo, Zhipeng and Cai, Zhongang and Zhou, Changqing and Zhang, Gongjie and Zhao, Haiyu and Yi, Shuai and Lu, Shijian and Li, Hongsheng and Zhang, Shanghang and Liu, Ziwei, "Unsupervised domain adaptive 3d detection with multi-level consistency," in *ICCV*, 2021, pp. 8866–8875. 3

[58] Karpathy, Andrej and Miller, Stephen and Fei-Fei, Li, "Object discovery in 3d scenes via shape analysis," in *ICRA*, 2013, pp. 2088–2095. 4

[59] Ganin, Yaroslav and Lempitsky, Victor, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015, pp. 1180–1189. 6

[60] Saito, Kuniaki and Ushiku, Yoshitaka and Harada, Tatsuya and Saenko, Kate, "Strong-weak distribution alignment for adaptive object detection," in *CVPR*, 2019, pp. 6956–6965. 6

[61] Lin, Tsung-Yi and Goyal, Priya and Girshick, Ross and He, Kaiming and Dollár, Piotr, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988. 6

[62] Wu, Zhirong and Song, Shuran and Khosla, Aditya and Yu, Fisher and Zhang, Linguang and Tang, Xiaoou and Xiao, Jianxiong, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920. 8

[63] Dai, Angela and Chang, Angel X. and Savva, Manolis and Halber, Maciej and Funkhouser, Thomas and Nießner, Matthias, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017, pp. 5828—-5839. 8

[64] Chang, Angel and Dai, Angela and Funkhouser, Thomas and Halber, Maciej and Niessner, Matthias and Savva, Manolis and Song, Shuran and Zeng, Andy and Zhang, Yinda, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017. 8

[65] Xie, Qian and Lai, Yu-Kun and Wu, Jing and Wang, Zhoutao and Zhang, Yiming and Xu, Kai and Wang, Jun, "Mlcvnet: Multi-level context votenet for 3d object detection," in *CVPR*, 2020, pp. 10 447–10 456. 8, 10

[66] Zhang, Zaiwei and Sun, Bo and Yang, Haitao and Huang, Qixing, "H3dnet: 3d object detection using hybrid geometric primitives," in *ECCV*, 2020, pp. 311–329. 8

[67] Griffiths, David and Boehm, Jan and Ritschel, Tobias, "Finding your (3d) center: 3d object detection using a learned loss," in *ECCV*. Springer, 2020, pp. 70–85. 8

[68] Kingma, Diederik P and Ba, Jimmy, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 9

[69] Kisantal, Mate and Wojna, Zbigniew and Murawski, Jakub and Naruniec, Jacek and Cho, Kyunghyun, "Augmentation for small object detection," *arXiv preprint arXiv:1902.07296*, 2019. 9

[70] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *arXiv preprint arXiv:2305.10764*, 2023. 13

[71] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153. 13

Xiuwei Xu He received the B.Eng degree from Tsinghua University in 2021. He is currently a PhD candidate in the Department of Automation at Tsinghua University. His research interests lie data/computation-efficient learning, 3D vision and robotic systems.

Ziwei Wang received the B.S. degree from the Department of Physics, Tsinghua University, China, in 2018. He is currently working toward the PhD degree in the Department of Automation, Tsinghua University, China. His research interests include tiny machine learning, scene understanding and robotic vision. He has published more than 10 scientific papers in TPAMI, CVPR, ICCV, ECCV and IROS. He serves as a regular reviewer member for TIP, TCSVT, CVPR, ICCV, ECCV, NeurIPS, ICML, ICLR, WACV and ICME.

Jie Zhou (M'01-SM'04) received the BS and MS degrees both from the Department of Mathematics, Nankai University, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), China, in 1995. From then to 1997, he served as a postdoctoral fellow in the Department of Automation, Tsinghua University, China. Since 2003, he has been a full professor in the Department of Automation, Tsinghua University. His research interests include computer vision and pattern recognition. In recent years, he has authored more than 100 papers have been published in TPAMI, TIP and CVPR. He is an associate editor for TPAMI and two other journals. He received the National Outstanding Youth Foundation of China Award. He is a senior member of the IEEE and Fellow of the IAPR.

Jiwen Lu (M'11-SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, and a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He serves as the Co-Editor-of-Chief for PRL, an Associate Editor for the TIP, TCSVT, the TBIOM, and Pattern Recognition. He also serves as the Program Co-Chair of IEEE FG'2023, VCIP'2022, AVSS'2021 and ICME'2020. He received the National Outstanding Youth Foundation of China Award. He is an IAPR Fellow.