DODA: Data-Oriented Sim-to-Real Domain Adaptation for 3D Semantic Segmentation

Runyu Ding[1], Jihan Yang[1], Li Jiang[2], and Xiaojuan Qi[1(✉)]

[1] The University of Hong Kong, Hong Kong, China
{ryding,jhyang,xjqi}@eee.hku.hk
[2] MPI for Informatics, Saarbürcken, Germany
lijiang@mpi-inf.mpg.de

Abstract. Deep learning approaches achieve prominent success in 3D semantic segmentation. However, collecting densely annotated real-world 3D datasets is extremely time-consuming and expensive. Training models on synthetic data and generalizing on real-world scenarios becomes an appealing alternative, but unfortunately suffers from notorious domain shifts. In this work, we propose a **D**ata-**O**riented **D**omain **A**daptation (DODA) framework to mitigate pattern and context gaps caused by different sensing mechanisms and layout placements across domains. Our DODA encompasses virtual scan simulation to imitate real-world point cloud patterns and tail-aware cuboid mixing to alleviate the interior context gap with a cuboid-based intermediate domain. The first unsupervised sim-to-real adaptation benchmark on 3D indoor semantic segmentation is also built on 3D-FRONT, ScanNet and S3DIS along with 8 popular Unsupervised Domain Adaptation (UDA) methods. Our DODA surpasses existing UDA approaches by over 13% on both 3D-FRONT → ScanNet and 3D-FRONT → S3DIS. Code is available at https://github.com/CVMI-Lab/DODA.

Keywords: Domain adaptation · 3D semantic segmentation

1 Introduction

3D semantic segmentation is a fundamental perception task receiving incredible attention from both industry and academia due to its wide applications in robotics, augmented reality, and human-computer interaction, to name a

R. Ding and J. Yang—Equal contribution.

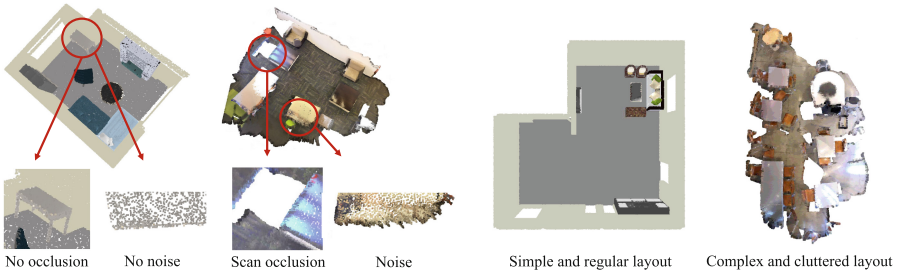
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19812-0_17.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13687, pp. 284–303, 2022.
https://doi.org/10.1007/978-3-031-19812-0_17

few. Data hungry deep learning approaches have attained remarkable success for 3D semantic segmentation [22,30,46,55,61,63]. Nevertheless, harvesting a large amount of annotated data is expensive and time-consuming [3,7].

An appealing avenue to overcome such data scarcity is to leverage simulation data where both data and labels can be obtained for free. Simulated datasets can be arbitrarily large, easily adapted to different label spaces and customized for various usages [8,18,26,34,53,72]. However, due to notorious domain gaps in point patterns and context (see Fig. 1), models trained on simulated scenes suffer drastic performance degradation when generalized to real-world scenarios. This motivates us to study sim-to-real unsupervised domain adaptation (UDA), leveraging labeled source data (simulation) and unlabeled target data (real) for effectively adapting knowledge across domains.

Recent efforts on 3D domain adaptation for outdoor scene parsing have obtained considerable progress [23,29,60,71]. However, they often adopt LiDAR-specific range image format, not applicable for indoor scenarios with scenes constructed by RGB-D sequences. Besides, such outdoor attempts could be sub-optimal in addressing the indoor domain gaps raised from different scene construction processes. Further, indoor scenes have more sophisticated interior context than outdoor, which makes the context gap a more essential issue in indoor settings. Here, we explore sim-to-real UDA in the 3D indoor scenario which is challenging and largely under explored.



(a) Point pattern gap between simulated scene (left) and real-world scene (right).

(b) Context-level gap between simulated scene (left) and real-world scene (right).

Fig. 1. The domain gaps between simulated scenes from 3D-FRONT [8] and real-world scenes from ScanNet [7]. (a): The point pattern gap. The simulated scene is perfect without occlusions or noise, while the real-world scene inevitably contains scan occlusion and noise patterns such as rough surfaces. (b): The context gap. While the simulated scene applies simple layout with regularly placed objects, the real scene is complex with cluttered interiors.

Challenges. Our empirical studies on sim-to-real adaptation demonstrate two unique challenges in this setting: the *point pattern gap* owing to different sensing mechanisms, and the *context gap* due to dissimilar semantic layouts. As shown in Fig. 1(a), simulated scenes tend to contain complete objects as well as smooth

surfaces, while real scenes include inevitable scan occlusions and noise patterns during reconstructing point clouds from RGB-D videos captured by depth cameras [3,7]. Also, even professionally designed layouts in simulated scenes are much simpler and more regular than real layouts as illustrated in Fig. 1(b).

To tackle the above domain gaps, we develop a holistic two stage pipeline DODA with a pretrain and a self-training stage, which is widely proved to be effective in UDA settings [49,64,73]. As the root of the challenges lies in "data", we thus design two data-oriented modules which are shown to dramatically reduce domain gaps without incurring any computational costs during inference. Specifically, we develop Virtual Scan Simulation (VSS) to mimic occlusion and noise patterns that occur during the construction of real scenes. Such pattern imitation yields a more transferable model to real-world data. Afterwards, to adapt the model to target domain, we design Tail-aware Cuboid Mixing (TACM) for boosting self-training. While source supervision is utilized to stabilize gradients with clean labels in self-training, it unfortunately introduces context bias. Thus, we propose TACM to create an intermediate domain by splitting, permuting, mixing and re-sampling source and target cuboids, which explicitly mitigates the context gap through breaking and rectifying source bias with target pseudo-labeled data, and simultaneously eases long-tail issue by oversampling tail cuboids.

To the best of our knowledge, we are the first to explore unsupervised domain adaptation on 3D indoor semantic segmentation. To verify the effectiveness of our DODA, we construct the first 3D indoor sim-to-real UDA benchmark on a simulated dataset 3D-FRONT [8] and two widely used real-world scene understanding datasets ScanNet [7] and S3DIS [3] along with 8 popular UDA methods with task-specific modifications as our baselines. Experimental results show that DODA obtains 22% and 19% performance gains in terms of mIoU compared to source only model on 3D-FRONT $\rightarrow$ ScanNet and 3D-FRONT $\rightarrow$ S3DIS respectively. Even compared to existing UDA methods, over 13% improvement is still achieved. It is also noteworthy that the proposed VSS can lift previous UDA methods by a large margin (8% $\sim$ 14%) as a plug-and-play data augmentation, and TACM further facilitates real-world cross-site adaptation tasks with 4% $\sim$ 5% improvements.

2 Related Work

3D Indoor Semantic Segmentation focuses on obtaining point-wise category predictions from point clouds, which is a fundamental while challenging task due to the irregularity and sparsity of 3D point clouds. Some previous works [41,53] feed 3D grids constructed from point clouds into 3D convolutional neural networks. Some approaches [6,16] further employ sparse convolution [17] to leverage the sparsity of 3D voxel representation to accelerate computation. Another line of works [25,45,46,61,70] directly extract feature embeddings from raw point clouds with hierarchical feature aggregation schemes. Recent methods [55,63] assign position-related kernel functions on local point areas to perform dynamic convolutions. Additionally, graph-based works [31,52,59] adopt

graph convolutions to mimic point cloud structure for point representation learning. Although the above methods achieve prominent performance on various indoor scene datasets, they require large-scale human-annotated datasets which we aim to address using simulation data. Our experimental investigation is built upon the sparse-convolution-based U-Net [6,16] due to its high performance.

Unsupervised Domain Adaptation aims at adapting models obtained from annotated source data towards unlabeled target samples. The annotation efficiency of UDA and existing data-hungry deep neural networks make it receive great attention from the computer vision community. Some previous works [38,39] attempt to learn domain-invariant representations by minimizing maximum mean discrepancy [5]. Another line of research leverages adversarial training [14] to align distributions in feature [10,20,50], pixel [13,20,21] or output space [56] across domains. Adversarial attacks [15] have also been utilized in [35,66] to train domain-invariant classifiers. Recently, Self-training has been investigated in addressing this problem [49] which formulate UDA as a supervised learning problem guided by pseudo-labeled target data and achieves state-of-the-art performance in semantic segmentation [73] and object detection [28,48].

Lately, with the rising of 3D vision tasks, UDA has also attracted a lot of attention in such 3D tasks as 3D object classification [1,47], 3D outdoor semantic segmentation [23,29,44,60,67] and 3D outdoor object detection [40,58,64,65,69]. Especially, Wu *et al.* [60] propose intensity rendering, geodesic alignment and domain calibration modules to align sim-to-real gaps of outdoor 3D semantic segmentation datasets. Jaritz *et al.* [23] explore multi-modality UDA by leveraging images and point clouds simultaneously. Nevertheless, no previous work studies UDA on 3D indoor scenes. The unique point pattern gap and the context gap also render 3D outdoor UDA approaches not readily applicable to indoor scenarios. Hence, in this work, we make the first attempt on UDA for 3D indoor semantic segmentation. Particularly, we focus on the most practical and challenging scenario – simulation to real adaptation.

Data Augmentation for UDA has also been investigated to remedy data-level gaps across domains. Data augmentation techniques have been widely employed to construct an intermediate domain [13,29,48] to benefit optimization and facilitate gradual domain adaptation. However, they mainly focus on image-like input formats, which is not suitable for sparse and irregular raw 3D point clouds. Different from existing works, we build a holistic pipeline with two data-oriented modules on two stages to manipulate raw point clouds for mimicking target point cloud patterns and creating a cuboid-based intermediate domain.

3 Method

3.1 Overview

In this work, we aim at adapting a 3D semantic scene parsing model trained on a source domain $\mathcal{D}_s = \{(P_i^s, Y_i^s)\}_{i=1}^{N_s}$ of N_s samples to an unlabeled target

domain $\mathcal{D}_t = \{P_i^t\}_{i=1}^{N_t}$ of N_t samples. P and Y represent the point cloud and the point-wise semantic labels respectively.

In this section, we present DODA, a data-oriented domain adaptation framework to simultaneously close pattern and context gaps by imitating target patterns as well as breaking source bias with the generated intermediate domain. Specifically, as shown in Fig. 2, DODA begins with pretraining the 3D scene parsing model F on labeled source data with our proposed virtual scan simulation module for better generalization. VSS puts virtual cameras on the feasible regions in source scenes to simulate occlusion patterns, and jitters source points to imitate sensing and reconstruction noise in the real scenes. The pseudo labels are then generated with the pretrained model. In the self-training stage, we develop tail-aware cuboid mixing to build an intermediate domain between source and target, which is constructed by splitting and mixing cuboids from both domains. Besides, cuboids including high percentage tail classes are over-sampled to overcome the class imbalance issue during learning with pseudo labeled data. Elaborations of our tailored VSS and TACM are presented in the following parts.

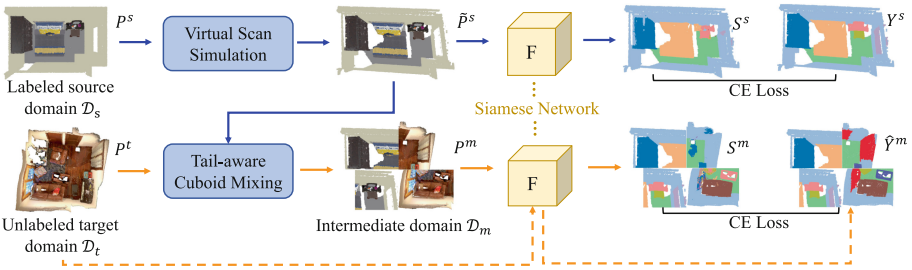


Fig. 2. Our DODA framework consists of two data-oriented modules: Virtual Scan Simulation (VSS) and Tail-aware Cuboid Mixing (TACM). VSS mimics real-world data patterns and TACM constructs an intermediate domain through mixing source and target cuboids. P denotes the point cloud; Y denotes the semantic labels and $\hat{Y}$ denotes the pseudo labels. The superscripts s, t and m stand for source, target and intermediate domain, respectively. The blue line denotes source training flow; the orange line denotes target training flow and the orange dotted line denotes target pseudo label generation procedure. Best viewed in color. (Color figure online)

3.2 Virtual Scan Simulation

DODA starts from training a 3D scene parsing network on labeled source data, to provide pseudo labels on the target domain in the next self-training stage. Hence, a model with a good generalization ability is highly desirable. As analyzed in Sect. 1, different scene construction procedures cause point pattern gaps across domains, significantly hindering the transferability of source-trained models. Specifically, we find that the missing of occlusion patterns and sensing or

reconstruction noise in simulation scenes raises huge negative transfer during the adaptation, which cannot be readily addressed by previous UDA methods (see Sect. 5). This is potentially caused by the fact that models trained on clean source data are incapable of extracting useful features to handle real-world challenging scenarios with ubiquitous occlusions and noise. To this end, we propose a plug-and-play data augmentation technique, namely virtual scan simulation, to imitate camera scanning procedure for augmenting the simulation data.

VSS includes two parts: the occlusion simulation that puts virtual cameras in feasible regions of simulated scenes to imitate occlusions in the scanning process, and the noise simulation that randomly jitters point patterns to mimic sensing or reconstruction errors, through which the pattern gaps are largely bridged.

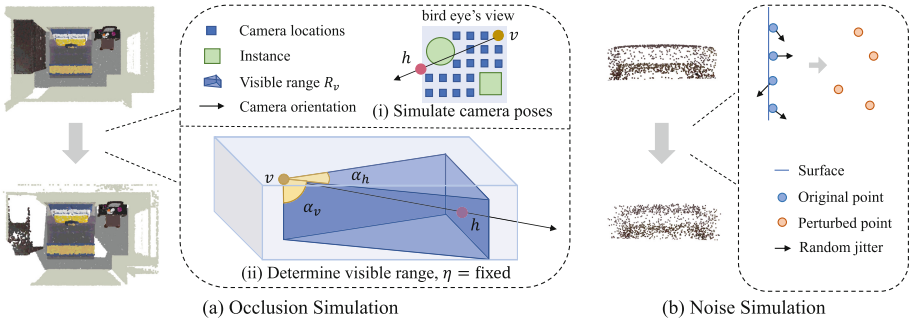


Fig. 3. Virtual scan simulation. (a): We simulate occlusion patterns by simulating camera poses and determining visible ranges. (b): We simulate noise by randomly jittering points to generate realistic irregular point patterns such as rough surfaces.

Occlusion Simulation. Scenes in real-world datasets are reconstructed from RGB-D frame sequences suffering from inevitable occlusions, while simulated scenes contain complete objects without any hidden points. We attempt to mimic occlusion patterns on the simulation data by simulating the real-world data acquisition procedures. Specifically, we divide it into the following three steps:

a) Simulate camera poses. To put virtual cameras in a given simulation scene, we need to determine camera poses including camera positions and camera orientations. First, feasible camera positions where a handheld camera can be placed are determined by checking free space in the simulated environment. We voxelize and project P^s to bird eye's view and remove voxels containing instance or room boundary. The centers of remaining free-space voxels are considered as feasible x-y coordinates for virtual cameras, as shown in Fig. 3(a) (i). For the z axis, we randomly sample the camera height in the top half of the room.

Second, for each camera position v, we randomly generate a camera orientation using the direction from the camera position v to a corresponding randomly

sampled point of interest h on the wall, as shown in Fig. 3(a) (i). This ensures that simulated camera orientations are uniformly distributed among all potential directions without being influenced by scene-specific layout bias.

b) Determine visible range. Given a virtual camera pose and a simulated 3D scene, we are now able to determine the spatial range that the camera can cover, *i.e.*, R_v, which is determined by the camera field of view (FOV) (see Fig. 3(a) (ii)). To ease the modeling difficulties, we decompose FOV into the horizontal viewing angle α_h, the vertical viewing angle α_v and the viewing mode η that determine horizontal range, vertical range and the shape of viewing frustum, respectively. For the viewing mode η, we approximate three versions from simple to sophisticated, namely fixed, parallel and perspective, with details presented in the supplementary materials. As illustrated in Fig. 3(a) (ii), we show an example of the visible range R_v with random α_h and α_v and η in the fixed mode.

c) Determine visible points. After obtaining the visible range R_v, we then determine the visibility of each point within R_v. Specifically, we convert the point cloud to the camera coordinate and extend [27] with spherical projection to filter out occluded points and obtain visible points. By taking the union of visible points from all virtual cameras, we finally obtain the point set P_v^s with occluded points removed. Till now, we can generate occlusion patterns in simulation scenes by mimicking real-world scanning process and adjust the intensity of occlusion by changing the number of camera positions n_v and FOV configurations to ensure that enough semantic context is covered for model learning.

Noise Simulation. Besides occlusion patterns, sensing and reconstruction errors are unavoidable when generating 3D point clouds from sensor-captured RGB-D videos, which unfortunately results in non-uniform distributed points and rough surfaces in real-world datasets (See Fig. 1(a)). To address this issue, we equip our VSS with another noise simulation module, which injects perturbations to each point as follows:

$$\tilde{P}^s = \{p + \Delta p \mid p \in P_v^s\}, \tag{1}$$

where Δ_p denotes the point perturbation following a uniform distribution ranging from $-\delta_p$ to δ_p, and $\tilde{P}^s$ is the perturbed simulation point cloud. Though simple, we argue that this module efficiently imitates the noise in terms of non-uniform and irregular points patterns as illustrated in Fig. 3.

Model Pretraining on Source Data. By adopting VSS as a data augmentation for simulated data, we train a model with cross-entropy loss as Eq. (2) following settings in [24,37].

$$\min \mathcal{L}_{pre} = \sum_{i=1}^{N_s} \mathrm{CE}(S_i^s, Y_i^s), \tag{2}$$

where $\mathrm{CE}(\cdot,\cdot)$ is the cross-entropy loss and S is the predicted semantic scores after performing *softmax* on logits.

3.3 Tail-aware Cuboid Mixing

After obtaining a more transferable scene parsing model with VSS augmentation, we further adopt self-training [32,54,62,69,73], to adapt the model by directly utilizing target pseudo-labeled data for supervision. Since target pseudo label is rather noisy, containing incorrect pseudo labeled data and leading to erroneous supervisions [65], we also introduce source supervision to harvest its clean annotations and improve the percentage of correct labels. However, directly utilizing source data unfortunately brings source bias and large discrepancies in joint optimization. Even though point pattern gaps have already been alleviated with the proposed VSS, the model still suffers from the context gap due to different scene layouts. Fortunately, the availability of target domain data gives us the chance to rectify such context gaps. To this end, we design Tail-aware Cuboid Mixing (TACM) to construct an intermediate domain $\mathcal{D}_m$ that combines source and target cuboid-level patterns (see Fig. 4), which augments and rectifies source layouts with target domain context. Besides, it also decreases the difficulty of simultaneously optimizing source and target domains with huge distribution discrepancies by providing a bridge for adaptation. TACM further moderates the pseudo label class imbalance issue by cuboid-level tail class oversampling. Details on pseudo labeling, cuboid mixing and tail cuboid oversampling are as follows.

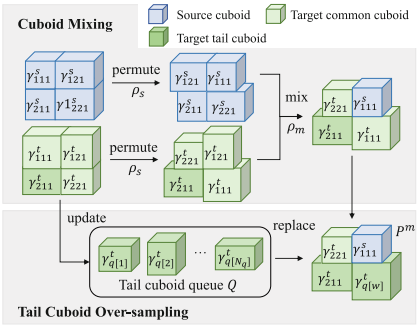


Fig. 4. An illustration of tail-aware cuboid mixing, which contains cuboid mixing and tail cuboid over-sampling. Notice that for clarity, we take $(n_x, n_y, n_z) = (2, 2, 1)$ as an example.

Pseudo Label Generation. To employ self-training after pretraining, we first need to generate pseudo labels $\hat{Y}^t$ for target scenes P^t. Similar to previous paradigms [24,62,65,73], we obtain pseudo labels via the following equation:

$$\hat{Y}^t_{i,j} = \begin{cases} 1\ , \textbf{if}\ \max(S^t_i) > T, j = \arg\max S^t_i, \\ 0\ , \text{otherwise}, \end{cases} \tag{3}$$

where $\hat{Y}^t_i = [\hat{Y}^t_{i,1}, \cdots, \hat{Y}^t_{i,c}]$, c is the number of classes and T is the confidence threshold to filter out uncertain predictions.

Cuboid Mixing. Here, given labeled source data and pseudo-labeled target data, we carry out the cuboid mixing to construct a new intermediate domain

$\mathcal{D}_m$ as shown in Fig. 2 and Fig. 4. For each target scene, we randomly sample a source scene to perform cuboid mixing. We first partition two scenes into several cuboids with varying sizes as the smallest units to mix cuboid as Eq. (4):

$$\begin{aligned} P = \{\gamma_{ijk}\}, i \in \{1, ..., n_x\}, j \in \{1, ..., n_y\}, k \in \{1, ..., n_z\}, \\ \gamma_{ijk} = \{p \mid p \text{ in } [x_{i-1}, y_{j-1}, z_{k-1}, x_i, y_j, z_k]\}, \end{aligned} \tag{4}$$

where γ_{ijk} denotes a single cuboid; n_x, n_y and n_z stand for the number of partitions in x, y and z axis, respectively; and each cuboid γ_{ijk} is constrained in a six-tuple bounding box $[x_{i-1}, y_{j-1}, z_{k-1}, x_i, y_j, z_k]$ defined by the partition positions x_i, y_j, z_k for corresponding dimensions, respectively. These partition positions are first initialized as equal-divisions and then injected with randomness to enhance diversities as below:

$$x_i = \begin{cases} \frac{i}{n_x} \max p_x + (1 - \frac{i}{n_x}) \min p_x, & \textbf{if } i \in \{0, n_x\}, \\ \frac{i}{n_x} \max p_x + (1 - \frac{i}{n_x}) \min p_x + \Delta\phi, & \text{otherwise}, \end{cases} \tag{5}$$

where $\Delta\phi$ is the random perturbation following uniform distribution ranging from $-\delta_\phi$ to δ_ϕ. The same formulation is also adopted for y_j and z_k. After partitioning, the source and target cuboids are first spatially permuted with a probability ρ_s and then randomly mixed with another probability ρ_m, as depicted in Fig. 4 and Fig. 2.

Though ConDA [29] shares some similarities with our cuboid mixing by mixing source and target, it aims to preserve cross-domain context consistency while ours attempts to mitigate context gaps. Besides, ConDA operates on 2D range images, inapplicable to reconstructed indoor scenes obtained by fusing depth images. Our cuboid mixing leverages the freedom of the raw 3D representation, *i.e.*, point cloud, and thus is generalizable to arbitrary 3D scenarios.

Tail Cuboid Over-Sampling. Besides embedding target context to source data, our cuboid mixing technique also allows adjusting the category distributions by designing cuboid sampling strategies. Here, as an add-on advantage, we leverage this nice property to alleviate the biased pseudo label problem [2,19,36,73] in self-training: tail categories only occupy a small percentage of pseudo labeled data. Specifically, we sample cuboids with tail categories more frequently, namely tail cuboid over-sampling, detailed as follows.

We calculate per-class pseudo label ratio $r \in [0, 1]^c$ and define n_r least common categories as tail categories. We then define tail cuboid whose pseudo label ratio is higher than the average value r on at least one of n_r tail categories. We construct a tail cuboid queue Q with size N_q to store tail cuboids. Formally, $\gamma^t_{q[w]}$ denotes the w^{th} tail cuboid in Q, as shown in Fig. 4. Notice that through training, Q is dynamically updated with First In, First Out (FIFO) rule since cuboids are randomly split in each iteration as Eq. (4). In each training iteration, we ensure that at least u tail cuboids are in each mixed scene by sampling cuboids from Q and replacing existing cuboids if needed. With such a simple over-sampling strategy, we make the cuboid mixing process tail-aware,

and relieve the class imbalance issue in the self-training. Experimental results in Sect. 6 further demonstrate the effectiveness of our tail cuboid over-sampling strategy.

Self-training with Target and Source data. In the self-training stage, for data augmentation, VSS is first adopted to augment the source domain data to reduce the pattern gap and then TACM mixes source and target scenes to construct a tail-aware intermediate domain $\mathcal{D}_m = \{P^m\}$ with labels $\hat{Y}^m$ mixed by source ground-truth and target pseudo labels. To alleviate the noisy supervisions from incorrect target pseudo labels, we minimize dense cross-entropy loss on source data $\tilde{P}^s$ and intermediate domain data P^m as below:

$$\min \mathcal{L}_{st} = \sum_{i=1}^{N_t} \mathrm{CE}(S_i^m, \hat{Y}_i^m) + \lambda \sum_{i=1}^{N_s} \mathrm{CE}(S_i^s, Y_i^s), \tag{6}$$

where λ denotes the trade-off factor between losses.

4 Benchmark Setup

4.1 Datasets

3D-FRONT [8] is a large-scale dataset of synthetic 3D indoor scenes, which contains 18,968 rooms with 13,151 CAD 3D furniture objects from 3D-FUTURE [9]. The layouts of rooms are created by professional designers and distinctively span 31 scene categories and 34 object semantic super-classes. We randomly select 4995 rooms as training samples and 500 rooms as validation samples after filtering out noisy rooms. Notice that we obtain source point clouds by uniformly sampling points from original mesh with CloudCompare [12] at 1250 surface density (number of points per square units). Comparison between 3D-FRONT and other simulation datasets are detailed in the nsupplemental materials.

ScanNet [7] is a popular real-world indoor 3D scene understanding dataset, consisting 1,613 real 3D scans with dense semantic annotations (*i.e.*,, 1,201 scans for training, 3,12 scans for validation and 100 scans for testing). It provides semantic annotations for 20 categories.

S3DIS [3] is also a well-known real-world indoor 3D point cloud dataset for semantic segmentation. It contains 271 scenes across six areas along with 13 categories with point-wise annotations. Similar to previous works [33,46], we use the fifth area as the validation split and other areas as the training split.

Label Mapping. Due to different category taxonomy of datasets, we condense 11 categories for 3D-FRONT → ScanNet and 3D-FRONT → S3DIS settings, individually. Besides, we condense 8 categories for cross-site settings between S3DIS and ScanNet. Please refer to the Suppl. for the detailed taxonomy.

4.2 UDA Baselines

As shown in Table 1 and 2, we reproduce 7 popular 2D UDA methods and 1 3D outdoor method as UDA baselines, encompassing MCD [51], AdaptSegNet [56], CBST [73], MinEnt [57], AdvEnt [57], Noisy Student [62], APO-DA [66] and SqueezeSegV2 [60]. These UDA baselines cover most existing streams such as adversarial alignment, discrepancy minimization, self-training and entropy guided adaptation. To perform these image-based methods on our setting, we carry out some task-specific modifications, which are detailed in supplemental materials.

5 Experiments

To validate our method, we benchmark DODA and other popular UDA methods with extensive experiments on 3D-FRONT [8], ScanNet [7] and S3DIS [3]. Moreover, we explore a more challenging setting, from simulated 3D-FRONT [8] to RGBD realistic dataset NYU-V2 [42], presented in the supplementary materials. To verify the generalizability of VSS and TACM, we further integrate VSS to previous UDA methods and adopt TACM in the real-world cross-site UDA setting. Note that since textures for some background classes are not provided in 3D-FRONT dataset, we only focus on adaptation using 3D point positions. The implementation details including network and training details are provided in the Suppl.

Comparison to Other UDA Methods. As shown in Table 1 and Table 2, compared to source only, DODA largely lifts the adaptation performance in terms of mIoU by around 21% and 19% on 3D-FRONT → ScanNet and 3D-FRONT → S3DIS, respectively. DODA also shows its superiority over other popular UDA methods, obtaining 14% ∼ 22% performance gain on 3D-FRONT → ScanNet and 13% ∼ 19% gain on 3D-FRONT → S3DIS. Even only equipping source only with VSS module, our DODA (only VSS) still outperforms UDA baselines by around 4% ∼ 10%, indicating that the pattern gap caused by different sensing mechanisms significantly harms adaptation results while previous methods have not readily addressed it. Comparing DODA with DODA (w/o TACM), we observe that TACM mainly contributes to the performance of instances such as bed and bookshelf on ScanNet, since cuboid mixing forces model to focus more on local semantic clues and object shapes itself inside cuboids. It is noteworthy that though DODA yields general improvement around almost all categories adaptation in both pretrain stage and self-training stage, challenging classes such as bed on ScanNet and sofa on S3DIS attain more conspicuous performance lift, demonstrating the predominance of DODA in tackling troublesome categories. However, the effectiveness of all UDA methods for column and beam on S3DIS are not obvious due to their large disparities in data patterns across domains and low appearing frequencies in source domain. To illustrate the reproducibility of our DODA, all results are repeated three times and reported as average performance along with standard variance.

Table 1. Adaptation results of 3D-FRONT → ScanNet in terms of mIoU. We indicate the best adaptation result in **bold**. † denotes pretrain generalization results with VSS

Method	mIoU	Wall	Floor	Cab	Bed	Chair	Sofa	Table	Door	Wind	Bksf	Desk
Source only	29.60	60.72	82.42	04.44	12.02	61.76	22.31	38.52	05.72	05.12	19.72	12.84
MCD [51]	32.27	62.86	88.70	03.81	38.50	57.51	21.48	41.67	05.78	01.29	18.81	15.69
AdaptSegNet [56]	34.51	61.81	83.90	03.64	36.06	55.05	34.26	44.21	06.59	05.54	31.87	16.64
CBST [73]	37.42	60.37	81.39	12.18	30.00	68.86	36.22	49.93	07.05	05.82	43.59	16.25
MinEnt [57]	34.61	63.35	85.54	04.66	26.05	61.98	33.05	48.38	05.20	03.15	35.84	13.49
AdvEnt [57]	32.81	64.31	79.21	04.39	35.01	61.05	24.36	41.64	05.97	01.60	29.07	14.32
Noisy student [62]	34.67	62.63	86.27	01.45	17.13	69.98	37.58	47.87	06.01	01.66	35.79	15.06
APO-DA [66]	31.73	62.84	85.43	02.77	15.08	64.24	34.41	46.41	03.94	03.59	18.88	11.41
SqueezeSegV2 [60]	29.77	61.85	72.74	02.50	16.89	58.79	16.81	38.19	05.08	03.24	35.68	15.72
DODA (only VSS)†	40.52±0.80	67.36	90.24	15.98	39.98	63.11	46.38	48.05	07.63	13.98	33.17	19.86
DODA (w/o TACM)	48.13±0.25	72.22	93.43	24.46	56.30	70.40	53.33	56.57	**09.44**	19.97	47.05	26.25
DODA	**51.42±0.90**	**72.71**	**93.86**	**27.61**	**64.31**	**71.64**	**55.30**	**58.43**	08.21	**24.95**	**56.49**	**32.06**
Oracle	75.19	83.39	95.11	69.62	81.15	88.95	85.11	71.63	47.67	62.74	82.63	59.05

Table 2. Adaptation results of 3D-FRONT → S3DIS in terms of mIoU. We indicate the best adaptation result in **bold**. † denotes pretrain generalization results with VSS

Method	mIoU	Wall	Floor	Chair	Sofa	Table	Door	Wind	Bkcase	Ceil	Beam	Col
Source only	36.72	67.95	88.68	57.69	04.15	38.96	06.99	00.14	44.90	94.42	00.00	00.00
MCD [51]	36.62	64.53	92.16	54.76	13.31	46.67	8.54	00.08	28.86	93.89	00.00	00.00
AdaptSegNet [56]	38.14	68.14	93.17	55.14	05.31	43.14	14.67	00.33	45.75	93.88	00.00	00.00
CBST [73]	42.47	71.60	92.07	68.09	03.28	60.45	17.13	00.18	58.45	95.87	00.00	00.00
MinEnt [57]	37.08	66.15	87.92	52.30	06.27	25.79	15.70	04.44	55.72	93.58	00.00	00.00
AdvEnt [57]	37.98	66.94	91.84	57.96	02.39	46.18	15.14	00.54	44.31	92.50	00.00	00.00
Noisy student [62]	39.44	68.84	91.78	65.53	06.65	48.67	02.27	00.00	53.67	**96.46**	00.00	00.00
APO-DA [66]	38.23	68.63	89.66	58.84	03.51	40.66	13.73	02.61	47.88	94.97	**00.04**	00.00
SqueezeSegV2 [60]	36.50	65.01	89.95	54.29	06.79	45.75	10.23	01.70	32.93	94.81	00.00	00.00
DODA (only VSS) †	46.85±0.78	70.96	96.12	68.70	25.47	58.47	17.87	27.65	54.39	95.66	00.00	00.00
DODA (w/o TACM)	53.86±0.49	75.75	95.14	76.12	60.11	64.07	25.24	31.75	**68.49**	95.82	00.00	00.00
DODA	**55.54±0.91**	**76.23**	**97.17**	**76.89**	**63.55**	**69.04**	**25.76**	**38.22**	68.18	95.85	00.00	00.00
Oracle	62.29	82.82	96.95	78.16	40.37	78.56	56.91	47.90	77.10	96.29	00.41	29.69

VSS Plug-and-Play Results to Other UDA Methods. Since VSS works as a data augmentation in our DODA, we argue that it can serve as a plug-and-play module to mimic occlusion and noise patterns on simulation data, and is orthogonal to existing UDA strategies. As demonstrated in Table 3, equipped with VSS, current popular UDA approaches consistently surpass their original performance by around 8% $\sim$ 13%. It also verifies that previous 2D-based methods fail to close the point pattern gap in 3D indoor scene adaptations, while our VSS can be incorporated into various pipelines to boost performance.

TACM Results in Cross-Site Adaptation. Serving as a general module to alleviate domain shifts across domains, we show that TACM can consistently mitigate domain discrepancies on even real-to-real adaptation settings. For cross-site adaptation, scenes collected from different sites or room types also suffer a considerable data distribution gap. As shown in Table 4, the domain gaps in real-to-real adaptation tasks are also large when comparing the source only and

Table 3. UDA results equipped with VSS on 3D-FRONT → ScanNet

Method	VSS		Improv.
	w/o	w/	
MCD [51]	32.37	40.32	+7.95
AdaptSegNet [56]	34.51	45.75	+11.24
CBST [73]	36.30	47.70	+11.40
MinEnt [57]	34.61	43.26	+8.65
AdvEnt [57]	32.81	42.94	+10.13
Noisy Student [62]	34.67	48.30	+13.63
APO-DA [66]	31.73	43.98	+12.25
SqueezeSegV2 [60]	29.77	40.60	+10.83

Table 4. Cross-site adaptation results with TACM

Task	Method	mIoU
ScanNet→ S3DIS	Source only	54.09
	CBST [73]	60.13
	CBST+TACM	65.52
	Oracle	72.51
S3DIS → ScanNet	Source only	33.48
	Noisy student [62]	44.81
	Noisy student+TACM	48.47
	Oracle	80.06

oracle results. When adopting TACM in the self-training pipelines, they obtain 5.64% and 3.66% relative performance boost separately in ScanNet → S3DIS and S3DIS → ScanNet. These results verify that TACM is general in relieving data gaps, especially the context gap on various 3D scene UDA tasks. We provide the cross-site benchmark with more UDA methods in the Suppl.

6 Ablation Study

In this section, we conduct extensive ablation experiments to investigate the individual components of our DODA. All experiments are conducted on 3D-FRONT → ScanNet for simplicity. Default settings are marked in **bold**.

Component Analysis. Here, we investigate the effectiveness of each component and module in our DODA. As shown in Table 5, occlusion simulation brings the largest performance gain (around 9.7%), indicating that model trained on complete scenes is hard to adapt to scenes with occluded patterns. Noise simulation further supplements VSS to imitate sensing and reconstruction noise, obtaining about 1.3% boosts. Two sub-modules jointly mimic realistic scenes, largely alleviating the point distribution gap and leading to a more generalizable source only model. In the self-training stage, VSS also surpasses the baseline by around 13% due to its efficacy in reducing the point pattern gap and facilitating generating high-quality pseudo labels. Cuboid mixing combines cuboid patterns from source and target domains for moderating context-level bias, further boosting the performance by around 2.4%. Moreover, cuboid-level tail-class over-sampling yields 0.9% improvement with greater gains on tail classes. For instance, desk on ScanNet achieves 6% gain (see Suppl.).

VSS: Visible Range. Here, we study the effect of visible range of VSS, which is jointly determined by the horizontal angle α_h, vertical angle α_v, viewing mode η and the number of cameras n_v. As shown in Table 6, fewer cameras $n_v = 2$ and smaller viewing angle $\alpha_v = 45°$ draw around 2% performance degradation with

Table 5. Component Analysis for DODA on 3D-FRONT → ScanNet

Baseline	Virtual scan simulation		Tail-aware cuboid mixing		mIoU
	Occlusion sim	Noise sim	Cuboid mix	Tail samp	
Source only					29.60
Source only	✓				39.25 (+9.65)
Source only	✓	✓			40.52 (+1.27)
Noisy student					34.67
Noisy student	✓	✓			48.13 (+13.46)
Noisy student	✓	✓	✓		50.55 (+2.42)
Noisy student	✓	✓	✓	✓	**51.42** (+0.87)

Table 6. Ablation study of visible range design on 3D-FRONT → ScanNet

α_h	α_v	η	n_v	mIoU
180°	90°	Fixed	2	38.80
180°	90°	Fixed	4	**40.52**
90°	90°	Fixed	8	40.30
180°	90°	Parallel	4	39.08
180°	90°	Perspective	4	39.04
180°	45°	Perspective	4	36.64

Table 7. Ablation study of cuboid partitions on 3D-FRONT → ScanNet

(n_x, n_y, n_z)	# cuboid	mIoU
(1, 1, 1)	1	48.10
(2, 1, 1)	2	50.00
(2, 2, 1)	4	**50.55**
(3, 2, 1)	6	50.57
(3, 3, 1)	9	50.02
(1, 1, 2)	2	49.49
(2, 1, 2)	4	49.48

a smaller visible range. And decreasing α_h to 90° can also achieve similar performance with $\alpha_h = 180°$ with more cameras $n_v = 8$, demonstrating that enough semantic coverage is a vital factor. Besides, as for the three viewing modes η, the simplest fixed mode achieves the highest performance in comparison to parallel and perspective modes. Even though parallel and perspective are more similar to reality practice, they cannot cover sufficient range with limited cameras, since real-world scenes are reconstructed through hundreds or thousands of view frames. This again demonstrates that large spatial coverage is essential. To trade off between the effectiveness and efficiency of on-the-fly VSS, we use fixed mode with 4 camera positions by default here.

TACM: Cuboid Partition. We study various cuboid partition manners in Table 7. Notice that random rotation along z axis is performed before cuboid partition, so the partition on x or y axes can be treated as identical. While horizontal partitioning yields consistent performance beyond 50% mIoU, vertical partitioning does not show robust improvements, suggesting the mixing of vertical spatial context is not necessary. Simultaneous partitioning on x and y axes also improves performance (*i.e.*, (2,2,1) and (2,3,1)), while too small cuboid

Table 8. Ablation study of data-mixing methods on 3D-FRONT → ScanNet

Method	mIoU
Mix3D [43]	48.62
CutMix [68]	49.19
Copy-Paste [11]	48.51
TACM	**51.42**

Table 9. Investigation of pseudo label class imbalance issue on 3D-FRONT → ScanNet

Method	mIoU
Noisy student	48.13
TACM	**51.42**
CM + lovasz loss [4]	51.68
TACM + lovasz loss [4]	52.50

size (*i.e.*, (3,3,1)) results in insufficient context cues in each cuboid with a slight decrease in mIoU.

TACM: Data-Mixing Method. We compare TACM with other popular data-mixing methods in Table 8. Experimental results show the superiority of TACM since it outperforms Mix3D [43], CutMix [68] and Copy-Paste [11] by around 2.2% to 2.9%. TACM effectively alleviates the context gap while preserving local context clues. Mix3D, however, results in large overlapping areas, which is unnatural and causes semantic confusions. CutMix and Copy-Paste only disrupt local areas without enough perturbations of the broader context (see Suppl.).

TACM: Tail Cuboid Over-Sampling with Class-Balanced Loss. Tail cuboid over-sampling brings significant gains on tail classes as discussed in Sect. 6. As demonstrated in Table 9, the class-balanced lovasz loss [4] also boosts performance by considering each category more equally. We highlight that our TACM can also incorporate with other class-balancing methods during optimization since it eases long tail issue on the data-level.

7 Limitations and Open Problems

Although our model largely closes the domain gaps across simulation and real-world datasets, we still suffer from the inherent limitations of the simulation data. For some categories such as beam and column, the simulator fails to generate realistic shape patterns, resulting in huge negative transfer. Besides, room layouts need to be developed by experts, which may limit the diversity and complexity of the created scenes. Therefore, in order to make simulation data benefit real-world applications, there are still several open problems: how to handle the failure modes of the simulator, how to unify the adaptation and simulation stage in one pipeline, and how to automate the simulation process, to name a few.

8 Conclusions

We have presented DODA, a data-oriented domain adaptation method with virtual scan simulation and tail-aware cuboid mixing for 3D indoor sim-to-real unsupervised domain adaptation. Virtual scan simulation generates a more

transferable model by mitigating the real-and-simulation point pattern gap. Tail-aware cuboid mixing rectifies context biases through creating a tail-aware intermediate domain and facilitating self-training to effectively leverage pseudo labeled target data, further reducing domain gaps. Our extensive experiments not only show the prominent performance of our DODA in two sim-to-real UDA tasks, but also illustrate the potential ability of TACM to solve general 3D UDA scene parsing tasks. More importantly, we have built the first benchmark for 3D indoor scene unsupervised domain adaptation, including sim to real adaptation and cross-site real-world adaptation. The benchmark suit will be publicly available. We hope our work could inspire further investigations on this problem.

Acknowledgement. This work has been supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), HKU Startup Fund, and HKU Seed Fund for Basic Research.

References

1. Achituve, I., Maron, H., Chechik, G.: Self-supervised learning for domain adaptation on point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 123–133 (2021)
2. Araslanov, N., Roth, S.: Self-supervised augmentation consistency for adapting semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15384–15394 (2021)
3. Armeni, I., et al.: 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1534–1543 (2016)
4. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4413–4421 (2018)
5. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics **22**(14), e49–e57 (2006)
6. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3075–3084 (2019)
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5828–5839 (2017)
8. Fu, H., et al.: 3d-front: 3D furnished rooms with layouts and semantics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10933–10942 (2021)
9. Fu, H., et al.: 3D-future: 3D furniture shape with texture. Int. J. Comput. Vision **129**, 1–25 (2021)
10. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189 (2015)
11. Ghiasi, G., et al.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2918–2928 (2021)

12. Girardeau-Montaut, D.: Cloudcompare. EDF R&D Telecom ParisTech, France (2016)
13. Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: domain flow for adaptation and generalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2477–2486 (2019)
14. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
16. Graham, B., Engelcke, M., Van Der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9224–9232 (2018)
17. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
18. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: Understanding real world indoor scenes with synthetic data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4077–4085 (2016)
19. He, R., Yang, J., Qi, X.: Re-distributing biased pseudo labels for semi-supervised semantic segmentation: a baseline investigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6930–6940 (2021)
20. Hoffman, J., et al.: Cycada: cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
21. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
22. Hu, Q., et al.: Randla-net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108–11117 (2020)
23. Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., Pérez, P.: xmuda: cross-modal unsupervised domain adaptation for 3D semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12605–12614 (2020)
24. Jiang, L., et al.: Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6423–6432 (2021)
25. Jiang, L., Zhao, H., Liu, S., Shen, X., Fu, C.W., Jia, J.: Hierarchical point-edge interaction network for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10433–10441 (2019)
26. Kar, A., et la.: Meta-sim: learning to generate synthetic datasets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4551–4560 (2019)
27. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. In: ACM SIGGRAPH 2007 papers, pp. 24-es. x (2007)
28. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 480–490 (2019)
29. Kong, L., Quader, N., Liong, V.E.: Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. arXiv preprint arXiv:2111.15242 (2021)
30. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D instance segmentation via multi-task metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9256–9266 (2019)

31. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4558–4567 (2018)
32. Lee, D.H., et al.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, no. 2, p. 896 (2013)
33. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: convolution on x-transformed points. Adv. Neural Inf. Process. Syst. **31**, 820–830 (2018)
34. Li, Z., et al.: Openrooms: an end-to-end open framework for photorealistic indoor scene datasets. arXiv preprint arXiv:2007.12868 (2020)
35. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: a general approach to adapting deep classifiers. In: International Conference on Machine Learning, pp. 4013–4022 (2019)
36. Liu, Y.C., et al.: Unbiased teacher for semi-supervised object detection. arXiv preprint arXiv:2102.09480 (2021)
37. Liu, Z., Qi, X., Fu, C.W.: One thing one click: a self-training approach for weakly supervised 3D semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1726–1736 (2021)
38. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 (2015)
39. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: International Conference on Machine Learning, pp. 2208–2217 (2017)
40. Luo, Z., et al.: Unsupervised domain adaptive 3D detection with multi-level consistency. arXiv preprint arXiv:2107.11355 (2021)
41. Maturana, D., Scherer, S.: Voxnet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928. IEEE (2015)
42. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
43. Nekrasov, A., Schult, J., Litany, O., Leibe, B., Engelmann, F.: Mix3d: out-of-context data augmentation for 3D scenes. In: 2021 International Conference on 3D Vision (3DV), pp. 116–125. IEEE (2021)
44. Peng, D., Lei, Y., Li, W., Zhang, P., Guo, Y.: Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3D semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7108–7117 (2021)
45. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
46. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
47. Qin, C., You, H., Wang, L., Kuo, C.C.J., Fu, Y.: Pointdan: a multi-scale 3D domain adaption network for point cloud representation. arXiv preprint arXiv:1911.02744 (2019)
48. Ramamonjison, R., Banitalebi-Dehkordi, A., Kang, X., Bai, X., Zhang, Y.: Simrod: a simple adaptation method for robust object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3570–3579 (2021)

49. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: International Conference on Machine Learning, pp. 2988–2997. JMLR. org (2017)
50. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6956–6965 (2019)
51. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
52. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3693–3702 (2017)
53. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1746–1754 (2017)
54. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. Adv. Neural Inf. Process. Syst. **30**, 1195–1204 (2017)
55. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6411–6420 (2019)
56. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7472–7481 (2018)
57. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2517–2526 (2019)
58. Wang, Y., et al.: Train in Germany, test in the USA: making 3D object detectors generalize. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11713–11723 (2020)
59. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Trans. Graph. (tog) **38**(5), 1–12 (2019)
60. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: Squeezesegv 2: improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 4376–4382. IEEE (2019)
61. Wu, W., Qi, Z., Fuxin, L.: Pointconv: deep convolutional networks on 3D point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9621–9630 (2019)
62. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10687–10698 (2020)
63. Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3173–3182 (2021)
64. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d++: denoised self-training for unsupervised domain adaptation on 3D object detection. arXiv preprint arXiv:2108.06682 (2021)

65. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d: self-training for unsupervised domain adaptation on 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
66. Yang, J., et al.: An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12613–12620 (2020)
67. Yi, L., Gong, B., Funkhouser, T.: Complete & label: a domain adaptation approach to semantic segmentation of lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15363–15373 (2021)
68. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023–6032 (2019)
69. Zhang, W., Li, W., Xu, D.: Srdan: scale-aware and range-aware domain adaptation network for cross-dataset 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6769–6779 (2021)
70. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: Pointweb: enhancing local neighborhood features for point cloud processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5565–5573 (2019)
71. Zhao, S., et al.: epointda: an end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation, vol. 2, p. 3. arXiv preprint arXiv:2009.03456 (2020)
72. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3D: a large photo-realistic dataset for structured 3D modeling. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12354, pp. 519–535. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58545-7_30
73. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: European Conference on Computer Vision, pp. 289–305 (2018)