Cross-view Semantic Segmentation for Sensing Surroundings

Bowen Pan[1,2], Jiankai Sun[2,3], Alex Andonian[1], Aude Oliva[1], Bolei Zhou[3]
[1]Massachusetts Institute of Technology, [2]Shanghai Jiao Tong University,
[3]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 and the free space from the scene. To help machine vision with spatial understanding capabilities, we introduce the View Parsing Network (VPN) for cross-view semantic segmentation. In this framework, the first-view observations are parsed into a top-down-view semantic map indicating precise object locations. VPN contains a view transformer module, designed to aggregate the first-view observations taken from multiple angles and modalities, in order to draw a bird-view semantic map. We evaluate the VPN framework for cross-view segmentation on two types of environments, indoors and driving-traffic scenes. Experimental results show that our model accurately predicts the top-down-view semantic mask of the visible objects from the first-view observations, as well as infer the location of contextually-relevant objects even if they are invisible[1].

1. Introduction

Recent progress in deep learning, including deep convolutional neural networks, has enabled machine vision to segment an image precisely into meaningful regions and objects [47, 25]. However, a deeper awareness of the surroundings requires better sensing of the spatial configuration, such as the coordinates and relative positions of the nearby objects, in a space [7, 2, 5, 6]. So far, we can mostly rely on 3D vision to extract this information. However, collecting 3D vision data is costly in time and resources, involving imaging devices such as LiDAR [13], 3D cameras [8, 40], or the extensive scanning of RGB-D cameras [11, 33, 32, 42, 34], as well as the need for a large amount of manual human annotations.

In this work, we explore a new image-based scene understanding task, the *Cross-View Semantic Segmentation*. As shown in Figure 1, the goal of the cross-view semantic segmentation task is to predict the top-down-view semantic map from a set of first-view observations. The resulting top-down-view semantic map naturally outlines the spatial layout of surrounding objects as well as their semantic meanings. Such a light spatial representation facilitates an agent to represent its surroundings in a more efficient and more exhaustive manner.

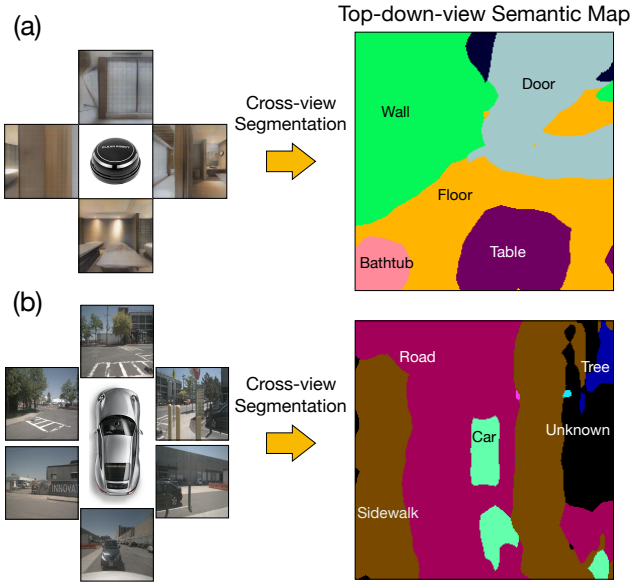

Figure 1: Top-down-view semantics are predicted from the first-view real-world observation in the cross-view semantic segmentation. We experiment our model in two common scenarios:(a) Indoor-room scene. (b) Driving-traffic scene.

One challenge in cross-view semantic segmentation is the difficulty of collecting the top-down-view semantic annotations. Recently, realistic simulation environments such as House3D [38] and Matterport3D [8, 41] 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. Although there is a domain gap between simulated scenes and real-world scenes, the generalization ability of the trained models across

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

domains can be improved using abstract representations such as depth map and semantic mask, or adopting domain adaptation techniques [37, 18].

This work explores a novel framework for cross-view semantic segmentation using simulation environments, and then transfers them to real-world environments. In the View Parsing Network (VPN), a view transformer module aggregates the information from multiple first-view observations such as the RGB images, depth maps, and semantic masks. It further outputs the top-down-view semantic map with the spatial layout of objects. We evaluate the proposed models on indoor scenes from House3D environment [38] and on outdoor driving scenes from CARLA environment [12]. Experiments show that we achieve 85.0% pixel accuracy and 41.0% Mean IoU on synthetic data of House3D as well as 84.7% pixel accuracy and 33.2% Mean IoU in CARLA environment. We also provide the domain adaptation results on Matterport3D dataset [8] in the Gibson environment [41] and nuScene dataset [2]. When transferred to real-world scenes, our VPN can still effectively represent the spatial configuration and localize the surrounding objects.

Our main contributions are as follows: (1) We introduce the cross-view semantic segmentation task with a View Parsing Network (VPN) to address it. (2) We narrow the gap between synthetic data and real data by transferring the VPN through abstract representation and domain adaptation techniques. (3) We demonstrate that the top-down-view maps provided by the VPN improve the surrounding exploration.

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 [25], SegNet [1] and PSPnet [46] are designed to segment the image pixel-wise within one-view. Image datasets with pixel-wise annotations such as CityScape [10] and ADE20K [47] 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 [50, 22, 43, 19, 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 [23, 48, 45, 30, 44, 20]. For example, [45, 30] describes a method to learn the transformation between ground and aerial imagery. [20, 44] synthesizes cross-views of objects and [48, 23] 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, 49, 41, 9, 21]). Several techniques have been proposed to address the domain adaptation issue [39, 3, 31, 36, 24, 26, 4], when models trained with simulated images are transferred to real scenes [18]. 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. The resulting top-down-view semantic map indicates the spatial layout and relations among objects, which is crucial for scene understanding and mobile robot navigation.

3.1. Problem Statement

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. 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 captured at different angles 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 segmentation and the visual SLAM [28, 29]. While both estimate the spatial configuration of an environment, our VPN aims at parsing the top-down-view semantic map rather than the full 3D reconstruction of the scene targeted by visual SLAM. Combining the semantic prediction with visual SLAM will be one possible extension of this work.

3.2. Framework of the View Parsing Network

Figure 2 illustrates the View Parsing Network. First-view observations are fed into the encoder to extract view feature maps. Each view feature map is transformed and aggregated

[2] https://www.nuscenes.org

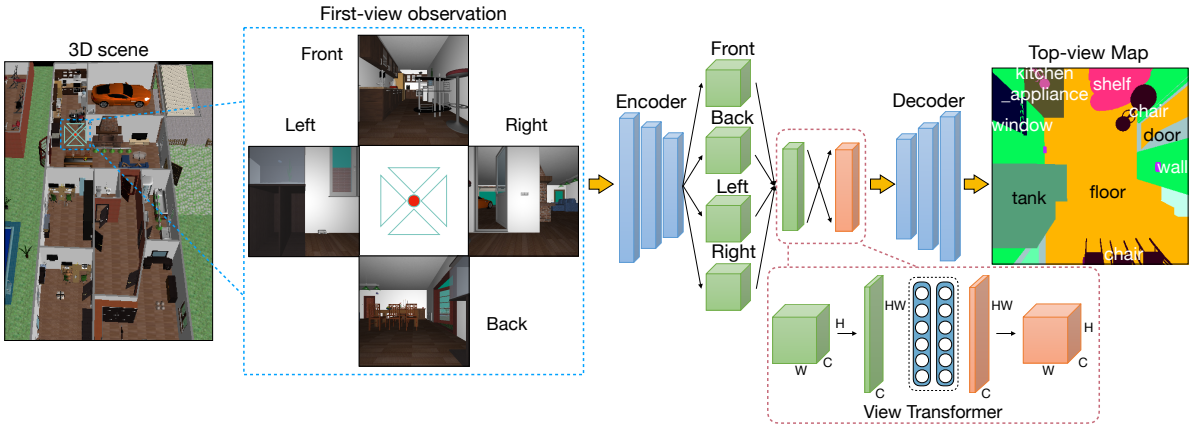

Figure 2: Framework of the View Parsing Network for cross-view semantic segmentation.

in the view transformer module, and then the aggregated feature map is decoded into a top-down-view semantic map.

Pipeline. As shown in Figure 2, from one spatial position of the floor map, we first sample N first-view observations (here $N = 4$ in Figure 2) in even angles so that all-around information is captured. Then, the first-view observations are encoded by a shared-weight encoder. This CNN-based encoder outputs a spatial feature map for each view. Then each feature map is fed into the View Transformer Module. After transforming these view feature maps from first-view space into the top-down-view feature space, we aggregate these feature maps into the top-down-view feature map. Finally, we decode it into a spatial map to predict the top-down-view semantic mask using a convolutional decoder.

View transformer. Although the encoder-decoder structure is a standard method in semantic segmentation networks [25, 46, 47, 1], our experiment shows that it performs poorly in the cross-view semantic segmentation task. While in the standard semantic segmentation the receptive field of the output spatial feature map is roughly aligned with the input spatial feature map, it is not the case for the cross-view semantic segmentation where the features are not spatially aligned. Ideally, the top-down-view map should take all input first-view feature maps into consideration, not just at a local receptive field region.

We designed the view transformer to learn the dependencies across all the spatial locations between the first-view feature map and the top-down-view feature map. This module will not change the shape of the feature map, so it can be plugged into any existing encoder-decoder type of network architecture for semantic segmentation. The diagram at the right corner of Figure 2 illustrates the transformation: The first-view feature map is first flattened while the channel dimension remains unchanged. Then we use the fully-connected layers to link the flattened first-view feature map and the flattened top-down-view feature map so that the receptive fields on the top-down-view feature map can cover the whole first-view feature map. Finally, the top-down-view feature map is reshaped back. Each view angle has its own view transformer module to transform the corresponding view observation. To aggregate the features from different view angles, we simply sum them up as

$$\mathcal{T} = \sum_{n=1}^{N} V_n(M^n), \tag{1}$$

$\mathcal{T}$ is the top-down-view output feature, N is the number of input first-view angles, M^n is the feature map of n^{th}-angle first-view observation and V_n is the corresponding view transformer submodule for each view angle.

Input observation. We train and evaluate our models with the input first-view observations 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 [47].

3.3. Sim-to-real Adaptation

To transfer the model trained in a simulation environment to a real-world environment, we use an adversarial training scheme to adapt it from synthetic source domain to the real-world target domain. The pipeline is mainly adopted from [37]. Here we have a view parsing network $\mathcal{G}$ to generate the top-down-view prediction P and use a discriminator $\mathcal{D}$ to discriminate if P is generated from source domain. The generator $\mathcal{G}$ is initialized by the weights of a well-trained VPN in synthetic environment. 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

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

Networks	RGB		Semantic Mask		Depth	
	Pixel Acc.	Mean IoU	Pixel Acc.	Mean IoU	Pixel Acc.	Mean IoU
1-view VPN	55.8%	6.5%	59.6%	13.2%	56.9%	7.6%
2-view VPN	70.1%	14.8%	75.7%	25.9%	70.2%	15.6%
4-view VPN	80.3%	27.7%	**85.0%**	40.6%	81.2%	27.6%
8-view VPN	**81.2%**	**28.5%**	84.7%	**41.0%**	**82.1%**	**29.9%**

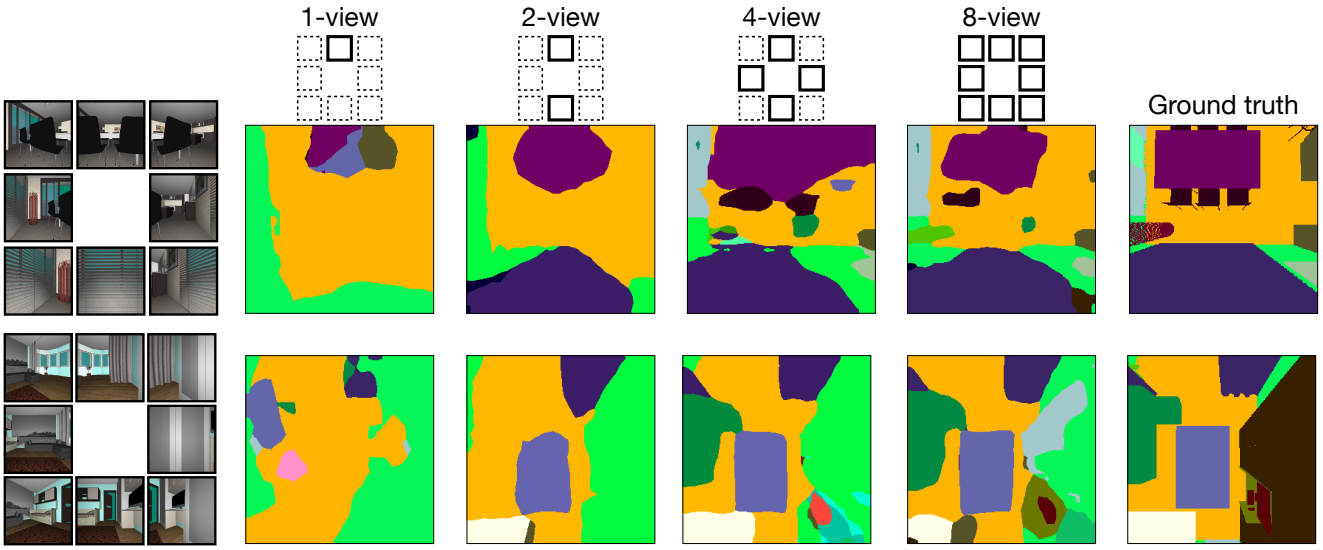

Figure 3: Cross-view segmentation result improves when the VPN receives more RGB views as input. The first column shows the input first-view RGB images at 8 evenly views. Other columns show the segmentation results by using 1-view, 2-view, 4-view and 8-view VPNs along with the ground truth respectively.

images from the target domain $\{I_t\}$ and use discriminator to distinguish whether F_t is from the source domain. 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 $\mathcal{D}$ to recognize which domain the feature output is from. The loss function to optimize $\mathcal{G}$ can be written as follows:

$$\mathcal{L}(\{I_s\},\{I_t\}) = \mathcal{L}_{seg}(\{I_s\}) + \lambda_{adv}\mathcal{L}_{adv}(\{I_t\}), \quad (2)$$

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 classification (source & target). Details can be referred to [37].

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 environment in Section 4.3. Finally, we demonstrate that our model trained on synthetic data can generalize to real-world data.

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 [46] as the decoder.

View transformer. We use a two-layers MLP as the baseline of the view transformer. Input and output dimensions of the view transformer 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. This simple design works well in practice, more sophisticated designs will be explored in future work.

Generator & Discriminator. For the generator, we use the architecture of the 4-view view parsing network. For the discriminator, we adopt the same architecture in [37]. It has 5 convolution layers, each of which is followed by a leaky ReLU [27] with the parameter 0.2 (except the last layer).

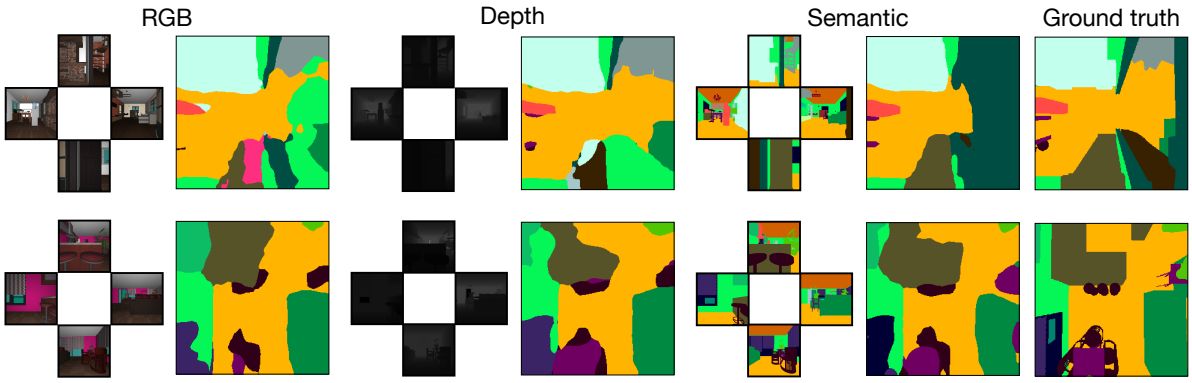

Figure 4: Cross-view segmentation on House3D using the input modality of RGB image, depth map and semantic mask respectively. Here the number of input views is fixed as 4. Ground truth is shown on the right.

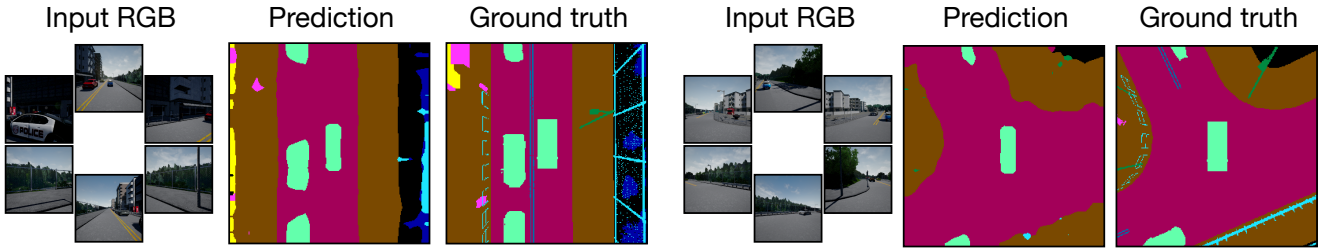

Figure 5: Qualitative cross-view segmentation results of a 6-view RGB VPN model trained on CARLA cross-view dataset. We train such a model in order to match the data modality and format of nuScenes dataset.

4.2. Benchmarks

Here we introduce **two synthetic datasets**, *House3D cross-view dataset* and *Carla cross-view dataset*, and **two real-world datasets**, *Matterport3D cross-view dataset* and *nuScenes dataset*. We train our VPNs on synthetic datasets and transfer them to the real-world datasets.

House3D cross-view dataset. We build a synthetic indoor-room dataset from House3D environment [38]. House3D is an interactive graphic environment built on top of the 3D indoor scenes of SUNCG dataset [35]. 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 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.

Matterport3D cross-view dataset. We also construct a real-world indoor-room dataset called *Matterport3D cross-view dataset*. Matterport3D is a large-scale RGB-D dataset with 10.8k real panoramic views from 194k RGB-D images of 90 real indoor scenes. We use the Gibson Environment [41] to render the panorama data and the top-down semantic masks. Gibson is a virtual environment which takes a sparse set of RGB-D panoramas in the input and renders a panorama from an arbitrary novel viewpoint. Additionally, it uses a technique called *Goggles* to eliminate the domain gap between the real scanned images and the rendered images. We choose 6 scenes of Matterport3D to extract the real-world cross-view data. The final benchmark dataset contains 4540 data pairs in total.

NuScenes dataset. NuScenes is a public large-scale dataset for autonomous driving. Each data sample contains first-view images from 6 directions (*Front, Front-right, Back-right, Back, Back-left, Front-left*). We select 17k data samples without top-down-view mask for unsupervised training.

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 RGB images captured from 6 directions.

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

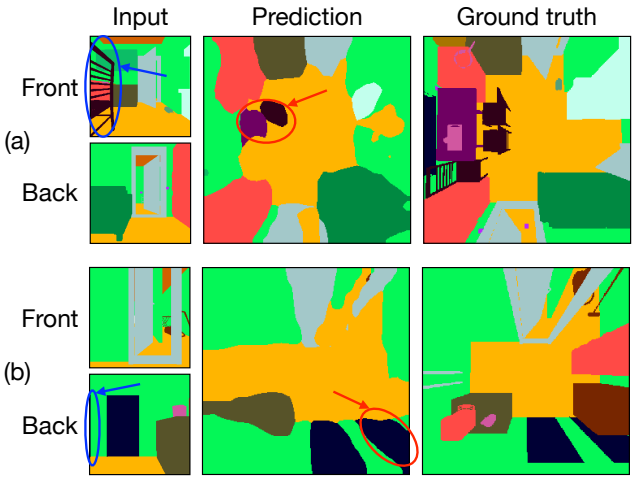

Figure 6: Predicting invisible objects: In (a), although the input semantic masks do not indicate that there is a table beside the chair, our model predicts it on the top-down-view mask based on the prior that tables and chairs usually appear together. In (b), in the Back view input mask, only a very small part of the second door can be observed, our model segments the door fully in the top-down-view prediction based on the shape prior of the door. Here the prediction is done by 2-view VPN with semantic input on the left.

Table 2: Ablation study of our VPN. We use **Pixel Accuracy** and **Mean IoU** to evaluate the results under two different view conditions.

Modality	Baseline		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%

segmentation using two commonly used metrics in semantic segmentation: **Pixel accuracy** which characterizes the proportion of correctly classified pixels, and **Mean IoU** which indicates the intersection-and-union between the predicted and ground truth pixels.

Results of VPNs. The quantitative segmentation results on our House3D cross-view dataset are listed in Table 1 with different modalities and different numbers of views. As the VPN receives more views, the segmentation results improve rapidly. Noticeably, taking semantic masks as input results to the best performance. We plot some qualitative segmentation results by our VPNs in Figure 3, Figure 4 and

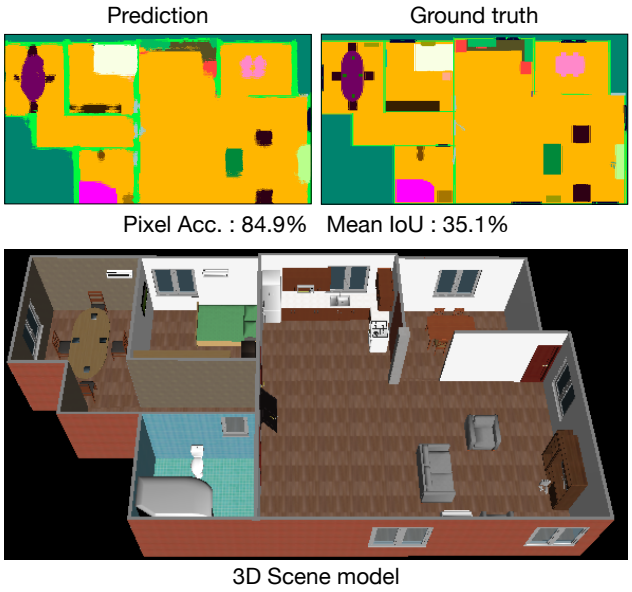

Figure 7: Integrating the local top-down-view semantic maps into a global semantic floor map for two scenes. The ground truth semantic floor map is generated from the ground truth top-down-view semantic maps. As a reference, the 3D scene model is also shown below.

Figure 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 detail. In Figure 4, we keep the number of input views fixed as 4, while varying the input modality. In Figure 5, we put some results of a 6-view RGB-input model trained in CARLA simulator, which achieves the performance of **84.7% Pixel Acc.** and **33.2% Mean IoU**. We further show interesting cases in Figure 6 where we find that our model has learned to infer some invisible objects.

Importance of View Transformer module. We further compare with the baseline networks in Table 2 to show the importance of the view transformer module in aggregating the information from multiple views. The baseline 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. We can see that our VPN 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.

Generating the semantic floor maps. Our VPN is able to integrate the local top-down-view maps into a semantic floor map. Since the ray spread into the top-down-view camera is not in parallel, we can not splice them directly. Thus we crop the central area of the predicted top-down-

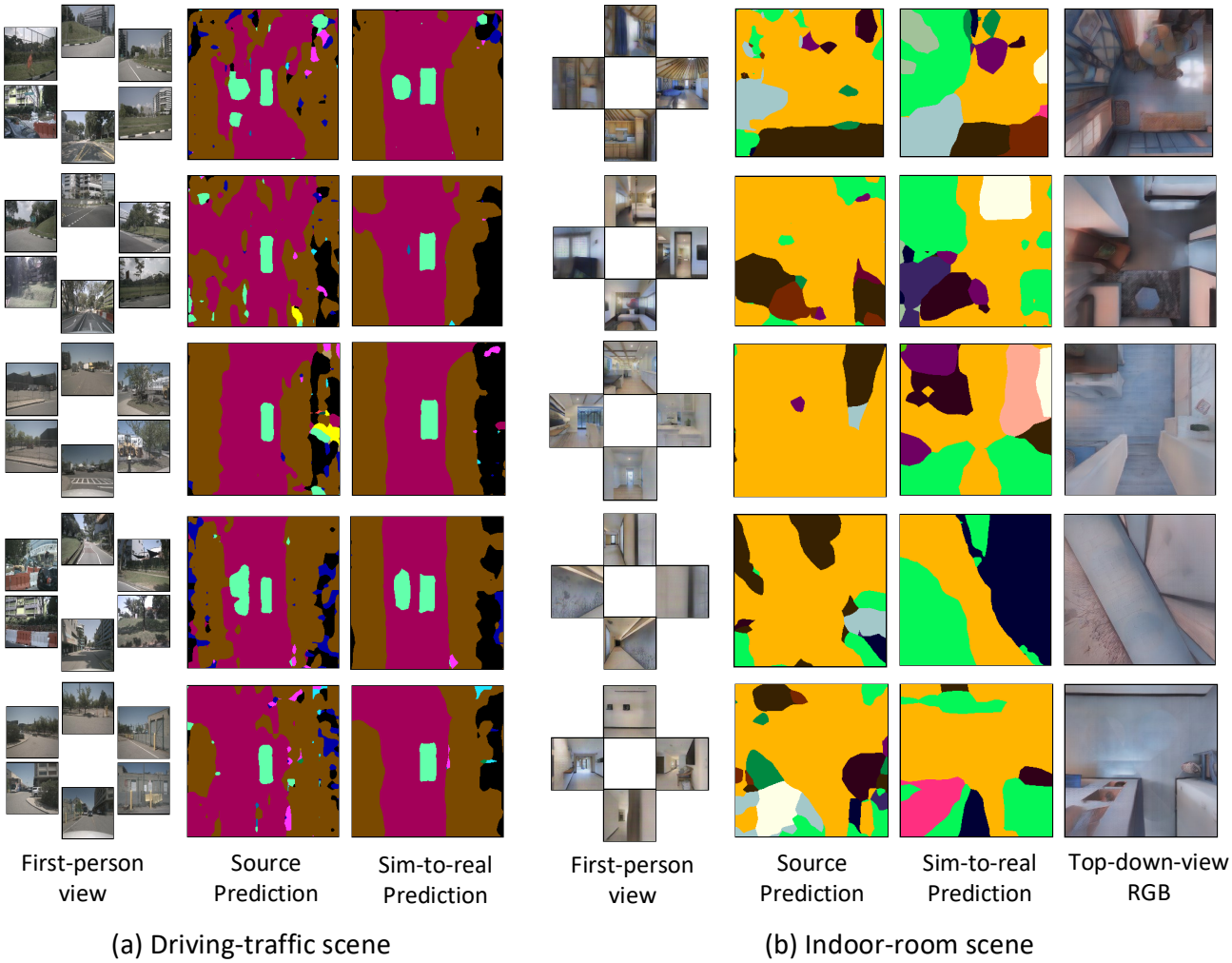

Figure 8: Qualitative results of sim-to-real adaptation. We can see that by exploiting domain adaptation techniques, the performance of our model in real-world data is significantly improved, comparing the sim-to-real prediction to the source prediction without domain adaptation. We can see that in driving-traffic scenes, our model is able to estimate the shape of road and roughly localize the surrounding vehicles. In indoor-room scenes, our model can effectively sense the spatial configuration of room, meanwhile, recognize the categories and locations of surrounding objects.

view map so that we can approximately assume that the ray emitted from this area is in parallel. We integrate the top-down-view maps in a max-pooling manner according to the predicted confidence map. Some qualitative results are shown in Figure 7. The integrated semantic floor map shows the spatial layout of all the objects in the environment.

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. In driving-traffic scenes, we first train a 6-view RGB VPN model in CARLA simulator and then transfer it to nuScenes dataset by using unsupervised domain adaptation techniques as depicted in Section 3.3. In indoor-room scenes, we train our model in House3D environment and transfer to Matterport3D dataset. Since the data distribution gap between the RGB images of House3D and Matterport3D is large, we first narrow this gap by exploiting some abstraction techniques. We first transform the RGB images of these two datasets into the semantic domain by utilizing the semantic mask, based on which we further deploy adversarial training (in Section 3.3) to adapt the output feature. Since there is no top-down-view ground truth in nuScenes dataset and the quality of semantic annotation of the top-down-view images in Matterport3D can not support to evaluate the model performance appropriately, we provide the qualitative results in Figure 8.

5. Application to Surrounding Exploration

To evaluate the effectiveness of the top-down-view mask generated from VPN for surrounding sensing, we test on a visual navigation task, *max-coverage surrounding exploration*. The task objective is to explore as much unknown

free space as possible within a certain number of steps in a new environment. We assume that the localization of the agent is given at each time step and each action is executed by the agent without noise.

5.1. Exploration with top-down-view map.

Humans exploring a space will head to space which they have not visited. This intuition reflects that exploration requires the agent to identify free space as well as remember which areas it has not visited yet. To achieve this goal, we make the agent able to identify the free space by training it to predict the top-down-view free-space map. Along with the state map which records the already-executed action sequence, the agent can remember the unvisited area.

Top-down-view free-space map. We train VPN to predict Top-down-view free-space map. Different from semantic map, free-space map has only two categories, obstacle and free space, which are denoted by 0, 1 respectively.

State map. Due to the ideal assumption made above, by memorizing the previous actions it has executed, the agent can easily build the state map which contains the information of the already-visited positions. We label the unvisited pixels as 0 and the already-visited pixels as 1 on the state map.

Exploration algorithm. We detail the navigation policy decision algorithm in Algorithm 1. At each time step t, we make the decision a_t and update the agent with the next top-down-view free-space map T_{t+1} and state map S_{t+1}. In both the top-down-view free-space map and the state map, we assume that the agent is always at the center of the map.

Algorithm 1 Exploration policy decision

Input: A top-down-view free-space map T_t and a state map S_t at time step t, where $T_t, S_t \in \{0,1\}^{L\times L}$.

Output: Policy action a_t, where $a_t \in$ {Forward, Back, Left-forward, Right-forward, Done}.

$U_t \leftarrow T_t \bigcap \neg S_t$; $a_t \leftarrow Done$; $d_s \leftarrow +\infty$
$D_t \leftarrow computeDistMap(U_t)$
for a in $\{Forward, Back, Left\text{-}forward, Right\text{-}forward\}$ **do**
    $d = execute(a)$
    if $d_s > d$ **then**
        $d_s \leftarrow d$; $a_t \leftarrow a$

$computeDistMap()$: Compute the shortest distance of each map pixel to the unvisited free-space region.
$execute()$: Return the shortest distance of the pixel to which the agent transit if execute the action a.

5.2. Result and comparison

We compare the following baselines for max-coverage exploration. **Random walk:** A random walk agent randomly chooses one action at each time step. **Top-down-view navigation with ground truth:** By planning on the ground truth top-down-view free-space map with Algorithm 1, we can obtain the upper-bound performance of our method. **Imitation learning without top-down-view:** A policy network learns to imitate the expert exploration trajectories given the first-view observations. Details of implementation can be referred to supplementary materials.

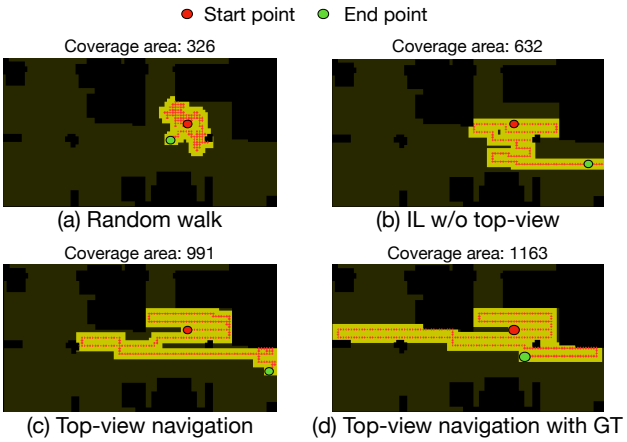

Figure 9: Examples of the max-coverage exploration. Each method has the same start point. The trajectory are shown and the explored area are lighted up.

Table 3: Comparison on max-coverage exploration.

Method	Coverage Area
Random walk	260.3 ± 82.7
IL w/o top-down-view	443.8 ± 340.6
Top-down-view navigation	673.8 ± 349.8
Top-down-view navigation with GT	1070.8±326.2

We run the algorithm directly on our predicted top-down-view map. For testing all the methods, we start the episode by initializing the state maps from zero, indicating that all free space is yet to be visited. *Coverage Area* is defined to measure the max-coverage exploration performance. We randomly choose 100 starting points on a scene map. For each starting point, we let the agent explore the space for 300 steps and compute the coverage area. Then final results are obtained by averaging the coverage area of these 100 episodes. Table 3 plots the exploration result for different methods and Figure 9 shows some sample trajectories. We can see that, equipped with the predicted top-down-view map from our VPN, the agent can act almost like an expert.

6. Conclusion

We proposed the View Parsing Network (VPN) for cross-view semantic segmentation. We transferred our model to real-scenes by abstract representation as well as domain adaptation. We showed that VPN can generalize to sense surroundings both in simulation and real-world environments.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. 2017. 2, 3

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

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. 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

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016. 2

[5] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 1

[6] E. Brachmann and C. 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

[7] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[8] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 2

[9] T. Chen, S. Gupta, and A. Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2019. 2

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. 2

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. CVPR*, volume 2, page 10, 2017. 1

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 2

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. CVPR*, 2012. 1

[14] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. *arXiv preprint arXiv:1702.03920*, 3, 2017. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 4

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

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

[18] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2

[19] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2017. 2

[20] D. Ji, J. Kwon, M. McFarland, and S. Savarese. Deep view morphing. In *Proc. CVPR*, volume 2, 2017. 2

[21] A. Kumar*, S. Gupta*, D. Fouhey, S. Levine, and J. Malik. Visual memory for robust path following. In *Advances in Neural Information Processing Systems*, 2018. 2

[22] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4865–4874, 2017. 2

[23] C.-C. Lin and M.-S. Wang. A vision based top-view transformation model for a vehicle parking assistant. *Sensors*, 12(4):4431–4446, 2012. 2

[24] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 2

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 1, 2, 3

[26] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 4

[28] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. 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. 2

[29] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. ICCV*, pages 2320–2327. IEEE, 2011. 2

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

[31] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. 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

[32] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608. IEEE, 2011. 1

[33] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[34] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 1

[35] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proc. CVPR*, 2017. 5

[36] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2

[37] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. 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

[38] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 1, 2, 5

[39] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[40] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[41] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1, 2, 5

[42] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision (ICCV)*, volume 00, pages 1625–1632, Dec. 2013. 1

[43] J. Xu, B. Stenger, T. Kerola, and T. 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

[44] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. 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

[45] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proc. CVPR*, volume 3, 2017. 2

[46] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. CVPR*, 2017. 2, 3, 4

[47] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. 2017. 1, 2, 3

[48] X. Zhu, Z. Yin, J. Shi, H. Li, and D. Lin. Generative adversarial frontal view to bird view synthesis. In *2018 International Conference on 3D Vision (3DV)*, pages 454–463. IEEE, 2018. 2

[49] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. 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

[50] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proc. CVPR*, 2018. 2