Cross-view Semantic Segmentation for Sensing Surroundings

Bowen Pan[1], Jiankai Sun[2], Alex Andonian[1], Bolei Zhou[2]
[1]Massachusetts Institute of Technology [2]The Chinese University of Hong Kong

Abstract

Sensing surroundings is ubiquitous and effortless to humans: It takes a single glance to extract the spatial configuration of objects as well as the free space from the observation. To facilitate machine perception with such a surrounding sensing capability, we introduce a novel framework for cross-view semantic segmentation. In this framework, the View Parsing Network (VPN) is proposed to parse the first-view observations into a top-down-view semantic map indicating the spatial location of all the objects at pixel-level. The view transformer module contained in VPN is designed to aggregate the surrounding information collected from first-view observations in multiple angles and modalities. To mitigate the issue of lacking real-world annotations, we train the VPN in simulation environment and utilize the off-the-shelf domain adaptation technique to transfer it to real-world data. We evaluate our VPN on both synthetic and real-world data. The experimental results show that our model can effectively make use of the information from different views and multi-modalities. Thus the proposed VPN is able to accurately predict the top-down-view semantic mask of the visible objects as well as barely seen objects, in both synthetic and real-world environments[1].

1. Introduction

Recent progress in deep neural networks enables machine to segment a scene precisely into meaningful regions and objects [46, 24]. So-called semantic segmentation networks have been widely used in many fields, such as autonomous driving [12, 35], human-computer interaction and robotics [38, 37, 34]. Though the semantic segmentation network is able to recognize semantic content in a static image, it is still far from enough to facilitate the machine to sense in an unknown environment and navigate freely there. The parsed first-view semantic mask is still at pure image level without providing any information about the spatial structure of the surroundings explicitly. There are many attempts to extract spatial knowledge from the image input [8, 3, 6, 7], however, the agents perceive the environment only in order to obtain their current location instead of having a deeper understanding of the surrounding objects. On the other hand, it takes excessive cost to rely on full 3D methods to reconstruct the environment then obtain the spatial structure. Meanwhile 3D method is sometimes fragile in the condition of limited view fields. To parse spatial configuration from pure image input, an intuitive approach is to explicitly train networks to infer the top-down-view semantic map which directly contains the spatial configuration information of the surrounding environment. Based on the top-down-view semantic map we can further know the position coordinates and functional properties of surrounding regions and objects.

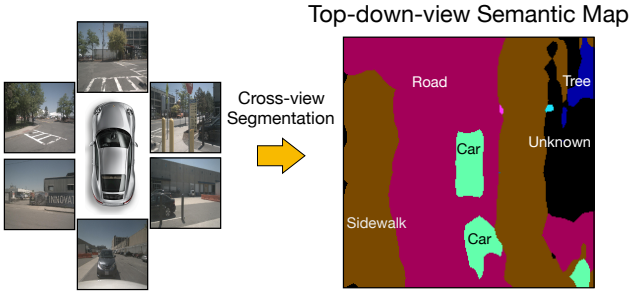

Figure 1: Top-down-view semantics are predicted from the first-view real-world observations in the cross-view semantic segmentation. Input observations from multiple angles are fused together.

In order to better capture the spatial structure of the surroundings, we explore a new image-based scene understanding task, *Cross-View Semantic Segmentation*. Different from the naive semantic segmentation which predicts the labels of each pixels in the input image, cross-view semantic segmentation aims at predicting the top-down-view semantic map from a set of first-view observations (cf. Figure 1). The resulting top-down-view semantic map, as a 2.5D spatial representation of the surrounding, naturally outlines the spatial layout of discrete objects and stuff classes such as floor and wall. Such a light spatial representation facilitates the machine to understand its surrounding more efficiently and exhaustively.

One challenge in cross-view semantic segmentation is the difficulty of collecting the top-down-view semantic an-

[1]Code and demo video are available at the project page: https://view-parsing-network.github.io

notations. Recently, realistic simulation environments such as House3D [39], Matterport3D [10, 41] and CARLA [13] have been proposed for training navigation agents. In these environments, cameras can be placed at any location in the simulated scene while the observations in multiple modalities such as RGB images, semantic masks, and depth maps can be extracted. Thus, leveraging simulation environments is an alternative way to acquire cross-view annotated data. Considering the domain gap between synthetic scenes and real-world scenes, we adapt the model trained in simulation environment for the real-world scenes. There is a huge literature of knowledge transfer and domain adaptation techniques in classical semantic segmentation area [36, 1]. Some of them can achieve pretty good performance and can be applied to the cross-view semantic segmentation with just slight modification.

In this work we propose a novel framework with View Parsing Network (VPN) for cross-view semantic segmentation using simulation environments, and then transfer them to real-world environments. In VPN, a view transformer module is designed to aggregate the information from multiple first-view observations with different angles and different modalities. It further outputs the top-down-view semantic map with spatial layout of objects. We evaluate the proposed models on indoor scene of the House3D environment [39] and outdoor driving scene of the CARLA environment [13]. Experiments show that our model achieves 85.0% pixel accuracy and 41.0% mIoU on synthetic data of House3D as well as 84.7% pixel accuracy and 33.2% mIoU in CARLA. We also provide the domain adaptation result on nuScenes [9], a real-world dataset for autonomous driving on which we get the performance of 78.8% pixel accuracy, showing the potential of our work for parsing real-world data.

Our main contributions are as follows: (1) We introduce a novel task named **cross-view semantic segmentation** to help agents flexibly sense surrounding environment. (2) We propose a framework with **View Parsing Network** which effectively learns and aggregates features across first-view observations with multiple angles and modalities. (3) We further adapt our model, mainly by modifying current domain adaptation techniques for classical semantic segmentation, to work in real world while without any extra annotations.

2. Related Work

Semantic segmentation. The semantic segmentation task generates a pixel-wise semantic map to label each pixel of the input image. Deep learning networks for semantic segmentation, such as the FCN [24], SegNet [2] and PSPnet [45] are designed to segment the image pixel-wise within one-view. Image datasets with pixel-wise annotations such as CityScapes [12] and ADE20K [46] are used for the training of semantic segmentation networks. Note these pixel-wise annotated datasets require a large amount of annotation. In our cross-view segmentation task, we leverage the coupled cross-view annotation data pulled from the simulated environments for free as training data, and explore the corresponding cross-view segmentation network architectures.

Layout estimation and view synthesis. Estimating layout has been an active topic of research (i.e. room layout estimation [49, 21, 42, 18, 16], free space estimation [17]). Most of the previous methods use annotations of the layout or geometric constraints for the estimation, while our proposed framework estimates the top-down-view map directly from the image, without the intermediate step of estimating the 3D structure of the scene. On the other hand, view synthesis has been explored in many works [22, 47, 44, 29, 43, 19]. For example, [44, 29] describes a method to learn the transformation between ground and aerial imagery. [19, 43] synthesizes cross-views of objects and [47, 22] synthesizes driving scenes. View synthesis focuses on generating realistic cross-view images while our cross-view segmentation aims at parsing semantics across different views.

Simulated environment learning. Given that current graphics simulation engines can render realistic scenes, recognition algorithms can be trained on data pulled from simulation engines (i.e. for visual navigation models [14, 48, 41, 11, 20]). Several techniques have been proposed to address the domain adaptation issue [40, 4, 30, 33, 23, 25, 5], when models trained with simulated images are transferred to real scenes [1]. Rather than working on the task of visual navigation directly, our project aims at parsing the top-down-view semantic map from first-view observations. The resulting top-down-view map will further facilitate the visual navigation and path planning.

3. Cross-View Semantic Segmentation

The task of parsing top-down-view semantics from scene images is much less explored than object detection and segmentation. Thus we propose a framework called **View Parsing Network** which takes multiple first-view observations from different angles and modalities as input, then output a top-down-view semantic map. The resulting top-down-view semantic map indicates the spatial layout and relations among objects, which is crucial for deeper awareness of surrounding environments and mobile robot navigation.

3.1. Problem Formulation

The objective of cross-view semantic segmentation is as follows: given the first-view observations as input, the algorithm must generate the top-down-view semantic map. Given axes of the original frame x, y, z where z axis is vertically up, the axes of the rotated frame as X, Y, Z, and N axis which is $z \times Z$, we define α, β, γ as the angles between x axis and N axis, z axis and Z axis, N axis and X axis, respectively. The top-down-view semantic map is a map captured by a camera at a certain height from the top-down view

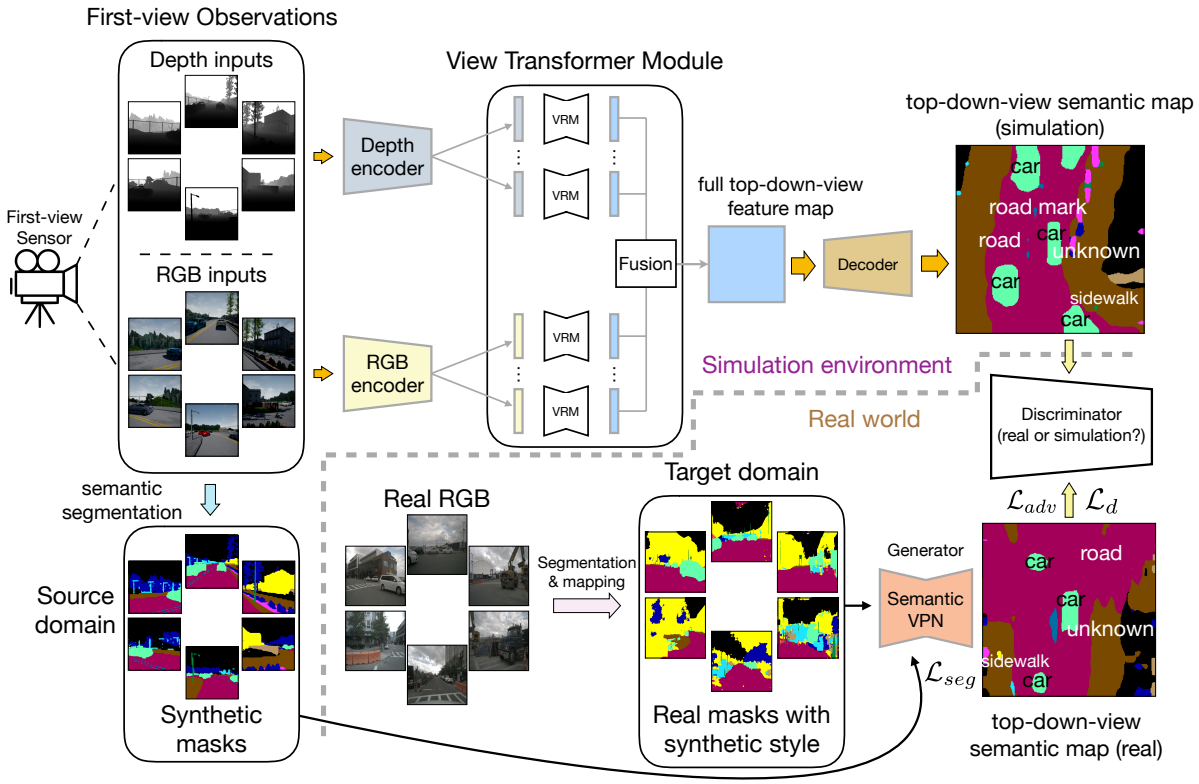

Figure 2: Framework of the View Parsing Network for cross-view semantic segmentation. The simulation part illustrates the architecture and training scheme of our VPN. And the real-world part demonstrates our domain adaptation process for transferring our VPN to the real world.

($[\alpha, \beta, \gamma] = [0, \pi, 0]$) with the annotations of the semantic label of each pixel. The top-down-view semantic map not only contains the semantic labels but also tells the approximate coordinate information of each object in the scene. The input first-view observations can be a single image or a set of images with different modalities. And they are captured at different angles of α while keep the $\beta = \frac{\pi}{2}, \gamma = 0$ from the same spatial location. In our basic setting, the network predicts the semantic map at local spatial positions. We are also integrating multiple local semantic maps into a semantic floor map for further applications.

Note that there is a fundamental difference between the cross-view semantic segmentation and the visual SLAM [27, 28]. While both of them estimate the spatial configuration of surrounding environment, our VPN aims at parsing the top-down-view semantic map rather than the full 3D reconstruction of the scene which is targeted by visual SLAM.

3.2. Framework of the View Parsing Network

Figure 2 illustrates two stages of our framework. In the first stage, we propose View Parsing Network (VPN) to learn and aggregate features from multiple first-view observations in simulation environment. In VPN, first-view observations are first fed into the encoder to extract first-view feature maps. Notice that the first-view observations can be in different modalites, such as RGB, depth, semantics, and for each modality, VPN has a corresponding encoder to process it. All of these first-view feature maps from different angles and different modalites are transformed and fused in the **View Transformer Module**. Then the aggregated feature map is decoded into a top-down-view semantic map. In the second stage of our framework, we transfer the knowledge which VPN learns from simulation environment to the real-world data. We slightly modified the domain adaptation algorithm proposed by [36] to fit our cross-view semantic segmentation task and our VPN architecture. More details of this part will be revealed in Section 3.3.

Pipeline. As shown in Figure 2, from one spatial position in a 3D environment, we first sample $N \times M$ first-view observations from N angles and M modalities (here $N = 4, M = 2$ in Figure 2) in even angles so that all-around information is captured. For example, for a 4-view model, we take the first-view inputs with angle α of $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$. The first-view observations are encoded by M encoders for M corresponding modalities respectively. Notice that encoder for inputs with same modality but different view angles are share-weights since we believe that feature patterns within same modality should be consistent. These CNN-based encoder extract $N \times M$ spatial feature maps for their first-view input. Then all of these feature maps are fed into the View Transformer Module (VTM). VTM transforms these view feature maps from first-view space into the top-down-

view feature space and fuse them to get one final feature map which already contains sufficient spatial information. Finally, we decode it to predict the top-down-view semantic map using a convolutional decoder.

View Transformer Module. Although the encoder-decoder structure gets huge success in the classical semantic segmentation area [24, 45, 46, 2], our experiment (cf. Table 3) shows that it performs poorly in the cross-view semantic segmentation task. The reason is that in the standard semantic segmentation, the receptive field of the output spatial feature map is roughly aligned with the input spatial feature map. However, it is not the case for the cross-view semantic segmentation, where the input and output features are not spatially aligned. Ideally, in cross-view semantic segmentation, each pixel on the top-down-view map should take all input first-view feature maps into consideration, not just at a local receptive field region.

After thinking about the flaws of current semantic segmentation structure, we design the View Transformer Module (VTM) to learn the dependencies across all the spatial locations between the first-view feature map and the top-down-view feature map. VTM will not change the shape of input feature map, so it can be plugged into any existing encoder-decoder type of network architecture for classical semantic segmentation. It consists of two parts: View Relation Module (VRM) and View Fusion Module (VFM). The diagram at the central of Figure 2 illustrates the whole process: The first-view feature map is first flattened while the channel dimension remains unchanged. Then we use a view relation module R to learn the relations between the any two pixel positions in flattened first-view feature map and flattened top-down-view feature map. That is:

$$t[i] = R_i(f[1], ..., f[j], ..., f[HW]), \tag{1}$$

where i, j is the indexes of top-down-view feature map $t \in R^{HW \times C}$ and first-view feature map $f \in R^{HW \times C}$ respectively along the flattened dimension. Here we simply use multilayer perceptron (MLP) to be our view relation module R. After that, the top-down-view feature map is reshaped back to $H \times W \times C$. Notice that each first-view input has its own VRM to get the top-down-view feature map $t^i \in R^{H \times W \times C}$ based on its own observation. To aggregate the information from all observation inputs, we fuse these top-down-view feature map t^i by using VFM. That is:

$$f_{out} = V(t^1, t^2, ..., t^{MN}), \tag{2}$$

where f_{out} is the final top-down-view output feature, N is the number of input first-view angles, M is the number of input modalities, t^i is the top-down-view feature map from i^{th} observation input and V is the fusion function.

Input observations. The first-view input observations can be in three modalities: RGB image, depth map, and semantic mask. *RGB image* contains the appearance information of the scene including texture, color, and illumination. *Depth map* contains the geometry information of the objects to the camera and their shapes. *Semantic mask* provides the semantic information of each pixel. In practice, it is plausible to integrate the observations from multiple modalities when we deploy the algorithm to the actual mobile robot: both RGB image and depth map can be directly obtained by an RGB-D camera, while semantic masks can be predicted by existing semantic segmentation networks from the RGB images [46, 32].

3.3. Sim-to-real Adaptation

To generalize our VPN to real-world data without the real-world ground truth, we implement the sim-to-real domain adaptation scheme shown in Figure 2 to narrow the gap. This scheme contains the following pixel-level adaptation and output space adaptation.

Pixel-level adaptation. In order to mitigate the domain shift, we adopt the pixel-level adaptation on the real-world inputs to make them look more like the style of the simulation data. Considering that it is still difficult to perfectly transfer the real RGB pixels to the simulation space, we transform the RGB data in both real world and simulation environment to semantic masks. In our cross-view semantic segmentation task, semantic mask is an ideal mid-level representation without texture gap while including sufficient information and it is easy to transfer. In simulation environment, semantic masks can be easily generated and we can use them to learn to segment RGB images. And in real world, we use some existing segmentation models to extract the semantic mask from RGB images, and map the categories to match the categories in the simulation environment. Then we can obtain the synthetic-style real-world mask.

Output space adaptation. Beyond the pixel-level transfer on input data, we also devise an adversarial training scheme in structured output space mainly by modifying the method in [36]. Here the generator $\mathcal{G}$ is a view parsing network generating the top-down-view prediction P, and the discriminator $\mathcal{D}$ is used to discriminate if P is generated from source domain. The generator $\mathcal{G}$ is initialized by the weights of a VPN trained on the semantic data in simulation environment as we illustrated before. During the training phase, We first forward a group of input images from the source domain $\{I_s\}$ to $\mathcal{G}$ and optimize it with a normal segmentation loss $\mathcal{L}_{seg}$. Then we use $\mathcal{G}$ to extract the feature map F_i (before the softmax layer) of the images from the target domain $\{I_t\}$ and use discriminator to distinguish whether F_t is from the source domain. Notice that $\{I_s\}$ is a cluster of synthetic semantic masks in simulation environment and $\{I_t\}$ contains the real-world mask with synthetic style. Here $\mathcal{G}$ is updated by the gradients propagated from $\mathcal{D}$, while the weights of $\mathcal{D}$ is fixed, which encourages $\mathcal{G}$ to unify the feature distributions of source domain and target domain. Lastly, we optimize the discriminator by training $\mathcal{D}$

Table 1: Results on House3D cross-view dataset with different modalities and view numbers.

	3D Geometric Baseline		RGB VPN		Semantic VPN		Depth VPN	
Networks	PA	mIoU	PA	mIoU	PA	mIoU	PA	mIoU
1-view model	3.9%	1.7%	55.8%	6.5%	59.6%	13.2%	56.9%	7.6%
2-view model	5.9%	1.9%	70.1%	14.8%	75.7%	25.9%	70.2%	15.6%
4-view model	15.6%	5.5%	80.3%	27.2%	**85.0%**	40.6%	77.3%	22.0%
8-view model	29.3%	10.5%	**81.2%**	**28.5%**	84.7%	**41.0%**	**82.1%**	**29.9%**

Table 2: Results of cross-modality learning for VPN. Here we compare the results with the inputs from 4 views.

Method	Pixel Accuracy	Mean IoU
RGB VPN	80.5%	27.2%
Depth VPN	77.3%	22.0%
Semantic VPN	85.0%	40.6%
R+D (late fusion)	81.2%	27.3%
R-D VPN	**82.8%**	**31.2%**
D+S (late fusion)	83.5%	33.9%
D-S VPN	**86.3%**	**43.2%**
S+R (late fusion)	84.3%	35.7%
S-R VPN	**85.1%**	**42.3%**

Table 3: Ablation study of View Transformer Module.

Modality	VPN w/o VTM		VPN	
1-view	Pix. Acc.	Mean IoU	Pix. Acc.	Mean IoU
RGB	53.9%	6.3%	**55.8%**	**6.5%**
Depth	55.7%	6.5%	**56.9%**	**7.6%**
Semantic	57.4%	10.0%	**59.6%**	**13.2%**
8-view	Pix. Acc.	Mean IoU	Pix. Acc.	Mean IoU
RGB	60.5%	8.7%	**81.2%**	**28.5%**
Depth	43.8%	2.5%	**82.1%**	**29.9%**
Semantic	47.6%	6.5%	**84.7%**	**41.0%**

to recognize which domain the feature output is from. The loss function to optimize $\mathcal{G}$ can be written as follows:

$$\mathcal{L}(\{I_s\},\{I_t\}) = \mathcal{L}_{seg}(\{I_s\}) + \lambda_{adv}\mathcal{L}_{adv}(\{I_t\}), \quad (3)$$

where the $\mathcal{L}_{seg}$ is a normal cross-entropy loss for semantic segmentation, and $\mathcal{L}_{adv}$ is designed to train the $\mathcal{G}$ and fool the discriminator $\mathcal{D}$. The loss function for the discriminator training $\mathcal{L}_d$ is a cross-entropy loss for binary source & target classification.

4. Experiments

We first go through the network configuration in Section 4.1 and the overview of the cross-view segmentation datasets in Section 4.2. Then we show the performance of VPN on synthetic data of the House3D and CARLA environment in Section 4.3. Finally in Section 4.4, we demonstrate the real-world performance of our VPN which is trained in simulation environment. In supplementary material, we further conduct two application experiments of VPN, the first of which shows that our VPN is able to integrate the local top-down-view patches into a holistic semantic floor map, and the other one demonstrates that our VPN can help the navigation agent to explore the unseen environment more efficiently.

4.1. Network configuration

View encoder and decoder. To balance efficiency and performance, we use ResNet-18 [15] as the encoder. We remove the last Residual Block and the Average Pool layer so that the resolution of the encoding feature map remains large, which better preserves the details of the view. We employ the pyramid pooling module used in the standard scene parsing [45] as the decoder.

View Transformer Module. For each view relation module, we simply use the two-layers MLP. Input and output dimensions of the VRM are both $H_I W_I$, where H_I and W_I are respectively the height and width of the intermediate feature map. We flatten the intermediate feature map to $C_I \times W_I H_I$ before we input and reshape it back to $C_I \times W_I \times H_I$ after that. As for the view fusion module, we just add all the feature up to keep the shape consistent. Notices that we can actually use more sophisticated designs for VRM and VFM. For example, any non-local layer which can model relations between one output pixel and all of the input pixels can be use in VRM, and some fusion techniques with attention mechanism can also be applied to VFM. However, we currently do not focus on design a complicated architecture but do care about the whole pipeline of the cross-view semantic segmentation. Thus we just use the most simplified architecure to show our methodology.

Sim-to-real. For the generator $\mathcal{G}$, we use the architecture of the 4-view VPN. For the discriminator $\mathcal{D}$, we adopt the same architecture in [36]. It has 5 convolution layers, each of which is followed by a leaky ReLU [26] with the parameter 0.2 (except the last layer). We use HRNet [32] pretrained on CityScapes dataset [12] to extract the semantic mask from real-world images.

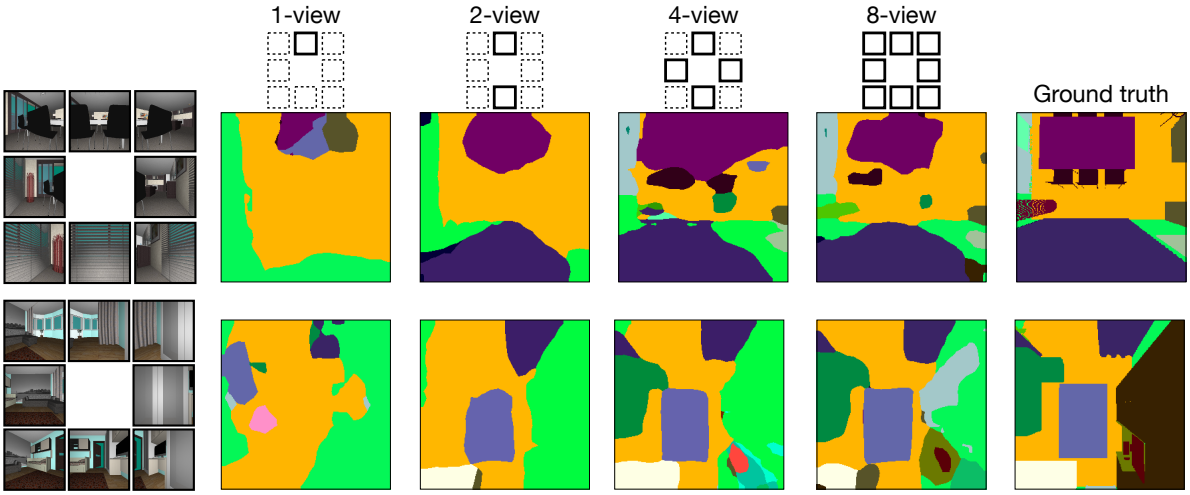

Figure 3: Cross-view segmentation result improves when the VPN receives more RGB views as input. The first column shows the input first-view RGB images at 8 evenly views. Other columns show the segmentation results by using 1-view, 2-view, 4-view and 8-view VPNs along with the ground truth respectively.

4.2. Benchmarks

Here we introduce **two synthetic cross-view datasets**, *House3D cross-view dataset* and *Carla cross-view dataset*, and **one real-world cross-view dataset**, *nuScenes dataset*.

House3D cross-view dataset. We build a synthetic indoor-room dataset from House3D environment [39]. House3D is an interactive graphic environment built on top of the 3D indoor scenes of SUNCG dataset [31]. In our experiments, we select 410 scenes in House3D to construct a dataset with cross-view data annotations. We refer to this dataset as *House3D Cross-view Dataset*. Each data pair contains 8 first-view input images captured from 8 different orientations (with angle α of $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$ respectively) at a spatial location of a scene. Additionally, each data pair comes with the top-down-view semantic mask captured in the ceiling-level height. To be complete, we store the input image with multiple modalities including the RGB images, depth maps, and semantic masks. In each scene, we sample the data with a 0.5-meter stride over the floor map. Here each scene is an independent house with different rooms and objects. We split the dataset into training and validation set based on scenes. The training set contains 143k data pairs from 342 scenes while the validation set contains 20k data pairs from 68 scenes.

NuScenes dataset. NuScenes is a public large-scale dataset for autonomous driving which contains 1000 driving scenes in Boston and Singapore. Each scene has a 20-second-length driving clip consists of multiple data samples. And each data sample contains first-view RGB images from 6 directions (*Front, Front-right, Back-right, Back, Back-left, Front-left*) in different modalities. NuScenes also provide the map extension of whole scene. Thus we utilize this map to extract the local top-down view map for each data sample. However, categories on the map extension focus more on the functional properties of each part of road, such as lane and parking lot, which do not exactly match our semantic model trained in simulation environment. To effectively use the map information, we only extract the local top-down-view map with only drivable area information to quantitatively evaluate our VPN in real world. We select 919 data samples without top-down-view mask for unsupervised training and 515 data samples with binary top-down-view mask for evaluation.

CARLA cross-view dataset. To match the data composition in NuScenes, we use CARLA simulator to extract data. CARLA is a popular open-source simulator for training and evaluation of autonomous driving. To build the synthetic source domain dataset, we extract $28,000$ data pairs with top-down-view annotations and different input modalities from 14 driving episodes. Each data pair contains 6 first-view input image sets captured from 6 directions (*Front, Front-right, Back-right, Back, Back-left, Front-left*).

4.3. Evaluation

We present VPN performances on the synthetic data of House3D cross-view and CARLA cross-view datasets.

Metrics. We report the results of cross-view semantic segmentation using two commonly used metrics in semantic segmentation: PIXEL ACCURACY (PA) which characterizes the proportion of correctly classified pixels, and MEAN IOU (MIOU) which indicates the intersection-and-union between the predicted and ground truth pixels.

Baselines. Since there is no previous work experiments on this task, we propose an intuitive 3D geometric method to be our baseline. More specifically, with the observed depth images and semantic mask, we can reconstruct the 3D points

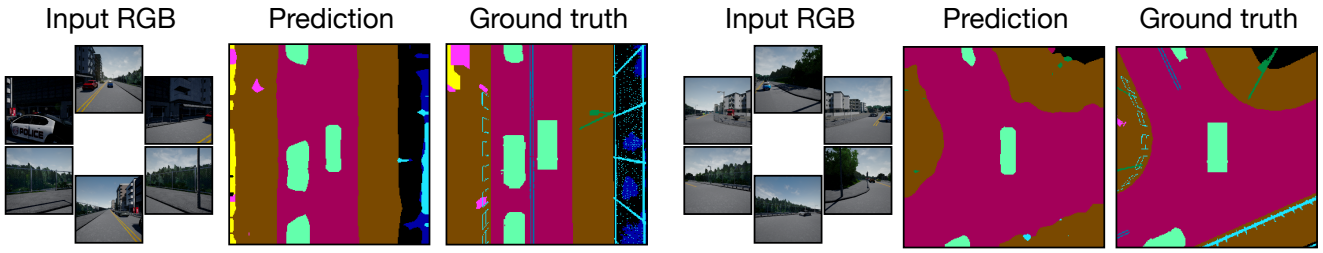

Figure 4: Qualitative cross-view segmentation results of a 6-view RGB VPN model trained on CARLA cross-view dataset. We train such a model in order to match the data modality and format of nuScenes dataset.

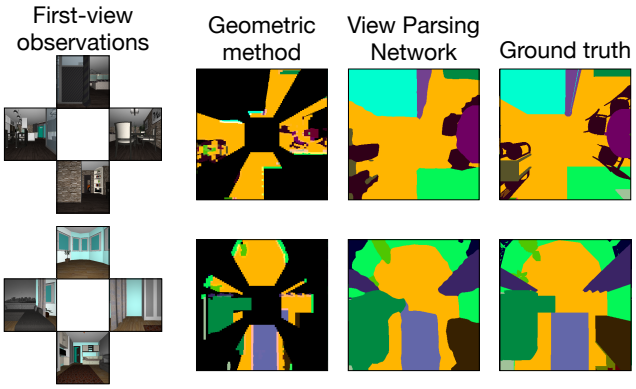

Figure 5: Qualitative results of 3D geometric method and our VPN. Considering that the geometric method requires semantic mask and depth map, we use the 4-view Depth-Semantic VPN to predict the top-down-view semantic map to fairly compare these two methods.

cloud with voxel-level semantic label. For each pixel on the depth image, we back-project it from pixel coordinates to the 3D coordinates in world frame by using camera's intrinsic and extrinsic matrices. We then attach its semantic label from 2D semantic mask image to the voxel. We finally obtain the top-down-view semantic map by reprojecting 3D semantic points cloud into the top-down view.

4.3.1 Results of VPNs

We present the results of our VPN for cross-view semantic segmentation, including the ones of single-modality and multi-modalities VPN respectively. We also show the comparison with geometric baseline and the ablation study of View Transformer Module. And in Figure 4, we put some results of a 6-view RGB-input model trained in CARLA, which achieves the performance of 84.7% pixel acc. and 33.2% mIoU. We further show some interesting cases in supplementary materials, where we find that our model has learned to infer some invisible objects.

Single-modality VPN. We show the results of single-modality VPN with different modalities and different numbers of views in Table 1. We can see that as VPN receives more views, the segmentation results improve rapidly. Noticeably, taking semantic masks as input achieves the best performance. We also plot some qualitative results by our VPNs in Figure 3, 4, 5. In Figure 3, we keep the input modality fixed as an RGB image and vary the number of input views. We can see that, as the number of views increases, the segmentations are refined to capture more details. We also provide the segmentation results of the other modalities, such as depth and semantics, in supplementary materials.

Multi-modalities VPN. We demonstrate the results of multi-modalities VPN in Table 2 to show that our VPN can effectively synthesize information from multiple modalities. We set the late-fusion baseline to compare with our multi-modalities VPN, which simply averages the softmax outputs of each single-modality VPN to obtain the final results. We find that the Depth-Semantic VPN achieves the best performance and makes great improvement. This may be because semantic mask and depth map are two complementary information. However, the Semantic-RGB combination does not bring too much improvement. The reason can be that, for this cross-view semantic segmentation task, semantic input contains the most of the useful information in the RGB.

Comparing with baseline. Table 1 shows that our VPN can easily outperform the 3D geometric method. And the reason why there is a huge gap between the performance of geometric method and VPN is that geometric method is very easy to fail when there are obstacles. As we can see from Figure 5 that, with same first-view inputs, 3D geometric method is unable to reconstruct the objects which can not be directly observed, such as the desk behind the chairs. While our VPN can still work robustly even with the limited view.

Importance of View Transformer Module. We further compare with the baseline networks in Table 3 to show the importance of the view transformer module in aggregating the information from multiple views. The baseline network is a classic encoder-decoder architecture used in the standard semantic segmentation, in which the encoder and the decoder are same as our VPN. We train the baseline model with input view number as 1 and 8 respectively. We simply sum up the feature maps from different views and then feed it to the decoder. Our VPN easily outperforms the baseline and, in some multi-view cases, the baseline model does even worse than single-view one due to the bad fusion strategy.

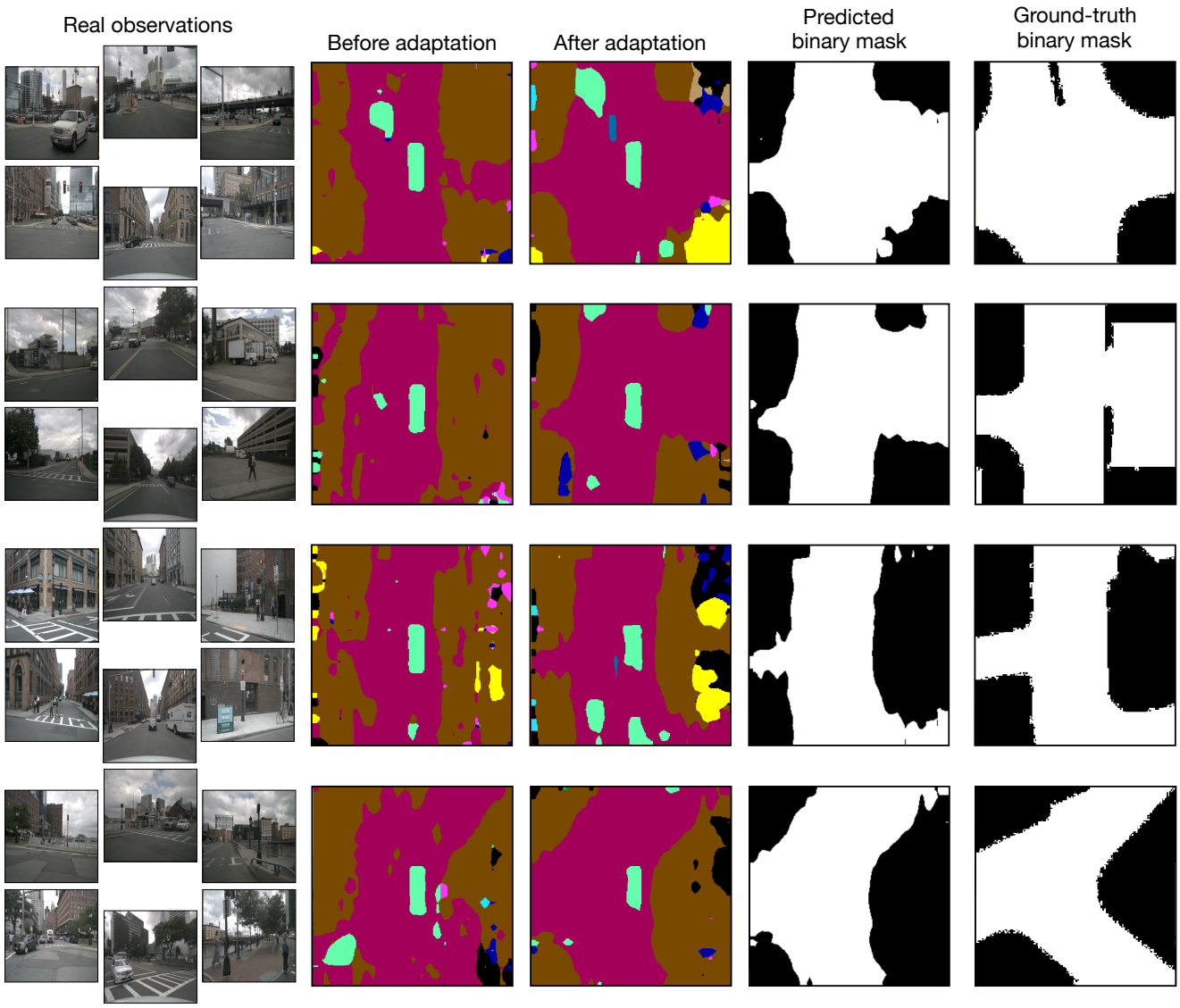

Figure 6: Qualitative results of sim-to-real adaptation. We provide the results of source prediction both before and after domain adaptation, drivable area prediciton after adaptation and the groud-truth drivable area map.

4.4. Results of sim-to-real adaptation

After we train and test our VPNs in simulation environment, we transfer our model to the real-world data. Notice that such a process is non-trivial since there is a huge gap between image appearance distributions in real world and CARLA simulator. We first train a 6-view semantic VPN model on the predicted semantic masks in CARLA simulator and then transfer it to nuScenes dataset by using unsupervised domain adaptation process as depicted in Section 3.3. We provide the qualitative results above in Figure 6, from which we can see that our VPN is able to roughly segment various road shape like crossroad and also sketch the relative locations of surrounding objects such as cars and buildings. Notice that our predicted masks are not strictly aligned with the ground-truth map since the camera poses in the NuScenes dataset are slightly different from those in CARLA environment. But we still evaluate the quantitative results of real-world performance by using binary drivable-area ground truth. The **pixel accuracy** of our predicted binary masks before and after adaptation are **72.6%** and **78.8%** respectively. Combined with our qualitative results, we can see an obvious improvement on VPN's performance in real world.

5. Conclusion

We first proposed cross-view semantic segmentation task to sense the environment. Then we designed a targeted architecture View Parsing Network (VPN) trained in simulation environment to do this task. At last, we transfer our VPN to real world by some domain adaptation techniques.

References

[1] Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018. 2

[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*, 2017. 2, 4

[3] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[4] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 2

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016. 2

[6] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 1

[7] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018. 1

[8] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[9] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 2

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2

[11] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2019. 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. 1, 2, 5

[13] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 2

[14] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 5

[16] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. ICCV*, 2009. 2

[17] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering free space of indoor scenes from a single image. In *Proc. CVPR*, 2012. 2

[18] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2017. 2

[19] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *Proc. CVPR*, volume 2, 2017. 2

[20] Ashish Kumar*, Saurabh Gupta*, David Fouhey, Sergey Levine, and Jitendra Malik. Visual memory for robust path following. In *Advances in Neural Information Processing Systems*, 2018. 2

[21] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4865–4874, 2017. 2

[22] Chien-Chuan Lin and Ming-Shi Wang. A vision based top-view transformation model for a vehicle parking assistant. *Sensors*, 12(4):4431–4446, 2012. 2

[23] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 2

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 1, 2, 4

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105. JMLR.org, 2015. 2

[26] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, page 3, 2013. 5

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 3

[28] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. ICCV*, pages 2320–2327. IEEE, 2011. 3

[29] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *Proc. CVPR*, pages 3501–3510, 2018. 2

[30] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from

simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2107–2116, 2017. 2

[31] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proc. CVPR*, 2017. 6

[32] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 4, 5

[33] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2

[34] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 International Conference on 3D Vision (3DV)*, pages 537–547. IEEE, 2017. 1

[35] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018. 1

[36] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018. 2, 3, 4, 5

[37] Daniel Wolf, Johann Prankl, and Markus Vincze. Enhancing semantic segmentation for robotics: The power of 3-d entangled forests. *IEEE Robotics and Automation Letters*, 1(1):49–56, 2015. 1

[38] Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. Segicp: Integrated deep semantic segmentation and pose estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5784–5789. IEEE, 2017. 1

[39] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 2, 6

[40] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[41] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 2

[42] Jiu Xu, Björn Stenger, Tommi Kerola, and Tony Tung. Pano2cad: Room layout from a single panorama image. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 354–362. IEEE, 2017. 2

[43] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *In Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 2

[44] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proc. CVPR*, volume 3, 2017. 2

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. CVPR*, 2017. 2, 4, 5

[46] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 4

[47] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li, and Dahua Lin. Generative adversarial frontal view to bird view synthesis. In *2018 International Conference on 3D Vision (3DV)*, pages 454–463. IEEE, 2018. 2

[48] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017. 2

[49] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proc. CVPR*, 2018. 2