3D-to-2D Distillation for Indoor Scene Parsing

Zhengzhe Liu[1] Xiaojuan Qi[2]* Chi-Wing Fu[1]*
[1]The Chinese University of Hong Kong [2]The University of Hong Kong
{zzliu,cwfu}@cse.cuhk.edu.hk xjqi@eee.hku.edu.hk

Abstract

*Indoor scene semantic parsing from RGB images is very challenging due to occlusions, object distortion, and viewpoint variations. Going beyond prior works that leverage geometry information, typically paired depth maps, we present a new approach, a 3D-to-2D distillation framework, that enables us to leverage 3D features extracted from large-scale 3D data repositories (*e.g.*, ScanNet-v2) to enhance 2D features extracted from RGB images. Our work has three novel contributions. First, we distill 3D knowledge from a pretrained 3D network to supervise a 2D network to learn simulated 3D features from 2D features during the training, so the 2D network can infer without requiring 3D data. Second, we design a two-stage dimension normalization scheme to calibrate the 2D and 3D features for better integration. Third, we design a semantic-aware adversarial training model to extend our framework for training with unpaired 3D data. Extensive experiments on various datasets, ScanNet-V2, S3DIS, and NYU-v2, demonstrate the superiority of our approach. Also, experimental results show that our 3D-to-2D distillation improves the model generalization.*

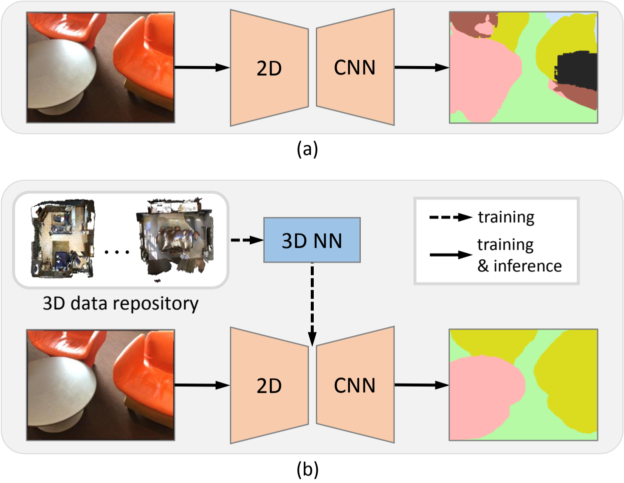

Figure 1. Compared with extracting features solely from the input image (a) for semantic parsing, our new approach (b) efficiently distills 3D features learned from a large-scale 3D data repository to train the 2D CNN to learn to enhance its features for better semantic parsing. Our framework needs point cloud inputs only in training but not in testing, and the point cloud can be paired or unpaired.

1. Introduction

Indoor scene parsing from images plays an important role in many applications such as robot navigation and augmented reality. Though a considerable amount of advancements have been obtained with convolutional neural networks, this task is still very challenging, since the task inherently suffers from various issues, including distorted object shapes, severe occlusions, viewpoint variations, and scale ambiguities.

One approach to address the issues is to leverage auxiliary geometric information to obtain structured information that complements the RGB input. For the auxiliary input, existing methods typically employ the depth map that associates with the input RGB image. However, earlier methods [15, 11, 6, 37, 41, 47, 10] require the availability of the depth map inputs not only in the training but also in the testing. As a result, they have limited applicability to general situations, in which depth is not available. This is in contrast to the ubiquity of 2D images, which can be readily obtained by the many photo-taking devices around us.

To get rid of the constraint, several methods [50, 58, 54, 59, 22] propose to predict a depth map from the RGB input, then leverage the predicted depth to boost the scene parsing performance. However, depth prediction from a single image is already a very challenging task on its own. Hence, the performance of these methods largely depends on the quality of the predicted depth. Also, the additional depth prediction raises the overall complexity of the network.

Besides the above issues, a common limitation of the prior works is that they only explore the depth map as the auxiliary geometry cue. Yet, a depth map can only give a partial view of a 3D scene, so issues like occlusions and viewpoint variations are severe. Further, they all require paired RGB-depth data in training. So, they are limited for use on datasets with

*: Corresponding authors

depth maps, which require tedious manual preparation, *e.g*., hardware setups, complicated calibrations, etc.

In this work, we present the first flexible and lightweight framework (see Figure 1), namely *3D-to-2D distillation*, to distill occlusion-free, viewpoint-invariant 3D representations derived from 3D point clouds for embedding into 2D CNN features by training the network to learn to simulate 3D features from the input image. Our approach leverages existing large-scale 3D data repositories such as ScanNet-v2 [8] and S3DIS [1] and recent advancements in 3D scene understanding [14, 7, 17, 21] for 3D feature extraction, and allows the use of unpaired 3D data to train the network.

For the 2D CNN to effectively learn to simulate 3D features, our 3D-to-2D distillation framework incorporates a two-stage *dimension normalization* (DN) module to explicitly align the statistical distributions of the 2D and 3D features. So, we can effectively reduce the numerical distribution gap between the 2D and 3D features, as they are from different data modalities and neural network models. Also, a *Semantic Aware Adversarial Loss* (SAAL) is designed to serve as the objective of model optimization without paired 2D-3D data to make the framework flexible to leverage existing 3D data repository and boost its applicability.

We conduct extensive experiments on indoor scene parsing datasets ScanNet-v2 [8], S3DIS [1], and NYU-v2 [44]. With only a negligible amount of extra computation cost, our approach consistently outperforms the baselines including the state-of-the-art depth-assisted semantic parsing approach [22] and our two baselines that leverage depth maps, manifesting the superiority of our approach. Besides, our further in-depth experiments on a depth reconstruction task implies that our framework can effectively embed 3D representations into 2D features and produce much better reconstruction results. More importantly, our model obtains a significant performance gain (19.08% *vs*. 27.22% mIoU), even when evaluated on data from an unseen domain, suggesting that the 3D information embedded by our 3D-to-2D distillation helps promote the generalizability of CNNs.

2. Related Work

Semantic segmentation. The computer vision community has gained remarkable achievements on semantic segmentation [29, 55, 33, 38, 2, 4, 61, 62, 51, 63, 56, 57, 49]. PSPNet [62] is a representative work that has inspired many follow-ups. In this work, we adopt PSPNet as the baseline model for semantic segmentation, as it has an open-source repository with good reproducibility and delivers competitive performance even compared with the latest works.

3D semantic segmentation. Methods for 3D semantic segmentation are generally point-based or voxel-based. *Point-based networks* adopt raw point clouds as input. Along this line of works, [35, 36] are pioneering ones. Later, various convolution-based methods [25, 46, 53, 3] were proposed for 3D semantic segmentation on point clouds. Recently, Kundu *et al*. [23] proposed to fuse features from multiple 2D views for 3D semantic segmentation.

On the other hand, *Voxel-based networks* first voxelize the raw data into regular 3D grids for feature learning [39, 14, 45, 9, 64]. The recently-proposed methods MinkowskiNet [7] and OccuSeg [17] are two of the representative works in this branch. In this work, we adopt PointWeb [60] and MinkowskiNet [7] as the architectures for extracting point-based and voxel-based 3D features, respectively.

Knowledge distillation. Our work shares a similar spirit as knowledge distillation techniques [19, 42, 16, 30, 12, 18, 28] in that we both aim to transfer features from a source model (i.e., the teacher model in knowledge distillation or the 3D network in our work) to a target model (*i.e*., the student model in knowledge distillation or the 2D network in our work) to enhance the performance of the target model. However, conventional knowledge distillation techniques are designed typically for scenarios, in which (i) the source data to be distilled has the same modality [19, 42, 16, 30] or similar modalities [12] as the target, (ii) the two networks for the feature extraction have the same or similar architecture (*e.g*., convolution neural networks), and (iii) the distillation objective typically requires paired source-target data.

Recently, cross-modality distillation has also been studied in [32, 13, 40]. However, their methods are not suitable in our, since they cannot distinguish features of different categories. To this end, we propose to associate 2D and 3D features by the object category and formulate the SAAL to enable unpaired training. The experimental result shows that our approach outperforms the most recent semantic segmentation knowledge distillation method [28].

3. Method

This section presents our proposed 3D-to-2D distillation framework for effective distillation of 3D features learned from point clouds to improve the performance of 2D indoor scene parsing. Figure 2 shows the overall architecture of the framework. During the training, our framework takes a 2D image and a 3D point cloud as its inputs. To begin, we use a 3D CNN to extract 3D feature f_{3D} from the input 3D point cloud (blue region in Figure 2). On the other hand, we use a 2D CNN to extract 2D feature f_{2D} from the input image, and produce also simulated 3D feature f_{3D_sim} in the 2D network (orange region in Figure 2). For the 2D CNN, we adopt PSPNet [62] for the case of semantic segmentation, whereas for the 3D CNN, we may adopt different 3D architectures, such as PointWeb [60] and MinkowskiNet [7]. Also, note that DN_1 and DN_2 are our dimension normalization modules.

An important insight in our approach is that during the training, we use f_{3D} to supervise the generation of f_{3D_sim}, so

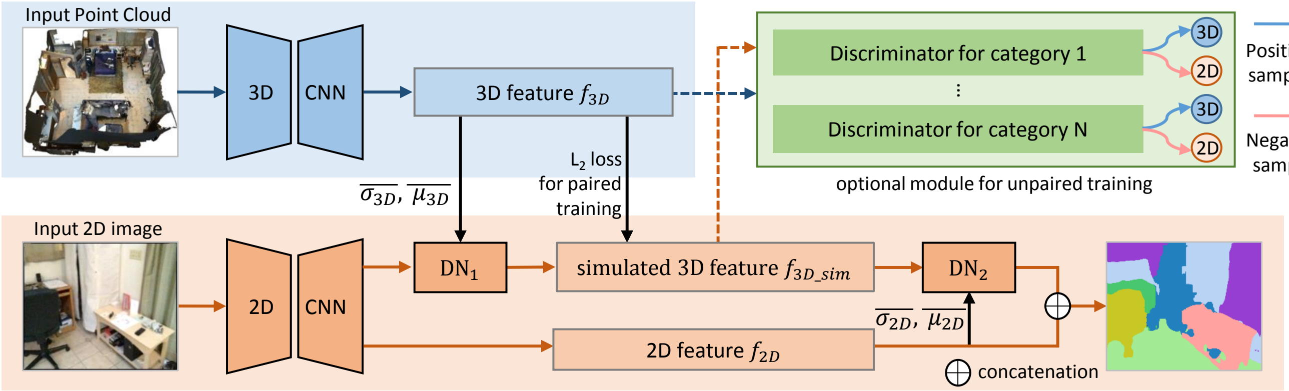
Input Point Cloud
3D
CNN
3D feature f_{3D}
$\overline{\sigma_{3D}}$, $\overline{\mu_{3D}}$
L_2 loss for paired training
Discriminator for category 1
⋮
Discriminator for category N
3D
2D
3D
2D
Positive sample
Negative sample
optional module for unpaired training
Input 2D image
2D
CNN
DN$_1$
simulated 3D feature f_{3D_sim}
DN$_2$
$\overline{\sigma_{2D}}$, $\overline{\mu_{2D}}$
2D feature f_{2D}
⊕ concatenation

Figure 2. Overview of our 3D-to-2D distillation framework for 2D semantic segmentation. During the training, the framework takes a 2D image and a 3D point cloud as inputs. We transfer 3D feature f_{3D} from the 3D network (blue) to the 2D network (orange) with an L_2 loss (for paired 2D-3D data), such that the 2D network can learn to produce simulated 3D feature $f_{\text{3D\_sim}}$ from the 2D image and we do not need f_{3D} and 3D point cloud input during the inference. Also, note the two dimension normalization modules (DN$_1$ and DN$_2$) for aligning 2D-3D features, and the optional module on top-right for training with unpaired 2D-3D data using the semantic-aware adversarial loss.

that the 2D network can learn to produce $f_{\text{3D\_sim}}$ that looks like f_{3D}. In this way, we do not need the point cloud input and 3D network when we use our framework to test on a 2D image; the 2D network alone can generate $f_{\text{3D\_sim}}$ solely from the image input. Also, to resolve the statistical difference between the 2D and 3D inputs, we design a two-stage dimension normalization module (DN$_1$ and DN$_2$) to effectively transform features before and after $f_{\text{3D\_sim}}$ (Figure 2), so that the statistical distributions of the 2D and 3D features in the 2D network are better aligned to facilitate the learning of the simulated 3D feature and also its integration with the remaining part of the 2D network. Further, we design a semantic-aware adversarial loss as an optional module in the training (Figure 2 (top right)) to extend our framework for training with 2D-3D inputs that are unpaired.

In this section, we first present how our framework is trained with paired 2D-3D inputs (Section 3.1). Then, we present our dimension normalization modules in Section 3.2 and how we extend the framework for unpaired training with the semantic aware adversarial loss in Section 3.3.

3.1. Training Objectives with Paired 2D-3D Data

Since 3D feature f_{3D} extracted from the input point cloud is point-wise in the 3D space, we cannot directly associate it with 2D feature f_{2D} and likewise the simulated 3D feature $f_{\text{3D\_sim}}$, which are both defined in 2D image space. To determine the associations, if paired 2D-3D data is available in the training, we can use the given camera parameters to transform and project the 3D points to the 2D image space. In this way, we can determine the pixel location associated with each point in the input point cloud, and obtain $f_{\text{3D\_}i}$, which denotes the projected 3D feature at pixel i. If more than one point projects to the same pixel, we consider only the point that is the nearest to the camera, since it should correspond to the visible pixel in the image input.

Then, we adopt an L_2 regression loss between f_{3D} and $f_{\text{3D\_sim}}$ to supervise the generation of $f_{\text{3D\_sim}}$ in the 2D network. Clearly, the projected 3D points are sparse in the image space, so we locate the pixels covered by the points and perform the regression only on the covered pixels:

$$L_p = \sum_i ||f_{\text{3D\_}i} - f_{\text{3D\_sim\_}i}||_2^2, \qquad (1)$$

where i indexes the pixels covered by the projected 3D points and L_p denotes the loss to guide the generation of $f_{\text{3D\_sim}}$.

To train the whole framework for the semantic segmentation task, we employ the cross-entropy loss below:

$$L_s = \sum_i \sum_c -\mathbb{1}_{i,c} \log p_{i,c}, \qquad (2)$$

where $p_{i,c}$ denotes the probability of pixel i belonging to category c and $\mathbb{1}_{i,c}$ is an indicator function that equals 1 if the ground-truth category of pixel i is c; otherwise, it is zero.

3.2. Dimension Normalization

To resolve the statistical difference between 2D and 3D representations induced by different data modalities and neural network architectures, we design the dimension normalization modules DN$_1$ and DN$_2$ to explicitly calibrate the distribution of the 2D and 3D features; see Figure 2.

Figure 3 illustrates the technical details. Give a batch of N feature maps as inputs to DN$_1$ or DN$_2$, each of the dimensions H (height), W (width), and C (channels), we compute channel-wise means and variances of the feature map over the N, H, and W dimensions: $(\mu_{\text{2D}}, \sigma_{\text{2D}})$ for the input 2D feature map to DN$_1$ or $(\mu_{\text{3D}}, \sigma_{\text{3D}})$ for the input 3D feature map to DN$_2$. On the other hand, as shown in Figure 2, DN$_1$ receives statistics ($\bar{\mu}_{\text{3D}}$, $\bar{\sigma}_{\text{3D}}$) of the 3D feature f_{3D}, whereas DN$_2$ receives statistics ($\bar{\mu}_{\text{2D}}$, $\bar{\sigma}_{\text{2D}}$) of the 2D feature f_{2D}. Below, we detail the procedure inside each DN:

BN	AdaBN	DN
$\hat{x} = \gamma\frac{x-\mu_{\text{2D}}}{\sigma_{\text{2D}}} + \beta$	$\hat{x} = \gamma\frac{x-\bar{\mu}_{\text{3D}}}{\bar{\sigma}_{\text{3D}}} + \beta$	$\hat{x}_{\text{3D}} = \Delta\sigma_{\text{3D}}(\bar{\sigma}_{\text{3D}}\frac{x-\mu_{\text{2D}}}{\sigma_{\text{2D}}} + \bar{\mu}_{3D}) + \Delta\mu_{3D}, \hat{x}_{\text{2D}} = \Delta\sigma_{\text{2D}}(\bar{\sigma}_{\text{2D}}\frac{x-\mu_{\text{3D}}}{\sigma_{\text{3D}}} + \bar{\mu}_{\text{2D}}) + \Delta\mu_{\text{2D}}$

Table 1. Batch Normalization (BN), Adaptive Batch Normalization (AdaBN), and Dimension Normalization (DN). γ, β and Δ are learnable parameters, whereas μ and σ indicate mean and variance.

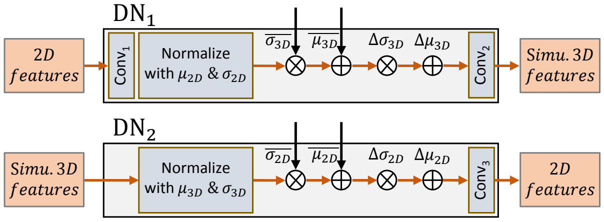

Figure 3. Dimension normalization modules. Note that $\otimes$ means element-wise multiplication; $\oplus$ means element-wise addition; and Conv$_1$ to Conv$_3$ are additional convolutional layers.

- The purpose of DN$_1$ is to transform the input 2D feature to align its distribution with that of the 3D feature for effective feature learning. Here, we first use two convolution layers of 3×3 and 1×1 (Conv$_1$ in Figure 3) to transform the input features to have the same channel dimension as the penultimate layer in 3D network f_{3D}. Then, we normalize the transformed features by μ_{2D} and σ_{2D}, and use $\bar{\mu}_{\text{3D}}$ and $\bar{\sigma}_{\text{3D}}$ to scale and adjust the normalized features. Ideally, $\bar{\mu}_{\text{3D}}$ and $\bar{\sigma}_{\text{3D}}$ should correspond to f_{3D}. However, f_{3D} is available only in the training but not in the inference, so we pre-compute $\bar{\mu}_{\text{3D}}$ and $\bar{\sigma}_{\text{3D}}$ globally over the entire data, and use the pre-computed values in both training and inference. As a result, we further use learnable parameters $\Delta\sigma_{\text{3D}}$ and $\Delta\mu_{\text{3D}}$ in DN$_1$ to adjust the features, followed by two subsequent convolution layers of 3×3 and 1×1 (Conv$_2$ in Figure 3) to produce the simulated 3D features.
- The purpose of DN$_2$ is to calibrate the learned 3D feature $f_{\text{3D\_sim}}$ back to f_{2D} for smooth 2D-3D feature concatenation. Its structure follows that of DN$_1$ (Figure 3), except that we do not need any additional convolutional layer to pre-transform the input feature.

Different from standard batch normalization (BN) [20], which further learns a linear transform to enhance the representative capability as shown in Table 1, our 3D-to-2D distillation module modulates the normalized distribution with a global pre-calculated 3D statistics $\bar{\sigma}_{\text{3D}}$ and $\bar{\mu}_{\text{3D}}$ and then uses learnable offsets $\Delta\sigma_{\text{3D}}$ and $\Delta\mu_{\text{3D}}$ to further adjust the result. With 3D-to-2D distillation, we can explicitly align the distributions of the 2D and 3D features. Yet, our approach still retains the advantage of BN by normalizing features in a batch-wise manner during the training. Then, during the inference, μ_{2D} and σ_{2D} are replaced by the accumulated $\bar{\mu}_{\text{2D}}$ and $\bar{\sigma}_{\text{2D}}$ similar to BN.

Relation to BN and AdaBN. Fundamentally, BN is proposed to facilitate the training of deep neural networks by normalizing the data distribution in a batch as shown in Table 1. It helps to reduce the internal co-variant shift [20] or smooth the objective function [43]. However, BN is domain-dependent, which has side-effects in our cross-modality knowledge transfer task. Further, AdaBN [26] is proposed for domain adaptation. In the inference stage, AdaBN uses $(\bar{\mu}_{\text{3D}}, \bar{\sigma}_{\text{3D}})$ of the target domain instead of the accumulated $(\bar{\mu}_{\text{2D}}, \bar{\sigma}_{\text{2D}})$ in the source domain to normalize the features. AdaBN is designed for the case, in which the model is trained in the source domain and tested in the target domain, which is different from our setup where training and inference are both performed in the 2D domain.

3.3. Adversarial Training with Unpaired Data

With paired 2D-3D data, we can distill 3D feature with an L_2 loss, since we can correspond the 2D and 3D features in the same image domain. However, paired 2D-3D data are typically expensive to acquire in a large quantity, given the amount of works needed in data collection, calibration, and annotation. To this end, we design a new adversarial training approach to correlate 2D and 3D features and supervise the generation of the simulated 3D features.

There are two key insights in our approach. First, we observe that existing 2D and 3D datasets usually have common object categories, e.g., ScanNet-v2 [8] has 20 object categories in its 3D point clouds, whereas NYU-v2 [44] has 40 object categories in its 2D images; the 20 categories in ScanNet-v2 can all be found in the categories in NYU-v2. Hence, we propose to *correspond 2D and 3D features by their associated object categories*. Second, given a 3D feature of a certain category and another 2D feature of the same category, we propose a novel adversarial training model with a per-category discriminator. The goal of our model is to generate simulated 3D features solely from the 2D features, such that *the discriminator cannot differentiate the 3D features from the simulated one of the same category*. In this way, the 2D network should learn to generate simulated 3D features solely without requiring paired 2D-3D data.

Figure 2 (top-right) shows the optional module we designed for unpaired training. It has N individual discriminators, where N is the number of object categories common to the 2D and 3D data. We implement each discriminator using six fully-connected layers and denote the discriminator of category c as D_c. There are two kinds of inputs to D_c: (i) a 3D feature vector $f_{\text{3D\_i}}$ from 3D network or (ii) a simulated 3D feature vector $f_{\text{3D\_sim\_j}}$ from 2D network, and both should belong to category c. For each input to D_c, it should predict a confidence score that indicates whether the input feature vector comes from the 2D or 3D network.

Before we present how we train the whole framework for unpaired data, we first denote Φ_{2D} as the 2D network. During the training, the 3D network is fixed, and we alternatively train Φ_{2D} and the set of discriminators $\{D_c\}$, similar to the way the generator (like Φ_{2D}) and discriminator (like D_c's) are trained in a conventional GAN model.

- When we train the discriminators, we fix Φ_{2D} and use the following objective to train each D_c:

$$\begin{aligned} L_{\text{adv}}(D_c) = &-\sum_i \mathcal{N}_{c,i} \log(1 - D_c(f_{\text{3D\_sim\_i}})) \\ &-\sum_j \mathcal{M}_{c,j} \log(D_c(f_{\text{3D\_j}})), \end{aligned} \tag{3}$$

  where $\mathcal{N}_{c,i}$ (or $\mathcal{M}_{c,j}$) equals 1, if the 2D pixel i (or 3D point j) belongs to category c; otherwise, it equals 0. Note that we treat $f_{\text{3D\_sim\_i}}$ as a negative sample and $f_{\text{3D\_j}}$ as a positive sample, so the goal of D_c is to learn to differentiate them; by then, Φ_{2D} should learn to generate better $f_{\text{3D\_sim\_i}}$ to deceive $\{D_c\}$.

- When we train the 2D network Φ_{2D}, we fix all D_c's and use the following objective in the training:

$$L_{\text{adv}}(\Phi_{\text{2D}}) = -\sum_i \sum_c \mathcal{N}_{c,i} \log(D_c(f_{\text{3D\_sim\_i}})). \tag{4}$$

Overall, to train Φ_{2D}, the overall loss is constructed by combining the softmax-cross entropy loss [62] for semantic segmentation with the paired regression loss L_p or with the adversarial loss $L_{\text{adv}}(\Phi_{\text{2D}})$, if paired data is unavailable. The loss weights are validated on a small validation set.

4. Experiments and Results

Datasets We conduct experiments on three indoor scene parsing datasets—ScanNet-v2 [8], S3DIS [1], and NYU-v2 [31]. *ScanNet-v2* [8] contains 1,513 scenes with 3D scans and 2D images. *S3DIS* [1] contains 3D scans with associated 2D images of 271 rooms, and each 3D point and 2D pixel are annotated as one of the 13 categories in the dataset. To be noted, for both datasets, we can only obtain sparse paired 2D-3D points, due to the calibration and projection issues as illustrated in the supplementary material. Only 16.38% and 10.57% of 2D pixels have corresponding 3D points in ScanNet-v2 and S3DIS, respectively. *NYU-v2* [31] contains 1,449 images without reconstructed 3D scene.

Implementation details. We implement our framework using PyTorch [34], and train all the models and baselines with the SGD optimizer. The batch size is 16 and the initial learning rate is 0.01, which is scheduled based on the "poly" learning rate policy with power 0.9 [62]. Since different datasets have different number of training samples, we train models with different number of epochs on different datasets to equalize the number of training iterations: ScanNet-v2 for 50 epochs; S3DIS for 20 epochs on Area 1,2,3,4,6; and NYU-v2 with the officially split training samples for 300 epochs. By default, we empirically set the loss weights on L_s, L_{adv} and L_p as 1, 0.01 and 0.03, respectively.

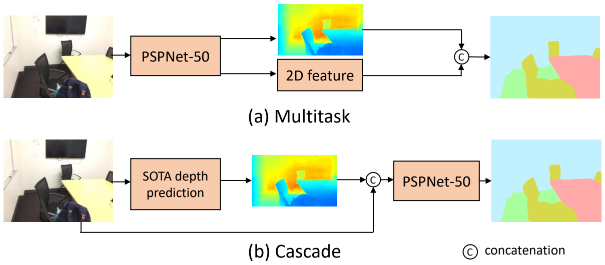

Figure 4. Illustrating the Multitask and Cascade approaches of leveraging depth for 2D semantic segmentation.

Network architectures. By default, we adopt the representative architecture PSPNet-50 [62] for semantic segmentation. For 3D feature extraction, we employ the voxel-based architecture, *i.e.*, MinkowskiNet [7] on ScanNet [8], and the point-based architecture, *i.e.*, PointWeb [60] on S3DIS. The 3D network weights are fixed during the training.

4.1. Comparing with Related Geometry-Assisted and Knowledge Distillation Methods

First, we present extensive experiments to compare our 3D-to-2D distillation approach with major competitors. They are alternative ways of leveraging depth or 3D information to enrich 2D features for semantic segmentation. In these experiments, we use the latest large-scale indoor scene parsing dataset ScanNet-v2 [8], which provides paired 2D-3D data for training various models for semantic segmentation.

(i) Baseline: the original PSPNet-50 model [62].

(ii) Multitask: we use PSPNet-50 to additionally predict a depth map (Figure 4 (a)), which is supervised by using data from ScanNet-v2; this model is similar to ours, in the sense that we replace the simulated 3D feature in our model by a depth map prediction.

(iii) Cascade: we use a state-of-the-art depth prediction method [24][1] to predict a depth map and take the predicted depth map as an auxiliary input to assist PSPNet (Figure 4 (b)); so, this approach also does not require 3D data in the inference like ours, but the knowledge comes from the depth map predicted by [24].

(iv) Geo-Aware [22]: this is the most recent method that distills features from depth maps for assisting 2D semantic segmentation; since code is not available, we

[1] https://paperswithcode.com/sota/monocular-depth-estimation-on-nyu-depth-v2

	Method	mIoU
Baseline	PSPNet-50	53.40
Depth methods	Multitask	54.08
	Cascade	53.72
	Geo-Aware [22]	56.90
KD	Structured KD	56.36
Ours	3D-2D Distillation	**58.22**

Table 2. Comparing the semantic segmentation performance (on ScanNet-v2 validation set) of various methods that use depth/3D information to assist the segmentation: two alternative methods (Multitask and Cascade) that predict depth maps, a depth-distillation method Geo-Aware [22], and Structured KD, which is a variant of [28]. All the methods are based on the ResNet-50 backbone.

simply report its ScanNet-v2 result on its paper in Table 2. Also, this method is finetuned on NYU-v2, so it leveraged more data than ours (ScanNet-v2 only).

(v) Structured KD: we adopt the most recent knowledge distillation approach for semantic segmentation [28]; since it is designed for distilling 2D information, we replace its teacher net with a 3D network to distill 3D information for comparison with our approach.

Table 2 reports the semantic segmentation performance of our approach and the competitors on the ScanNet-v2 validation set. Our approach outperforms all of them, and these results reveal the following:

- Comparing with Multi-task, we can show that our full approach of predicting the simulated 3D features leads to better results than explicitly predicting a depth map using the same paired data. This result demonstrates the richness of simulated 3D features from 3D point clouds, as compared with 2.5D depth maps.
- Comparing with Cascade, we can show that even we use the state-of-the-art depth prediction network [24] to provide the predicted depth map, the predicted depth map (which is 2.5D) cannot assist the PSPNet for semantic segmentation, as good as the simulated 3D features generated by our 3D-to-2D distillation approach.
- Comparing with Geo-Aware [22], we show again that distilling features from point clouds can lead to a better performance. Note that Multitask, Cascade, and Geo-Aware are all depth-assisted methods.
- Comparing with Structured KD, we show that our 3D-to-2D distillation approach can lead to a higher performance than simply adopting a general knowledge distillation model. To be noted, Structured KD also adopts an adversarial objective to align the global segmentation map; see Section 4.4.

4.2. Further Evaluations with Sparse Paired Data

Next, we conduct further experiments on the latest large-scale indoor scene parsing dataset—ScanNet-v2 and S3DIS.

Method	mIoU
FCN-8s [29]	45.87
ParseNet [27]	47.72
DeepLab-v2 [4]	43.89
AdapNet [48]	47.28
DeepLab-v3 [5]	50.09
AdapNet++ [47]	52.92
Ours (w/ PSPNet-50)	**57.76**

Table 3. Comparing our method with existing 2D image-based semantic segmentation methods on the ScanNet-v2 validation set. We follow the settings in [47], where the size of the input image is 384×768 without left-right flip and multi-scale test.

Method	mIoU
HRNet [49] + OCR [56]	60.56
HRNet [49] + OCR [56] + Our 3D-2D distillation	**61.36**

Table 4. We adopt our method into the state-of-the-art 2D semantic segmentation approach, HRNet [49] + OCR [56], and further boost its performance. The networks use HRNet-W48 [49] as the backbone, with a comparable network complexity as ResNet-101.

Method	PSPNet-50	Ours
mIoU	43.65	**46.42**

Table 5. Semantic segmentation results on S3DIS.

ScanNet-v2 semantic segmentation. First, we compare our method with several 2D image-based semantic segmentation approaches. Table 3 reports the results, showing that our model outperforms all of them for 4.8% to 11.8% mIoU.

Further, since our method is generic, we can easily incorporate it into existing semantic segmentation architecture. Hence, we adopt it into the state-of-the-art 2D semantic segmentation network—HRNet+OCR [57, 49], which has demonstrated top performance on various datasets. As shown in Table 4, our 3D-to-2D distillation approach can further boost the performance of HRNet+OCR.

S3DIS semantic segmentation. Further, we experiment on S3IDS [1] with PointWeb [60] as the 3D network for extracting 3D features. Results in Table 5 show that our method helps improve the baseline model, suggesting the generality and effectiveness of our method for use with different 3D network architectures on a different dataset.

Qualitative analysis. Visual results on ScanNet-v2 and S3DIS are shown in Figures 5 and 6, respectively. The semantic segmentation quality has been consistently improved with our approach. The failure modes of the original PSPNet model are largely caused by occlusions (Figure 5: door, bathtub and toilet, Figure 6: chair and table), uncommon viewpoints (Figure 5: table), and confusions due to similar color and texture (Figure 6: column, and chair). In comparison, our approach can successfully segment these objects. This demonstrates the potential of our distilled 3D feature for resolving issues with occlusions, viewpoints, and textures.

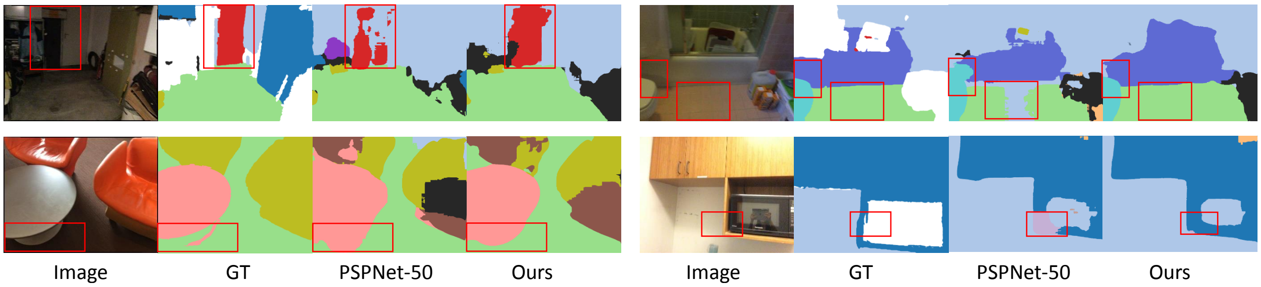

Figure 5. Visualization of the results on the ScanNet-v2 dataset. As marked by the red boxes, our 3D-to-2D distillation has better performance for the door, floor, table, and cabinet regions benefited from the rich embedded 3D information.

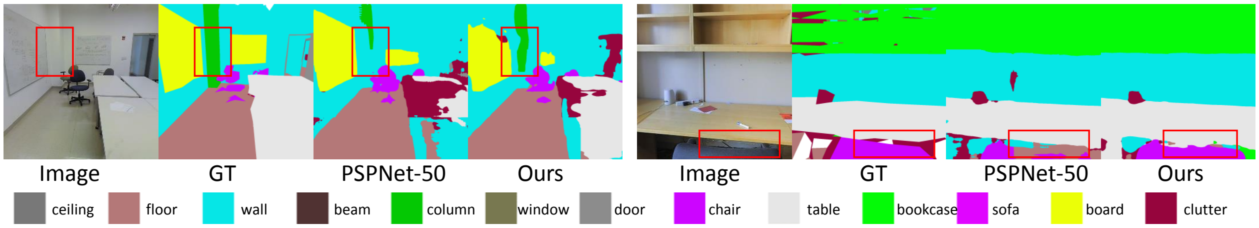

Figure 6. Visualization of the results on S3DIS. Ours show the result of our approach using the ResNet-50 baseline as the backbone.

Method	PSPNet-50	Ours
mIoU	49.50	**51.70**

Table 6. Results on NYU-v2 20 class without paired data.

This demonstrates that our model with distilled 3D feature can potentially better leverage 3D geometric information, such as the shape of objects and thus is more robust to ambiguities caused by occlusions, viewpoints and texture.

4.3. 3D-to-2D Distillation without Paired Data

When paired data is not available, we generally can only train the 2D network solely with 2D image inputs. With our adversarial training model (Section 3.3), we may distill unpaired 3D data to enrich features in the 2D network. This subsection presents an experiment to compare the performance of a 2D network (*i.e.*, PSPNet [62]) when it is trained (i) without 3D data and (ii) with unpaired 3D data. Here, we use MinkowskiNet [7] as the 3D network, NYU-v2 as the 2D data, ScanNet-v2 as the 3D data, and the 20 categories common to both data in training and testing.

From Table 6, we can see that without using any 3D data, the average performance of PSPNet on NYU-v2 20 classes is 49.50, and using unpaired 3D data (with almost negligible effort) can enrich the 2D features and improve the performance by almost 5% relatively, *i.e.*, from 49.50 to 51.70. Note that this is the very first work that explores the potential of using unpaired 3D data for 2D semantic segmentation. We did not explore more sophisticated techniques to further improve the results, but still, the results demonstrate the possibility of unpaired data, and we hope that this can open up a new direction for improving scene parsing from images.

Dataset	Ours (w/ BN)	Ours (w/ DN)
ScanNet-v2 S3DIS NYU-V2†	57.64 45.57 26.54	**58.22** **46.42** **27.22**
Dataset	Ours (w/o semantic)	Ours (w/ SAAL)
NYU-v2*	50.86	**51.84**

Table 7. Ablation studies. NYU-v2† is evaluated under the setting that the model is trained on ScanNet-V2 and tested on NYU-v2. NYU-v2* is evaluated with unpaired 2D-3D data. ScanNet-v2 and S3IDS results are evaluated on the setting with paired 2D-3D data. SAAL denotes our semantic aware adversarial loss.

4.4. Ablation Studies

Next, we ablate two major components in our approach: (i) the dimension normalization modules—we replace the two DN modules with BNs while keeping the convolution part in DNs; (ii) the semantic aware adversarial loss (SAAL)—we create a baseline that adopts a shared discriminator for all the categories, similar to that in [28].

From the results shown in Table 7, by comparing "Ours (w/ BN)" *vs.* "Ours (w/ DN)", we can see that DN consistently outperforms BN including the generalization to unseen domains, *i.e.*, NYU-v2†. Then, by comparing our SAAL with the ablated semantic aware design in the NYU-v2* row, we can see that SAAL boosts the performance, implying that our adversarial training model helps improve the discriminative ability of the features, while aligning the distributions to facilitate feature transfer.

4.5. Analysis

Further, we conduct an analysis to investigate whether our 3D-to-2D distillation can embed 3D information into the 2D CNN features and whether the 3D features can improve

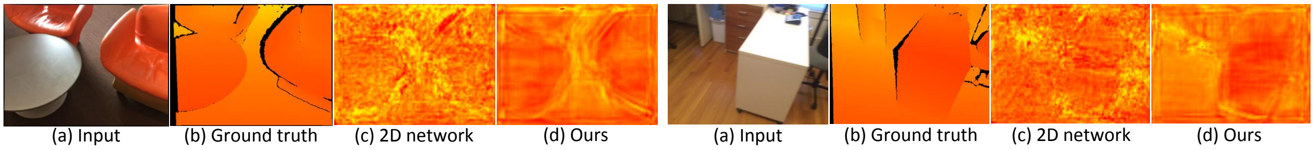

Figure 7. Visualization of depth reconstruction quality. Depth map of PSPNet-50 and ours are both derived from "res-block-4" features.

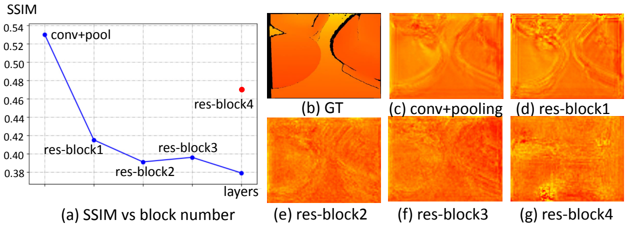

Figure 8. PSPNet-50 retains less 3D information in deep layers *vs*. shallow layers. (a) SSIM (larger is better) of depth images predicted from PSPNet-50 feature map in different layers. (b)-(g): Depth maps predicted from different layers. The PSPNet-50 is pre-trained for semantic segmentation. Please zoom in to see the details.

the model's generalizability. To start, we propose a metric to evaluate how much 3D information is embedded into the 2D CNN features. Our idea is to evaluate the ability of CNN features for reconstructing depth maps. We build a depth estimation network with features from each layer as input and train it to produce the corresponding depth map. Then, we evaluate the depth reconstruction quality with the Structural Similarity Index (SSIM) [52]—larger is better. The reason for adopting SSIM instead of Mean Square Error is that depth information from a single image has scale ambiguities, while SSIM focuses more on structural similarities with the ground truth rather than the absolute values.

3D features in baseline PSPNet. Taking inputs from different layers of PSPNet-50 trained on ScanNet-v2 for semantic segmentation, we obtain the depth prediction results shown in Figure 8. Figure 8 (a) shows that the SSIM value decreases as the feature map goes deeper. The associated visualizations in Figures 8 (b)-(g) show that the structure of the depth map becomes hard to identify when it is derived from deep features (Figure 8 (g)). Both quantitative and qualitative results suggest that deep features may retain less 3D information and have a lower capability of reconstructing depth in comparison with shallow features, even though they are more directly relevant to the final semantic parsing task.

3D features in our models. Further, we analyze whether the 2D network in our framework can better leverage 3D information. We take the deep "res-block-4" feature map of our model trained on ScanNet-v2. The SSIM is improved by 24% (baseline: 0.38 *vs*. our: 0.47) as compared with the baseline (the red point in Figure 8), and the quality of the reconstructed depth map is significantly improved as shown in Figure 7. The object structure and boundary can be better preserved. With 3D-to-2D distillation, higher quality depth can be reconstructed from our deep features, suggesting that our approach facilitates the 2D network to better utilize 3D information when constructing the deep features.

Method	PSPNet-50	PSPNet-101	Multitask	Cascade	Ours
mIoU	19.08	20.12	20.30	22.52	**27.22**

Table 8. Domain generalization results on NYU-v2 20-class using various models trained on ScanNet-v2.

4.6. Effectiveness in Boosting Model Generalization

The analysis in Section 4.5 implies that the embedded 3D information in deep features may help the CNN better utilize the robust 3D cues, such as shape, for recognition, and further improve the model's generalization abilities. Next, we investigate whether 3D-to-2D distillation can improve the generalizability of the model. To this end, we directly evaluate the baseline model and our model when trained on ScanNet-v2 and tested on NYU-v2. Here, we consider only the 20 classes common to ScanNet-v2 and NYU-v2.

Comparing Tables 8 and 2, we can see that the performance of all methods drops seriously when tested on unseen NYU-v2. However, with the 3D-enhanced 2D features, our model with the ResNet-50 backbone improves over PSPNet-50 by 43% for more than 8% mIoU. Also, it surpasses PSPNet-101 by more than 7% mIoU. The results imply that the embedded 3D feature helps improve the generalizability of the 2D network, and such improvement cannot be achieved by using a larger network, *i.e.*, PSPNet-101.

5. Conclusion

This paper presents a novel 3D-to-2D distillation framework that effectively leverages 3D features learned from 3D point clouds to enhance 2D networks for indoor scene parsing. At testing, the 2D network can infer simulated 3D features without any 3D data input. To bridge the statistical distribution gap between the 2D and 3D features, we propose a two-stage distillation normalization module for effective feature integration. Further, to broaden the applicability of our approach, we design an adversarial training model with the semantic aware adversarial loss to extend our framework for training with unpaired 2D-3D data. Experiments on three public indoor datasets suggest the superiority of our approach in various settings. We hope our further analysis on 3D and generalization could inspire future works on incorporating 3D information to improve model generalization.

References

[1] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 39(12):2481–2495, 2017.

[3] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(4):834–848, 2017.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3029–3037, 2017.

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

[10] Liuyuan Deng, Ming Yang, Tianyi Li, Yuesheng He, and Chunxiang Wang. RFBNet: deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. *arXiv preprint arXiv:1907.00135*, 2019.

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[12] Nuno C. Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[13] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. Learning with privileged information via adversarial discriminative modality distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 42(10):2581–2593, 2019.

[14] Ben Graham. Sparse 3D convolutional neural networks. In *The British Machine Vision Conference (BMVC)*, 2015.

[15] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from RGB-D images for object detection and segmentation. In *European conference on computer vision (ECCV)*, pages 345–360. Springer, 2014.

[16] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836, 2016.

[17] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2940–2949, 2020.

[18] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3771–3778, 2019.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4867–4876, 2020.

[22] Jianbo Jiao, Yunchao Wei, Zequn Jie, Honghui Shi, Rynson WH Lau, and Thomas S. Huang. Geometry-aware distillation for indoor semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2869–2878, 2019.

[23] Abhijit Kundu, Xiaoqi Michael Yin, Alireza Fathi, David Alexander Ross, Brian Brewington, Tom Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3D semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[24] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[25] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed points. In *Advances in neural information processing systems (NeurIPS)*, pages 820–830, 2018.

[26] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2017.

[27] Wei Liu, A Rabinovich, and C. Berg Alexander. ParseNet: Looking wider to see better. In *International Conference on Learning Representations (ICLR)*, 2016.

[28] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense predic-

tion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2020.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[30] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019.

[31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conferenece on Computer Vision (ECCV)*, 2012.

[32] Shivam Pande, Avinandan Banerjee, Saurabh Kumar, Biplab Banerjee, and Subhasis Chaudhuri. An adversarial approach to discriminative modality distillation for remote sensing image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 0–0, 2019.

[33] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[36] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems (NeurIPS)*, pages 5099–5108, 2017.

[37] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3D graph neural networks for RGBD semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5199–5208, 2017.

[38] Xiaojuan Qi, Zhengzhe Liu, Jianping Shi, Hengshuang Zhao, and Jiaya Jia. Augmented feedback in semantic segmentation under image level supervision. In *European Conference on Computer Vision (ECCV)*, pages 90–105. Springer, 2016.

[39] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3577–3586, 2017.

[40] Siddharth Roheda, Benjamin S Riggan, Hamid Krim, and Liyi Dai. Cross-modality distillation: A case for conditional generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2926–2930. IEEE, 2018.

[41] Eduardo Romera, José M. Alvarez, Luis M. Bergasa, and Roberto Arroyo. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2017.

[42] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[43] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2483–2493, 2018.

[44] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012.

[45] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SplatNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018.

[46] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019.

[47] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision (IJCV)*, pages 1–47, 2019.

[48] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651. IEEE, 2017.

[49] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2020.

[50] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015.

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[52] Zhou Wang, Alan C. Bovik, Hamid R Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004.

[53] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9621–9630, 2019.

[54] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Padnet: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 675–684, 2018.

[55] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

[56] Yuan Yuhui and Wang Jingdong. OCNet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[57] Yuan Yuhui, Chen Xilin, and Wang Jingdong. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[58] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018.

[59] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4106–4115, 2019.

[60] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5565–5573, 2019.

[61] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. ICNet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[63] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018.

[64] Yu Zhao, Xiang Li, Wei Zhang, Shijie Zhao, Milad Makkie, Mo Zhang, Quanzheng Li, and Tianming Liu. Modeling 4D fMRI data via spatio-temporal convolutional neural networks (ST-CNN). In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 181–189. Springer, 2018.