Learning Virtual View Selection for 3D Scene Semantic Segmentation

Tai-Jiang Mu, Ming-Yuan Shen, Yu-Kun Lai, *Member, IEEE*, and Shi-Min Hu, *Fellow, IEEE*

***Abstract*— 2D-3D joint learning is essential and effective for fundamental 3D vision tasks, such as 3D semantic segmentation, due to the complementary information these two visual modalities contain. Most current 3D scene semantic segmentation methods process 2D images "as they are", i.e., only real captured 2D images are used. However, such captured 2D images may be redundant, with abundant occlusion and/or limited field of view (FoV), leading to poor performance for the current methods involving 2D inputs. In this paper, we propose a general learning framework for joint 2D-3D scene understanding by selecting informative *virtual* 2D views of the underlying 3D scene. We then feed both the 3D geometry and the generated virtual 2D views into any joint 2D-3D-input or pure 3D-input based deep neural models for improving 3D scene understanding. Specifically, we generate virtual 2D views based on an information score map learned from the current 3D scene semantic segmentation results. To achieve this, we formalize the learning of the information score map as a deep reinforcement learning process, which rewards good predictions using a deep neural network. To obtain a compact set of virtual 2D views that jointly cover informative surfaces of the 3D scene as much as possible, we further propose an efficient greedy virtual view coverage strategy in the normal-sensitive 6D space, including 3-dimensional point coordinates and 3-dimensional normal. We have validated our proposed framework for various joint 2D-3D-input or pure 3D-input based deep neural models on two real-world 3D scene datasets, i.e., ScanNet v2 and S3DIS, and the results demonstrate that our method obtains a consistent gain over baseline models and achieves new top accuracy for joint 2D and 3D scene semantic segmentation. Code is available at https://github.com/smy-THU/VirtualViewSelection.**

***Index Terms*— Virtual view selection, 2D-3D joint learning, deep reinforcement learning, 3D semantic segmentation.**

I. Introduction

3D SCENE semantic segmentation is fundamental and essential for many applications, including autonomous driving, robotics, and beyond. Current designs of intelligent cars or robots [3] usually equip complementary and heterogeneous sensors to guarantee redundant safety. 2D cameras and 3D range sensors are the two most common kinds of visual sensors in such systems, which are complementary for depicting the underlying scene, i.e., 2D cameras capture the appearance of the scene, and 3D range sensors obtain the 3D surface representation of the scene.

Although 3D scene semantic segmentation can be achieved using pure 2D or 3D information, fusing 2D and 3D information is clearly beneficial to both 2D and 3D understanding [4], since features that are inseparable in a low dimensional space may be separable when mapped into a high dimensional space or in other dimensions. For example, objects similar in color and spatially close in the image domain may be far away to each other in depth; a painting is close to the wall it hangs on, but we easily separate it from the wall according to different colors.

The most intuitive way to incorporate 2D information into 3D scene understanding is to predict the semantics for 2D images from multiple perspectives using deep convolutional neural networks, and then re-project them back to the corresponding surface of the 3D scene via some aggregation strategies, according to the camera intrinsic parameters and the pose of each image. For these methods, the 2D view selection plays a key role in the final performance of 3D scene understanding. For the sake of time efficiency, choosing views that perfectly cover the underlying 3D scene [5], clustering similar views [4] or just uniformly sampling views from the captured sequence of 2D images [6] is usually adopted to replace the strategy of dense prediction on every 2D image. However, one main drawback for using such originally captured 2D images is that, due to the lack of freedom for capture paths and the limitation of device, the captured 2D images may be redundant, full of occlusion and/or with narrow/limited field of view (FoV), leading to poor performance for current deep neural networks.

Recently, Kundu et al. [5] proposed to use virtual views, which are rendered from the reconstructed 3D scene, to replace the originally captured 2D images. The advantages of using virtual views are two-fold: firstly, the FoV can be enlarged to cover more positional relationship information between objects; secondly, the virtual views can be controlled to avoid occlusion. However, the method has some limitations. On the one hand, the virtual views are directly aggregated for 3D segmentation without using 3D geometric information; on the other hand, the virtual view selection is determined without

Manuscript received 19 January 2023; revised 4 November 2023 and 26 February 2024; accepted 14 June 2024. Date of publication 10 July 2024; date of current version 15 July 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2021ZD0112902, in part by the National Natural Science Foundation of China under Grant 62220106003 and Grant 61902210, and in part by the Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. The associate editor coordinating the review of this article and approving it for publication was Prof. Lucio Marcenaro. *(Corresponding author: Shi-Min Hu.)*

Tai-Jiang Mu, Ming-Yuan Shen, and Shi-Min Hu are with the Key Laboratory of Pervasive Computing, Ministry of Education, and the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: taijiang@tsinghua.edu.cn; shenmingyuan99@gmail.com; shimin@tsinghua.edu.cn).

Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, CF24 4AG Cardiff, U.K. (e-mail: LaiY4@cardiff.ac.uk).

Digital Object Identifier 10.1109/TIP.2024.3421952

clear and quantitative assessment about the quality of selection, so may result in views that are redundant, uninformative or even having negative effects.

To overcome these limitations and to further improve the accuracy of 3D scene understanding, this paper proposes a general learning framework for joint 2D-3D scene understanding by selecting representative virtual 2D views of the underlying 3D scene. The generated virtual views, together with the 3D geometry, can be fed into any joint 2D-3D-input or pure 3D-input based deep neural models for 3D scene semantic segmentation to make the final prediction. However, directly predicting all the possible virtual views in the entire 6DoF (Degree of Freedom) space, i.e., a 3-DoF camera location and a 3-DoF camera orientation, is prohibitively expensive. We instead seek to predict regions that are likely to boost the 3D scene understanding, which will be the focus for detailed virtual view examination. Specifically, given a 3D scene understanding task, we generate virtual 2D views based on a learned information score map, which is estimated based on the prediction result of the task using a neural network called score net.

The training of score net is formalized as a deep reinforcement learning process, which rewards good predictions of the task using a convolutional scoring network. To render virtual 2D views that can jointly cover as many informative surfaces of the 3D scene as possible with a limited number of views, we further propose an efficient greedy virtual view coverage strategy in the normal-sensitive 6D space, i.e., 3-dimensional point coordinates and 3-dimensional normal. We have conducted experiments on two real-world 3D scene datasets [1], [2] with various base models to demonstrate the versatility of our method. The results show that our method obtains consistent gains over baseline models and achieves new top accuracy for joint 2D and 3D scene semantic segmentation.

In summary, this paper makes the following contributions:

- We propose to exploit the information score of prediction for improving 3D scene semantic segmentation and devise a general deep reinforcement learning framework to learn to predict the information score map effectively, which is applicable to any joint 2D-3D-input or pure 3D-input based deep neural networks.
- With the learned information score map of current prediction, we propose a greedy virtual 2D view generation method, which can render views that jointly cover as many highly informative regions of the 3D scene as possible, improving the final prediction results.
- We apply our framework to various joint 2D-3D-input or pure 3D-input based deep neural networks, and the results on both ScanNet v2 [1] and S3DIS [2] show that our framework can consistently boost the prediction accuracy of base models, achieving state-of-the-art performance for joint 2D and 3D scene semantic segmentation.

II. Related Work

The techniques for 3D scene understanding have seen significant evolution due to the demands in many real-world applications such as robotics and autonomous driving, as well as the release of public 3D scene datasets, such as ScanNet [1]. As it is a broad topic, we only review the most related work in the following.

A. Joint 2D and 3D Scene Understanding

Techniques for understanding of 2D image content progress rapidly and we have witnessed a lot of renowned models, such as DeconvNet [7], ResNet [8], FCN [9], Mask-RCNN [10], SSMA [11], DeepLab [12], Res2Net [13], PVT2 [14], etc. A straightforward integration of them into 3D is to map the semantics learned from 2D to 3D geometry along with a dense 3D reconstruction process, e.g., SemanticFusion [15], Semantic Reconstruction [16], PanopticFusion [17], MaskFusion [18], ProgressiveFusion [19] and 2D3DNet [20]. 2D views are also rendered from 3D representations to train 3D foundation models [21], [22]. Inspired by PanoContext [23], our method is most related to virtual multi-view fusion [5] which simply fuses the unary probability of pixels' semantic label from virtually rendered 2D views into 3D points. Concurrently, Rong et al. [24] select virtual views with active learning to refine the 3D semantic segmentation. However, these methods only learn the semantics from 2D views without making full use of the complementary 3D geometry, especially learning the geometric priors. Rong et al. [24] select the views based on a hand-crafted information score, which considers the cross-entropy of 2D semantic predictions, the cross-entropy of 3D semantic predictions fused from 2D views and the complexity of region (i.e., its point density). Our method exploits learning an information score map of current 3D semantic prediction to guide the virtual view selection, which is experimentally proved to be more effective than simply using cross-entropy of the prediction.

PointNet [25] and its variants [26], [27] have inspired the prosperity of performing semantic segmentation of 3D scenes directly on the 3D geometry with deep learning techniques. The key is to define proper convolution operations for different types of explicit representations, i.e., points [28], [29], [30], [31], [32], [33], [34], [35], voxels [36], [37], [38], [39], meshes [40], [41], [42], [43], and their combinations [44]. Special attention is drawn to adapt to the irregular and unordered properties of point and mesh data, by introducing transformer-like structure [45], [46], [47], [48], [49], [50]. Considering the sparsity of input high dimensional data, SparseConvNet [51] and MinkowskiNet [52] exploit sparse convolutions for efficient computation. Liu et al. [53] combine self-training with active learning for weakly supervised segmentation to reduce user annotations. Recently, effective deep neural networks [54], [55], [56], [57], [58] are also designed to handle large-scale scenes, such as 3D LiDAR point clouds in autonomous driving scenarios. In principle, these pure 3D-input based networks can be further improved with virtual 2D views using our framework, as demonstrated in our experiments.

Instead of directly fusing 2D semantic labels onto the 3D surface, recent joint 2D and 3D learning methods [59], [60], [61] usually first extract features from 2D views, and then integrate them with 3D features learned from the 3D geometry or simply feed them as the descriptor for deep geometric learning. We refer readers to [62], [63], [64], [65], and [66] for

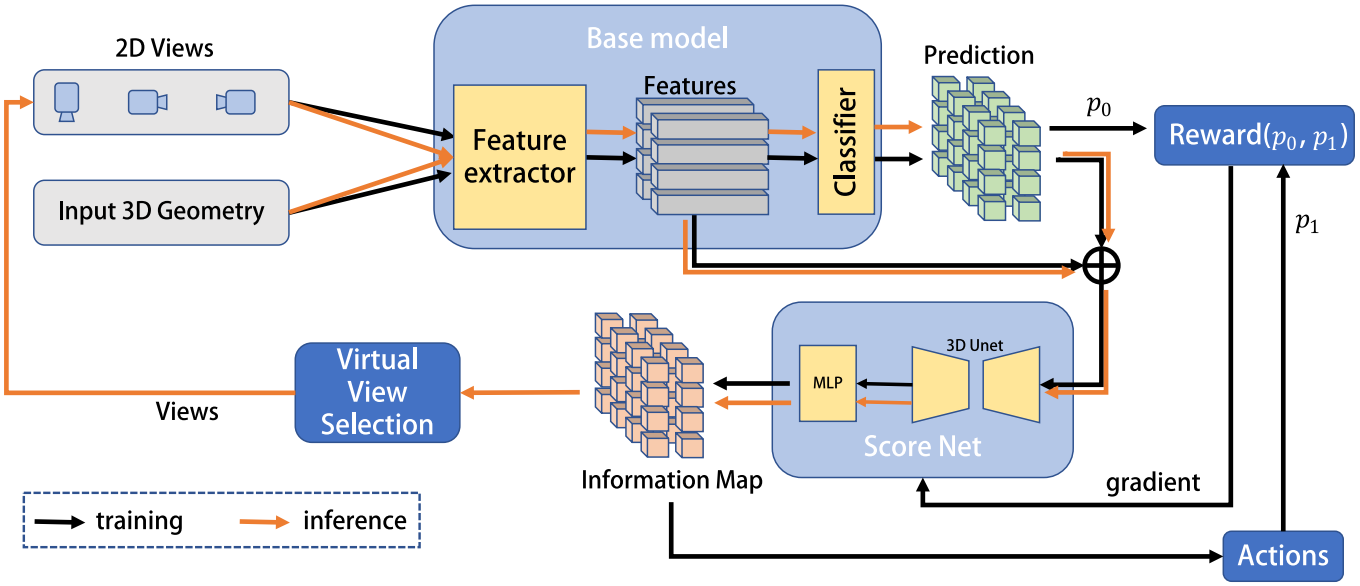


Fig. 1. The pipeline of our virtual view selection for 3D semantic segmentation. Our method takes both 3D geometry and initial 2D views as inputs and learns an information score map about the current prediction of the 3D scene using reinforcement learning. During the inference stage, we generate virtual views based on the information score map only once to refine the prediction.

comprehensive reviews on joint 2D and 3D learning for scene understanding. For the offline understanding of the whole 3D scene, MVPNet [6] feeds features extracted from multiple 2D views to PointNet [25]. Hu et al. [4] boost both the 2D and 3D semantic segmentations by explicitly fusing learned 2D features into 3D, and vice versa, using the bidirectional projection between 2D image pixels and 3D positions with the help of known camera intrinsic and extrinsic parameters. Robert et al. [67] propose a multi-view aggregation model to select the most relevant 2D features for joint 2D and 3D learning. LinkNet [68] also uses 2D-3D projection links and fuses the 2D and 3D features with an RNN (Recurrent Neural Network) module, making it suitable for online stable semantic segmentation. SVNet [69] is also designed for the purpose of online segmentation by defining convolution operations for on-surface 3D supervoxels and with the help of progressive voxel clustering. These methods perform 2D and 3D feature fusion with the originally captured images, which may be redundant, full of occlusion and/or with narrow/limited FoV, reducing the semantic segmentation performance of the 3D scene.

B. Deep Reinforcement Learning in Computer Vision

Reinforcement learning originated from the understanding of human decision-making process. The goal is to enable agents to determine their behaviors, i.e., taking actions according to the observed environment, called *states*. Different from traditional machine learning, reinforcement learning is supervised by the reward of choosing a specific action. Deep reinforcement learning is a combination of deep learning and reinforcement learning, which uses deep neural networks to learn the action space based on current observations.

Mnih et al. [70] proposed the first deep learning model supervised by a reinforcement learning method, namely a Deep Q-Network (DQN) combining a Convolutional Neural Network (CNN) with deep Q-learning [71], so that agents can achieve the performance equivalent to the human level in Atari games. From then on, deep reinforcement learning models have been applied in various vision tasks in recent years, such as face recognition in videos [72], video summarization [72], finding an object instance in videos [73], video action recognition [74], etc. These methods attempt to find the attentive regions or frames that are most informative to the given tasks, which are further formalized as a Markov decision process, and thus a deep reinforcement learning network can be adopted to learn the attention model to discard unwanted or misleading regions or frames while retaining the most important ones for the tasks.

Inspired by the above methods, we, for the first time, apply deep reinforcement learning to 3D scene semantic segmentation. In this paper, we regard the process of finding the 3D informative regions as a Markov decision process, and thus a deep neural network can be built to learn to predict such informative regions, which are further used to guide the virtual view selection to improve the performance of joint 2D and 3D scene understanding.

III. Virtual View Selection Using Reinforcement Learning

A. Overview

The general framework of our virtual view selection for joint 2D and 3D scene understanding is illustrated in Fig. 1. The main modules include the base model, the score network, the reward and the virtual view selection module.

Starting from the 3D geometry and an initial set of 2D image views of the scene as input, the base model, which can be any joint 2D and 3D scene semantic segmentation network, outputs an initial semantic segmentation prediction about the underlying 3D scene. Our framework is also applicable to

methods that only take 3D geometry information as input. In this case, the selected virtual views are treated as the input of an added 2D semantic segmentation network, which outputs the probabilities for each pixel to belong to different semantic labels, thus providing additional 2D predictions. The fused predictions from virtual 2D views and the base 3D model will be further used to calculate the reward. However, current joint 2D and 3D scene semantic segmentation methods usually directly adopt the original 2D views, which may be redundant, uninformative, or even have negative effects.

To select more informative 2D views to improve the current predictions, the score network is adopted to learn an information score map, indicating the quality of the current prediction of the base model. Actions, i.e., the selected virtual views, will be generated based on the information score map and fed back to the base model to produce new predictions. This score net is trained using reinforcement learning with the aim of maximizing the reward computed from the current and last predictions of the base model. During the inference, the 2D virtual view selection module renders virtual views of selected informative regions with poor prediction results implied by the learned score map. We then feed these virtual views into the base model for better prediction. This inference process is performed only once to save computational cost, as repeating this process does not give significant benefits in practice.

These core modules will be introduced in detail in the following subsections.

B. Score Network for Information Score Prediction

The base model can be any joint 2D and 3D neural network for 3D semantic segmentation, which takes the 3D geometry and the associated 2D views of the scene as input, such as BPNet [4] and MVPNet [6].

For those pure 3D methods such as VMNet [44], we introduce an extra 2D segmentation network and an extra 2D-3D fusion stage to get the final prediction. In this way, the whole method can still be viewed as a joint 2D and 3D learning process.

Given current 2D views and the 3D geometry, a typical base model for 3D semantic segmentation will first extract per-voxel features from the input and produce a probability of each voxel belonging to different classes with a classifier, e.g., in the form of a fully-connected layer followed by softmax activation. As our method is flexible with base models, for simplicity and fair evaluation, the base models we choose will be trained according to the settings reported in their original published papers, but the 2D feature extractors of the base models will be trained with rendered virtual views instead of the originally captured views. The parameters of the base models are fixed while training the score net.

Due to the limited FoV of ordinary cameras and use of a small set of 2D views during traditional 3D reconstruction, the original 2D views cannot perfectly cover the whole 3D scene from informative angles, leading to poor or even wrong predictions for uncovered regions. Although the input 3D geometry is complementary to 2D views, an informative selection of 2D views is still crucial for the final prediction, especially for those regions previously predicted poorly. Our method seeks to find the regions currently informative for the semantic prediction network. The philosophy of this idea is similar to the attention mechanism widely used in existing deep neural networks, but here the attentive regions are those with poor or even wrong predictions.

An intuitive way to find those poor predictions is to estimate their confidence by computing the entropy of the predicted probability of the semantic labels. Usually, one can assume the prediction is reliable when the confidence is high. However, there exist regions that are confidently predicted with wrong semantics, especially for those observed with few 2D views. Another way is to directly generate the information map using the cross-entropy between ground truth and predicted results; however, the ground truth labels are not available in real-world tests. We thus propose to learn a more indicative information score by encouraging correct and confident predictions, and penalizing wrong predictions.

We formulate this process as a reinforcement learning problem, as depicted in Fig. 1. Specifically, the agent, i.e., score net, learns by maximizing the total expected reward to generate the information score map for the current prediction, which is further used to select virtual views, updating the state and providing the reward. Next, we will give details about the core modules of the reinforcement learning for our task. We refer readers to [71] for a technical description about deep reinforcement learning in general.

State: The state here is the concatenation of the per-voxel features and the per-voxel semantic prediction of the 3D scene generated by the base model. These two components are complementary: the features contain the visual and geometric information about the 3D scene, and the predicted probability of the semantic segmentation provides an initial guess about the results. More specifically, we compute the entropy for each 3D voxel from its probability of semantic labels, instead of the probability itself, to provide an initial information score for the agent.

Agent: The agent makes decisions based on the reward. It is estimated via a 3D-UNet [75], followed by a Multi-Layer Perceptron (MLP). We choose a 3D-UNet architecture because we aim to predict the score for each nonempty voxel, and therefore the output should have the same spatial resolution as the input. 3D-UNet ensures details at the voxel level are better reserved, and is also compatible with sparse 3D convolution to better cope with the sparsity of the input (i.e., only available for voxels on the surface of the 3D scene). The 3D-UNet takes the state as input and produces a 64-dimensional feature f_i for each voxel i. To produce an information score in $[0, 1]$ for each nonempty voxel, we apply a shared MLP on each f_i. This information score also serves as the action selection probability P_i^a for each voxel i.

Action: We define two types of *actions* for indicating the selection of each voxel: "discarding" and "keeping". Unlike the action recognition of a video sequence, where the action can be performed on each frame, discarding or keeping one voxel is not informative enough to determine a virtual view. The regions matter. We thus perform the action selection on regions. We first cluster voxels into K regions with the k-means clustering algorithm in the normal-sensitive space,

i.e., a 6-dimensional space consisting of a 3-dimensional position q and a 3-dimensional normal n, to ensure that voxels in each cluster are sharing similar normals, so that we can render views that are frontally facing the surface of informative regions for better 2D feature extraction. Formally, the distance between voxel $v_1 = (q_1, v_1)$ and $v_2 = (q_2, v_2)$ is defined as $d(v_1, v_2) = ||q_1 - q_2||_2 + w_n \cdot ||n_1 - n_2||_2$, where w_n controls the weight of normal similarity. K is determined proportional to the area of the underlying scene A_{scene} with a ratio of $r_{cluster}$ as $K = \min(20, r_{cluster} \cdot A_{scene})$, ensuring at least 20 clusters. Then we compute the average information score $\hat{P}_j^a \in [0, 1]$ for each region j, representing the probability of choosing action "keeping" for the region. More specifically, we keep or discard a region by Bernoulli sampling:

$$a_j \sim \text{Bernoulli}(\hat{P}_j^a) \tag{1}$$

where $a_j = 1$ (with probability $\hat{P}_j^a$) means that region j is kept, and otherwise region j is discarded. We then render virtual views for each kept region (details will be given later) to update the state and provide reward for the agent.

Reward: Once the action for each region is determined, we feed the new virtual views of the selected region into the base model and obtain a new prediction p_1 for the 3D scene. The reward is used to evaluate how good the selected actions are and guide the training of the agent. Denoting the initial prediction as p_0, we define the reward r for each voxel i as:

$$r(i) = p_1^c(i) - p_0^c(i) \tag{2}$$

where c is the ground truth label of voxel i and $p^c(i)$ is the probability of choosing label c for voxel i.

To encourage the action selections that turn the prediction from incorrect to correct and to penalize those selections that turn the prediction from correct to incorrect, we give a large reward or penalty, respectively, as:

$$R(i) = \begin{cases} A, & \text{turning incorrect to correct} \\ -A, & \text{turning correct to incorrect} \\ r(i), & \text{otherwise} \end{cases} \tag{3}$$

The final reward R is the sum of the rewards of all voxels:

$$R = \sum_i R(i) \tag{4}$$

Training: Since we have K regions for a 3D scene, the size of the action set is exponential, making it too computationally expensive for DQN [71]. Inspired by [74], we train the agent with the policy gradient method. Similar to [74], we ran the agent on the same 3D scene for $N = 10$ episodes to approximate the gradient of expected reward by taking the average among episodes, and normalize the reward for easier convergence by subtracting the reward with a fixed baseline b, which was simply computed as the moving average of the rewards expected so far for the 3D scene. We refer readers to [74] for the details of training the agent.

C. Virtual View Generation

With the predicted information score map available for current 3D geometry and input 2D views, we can now generate virtual 2D views to improve the final prediction at the inference stage. Our goal is to render as few 2D views as possible that can cover as many regions of high information as possible. This is the classical camera placement problem, usually formulated as the Set Covering Problem (SCP), which is an NP-complete problem.

Algorithm 1 Virtual View Generation at Inference Stage

input: 3D scene in voxels $\{v = (q, n)\}$, information score map $\{\sigma\}$, the number of clusters K, covering threshold $\tau \in [0, 1]$, information score threshold τ_0, virtual views per cluster n_{views}
output: virtual 2D views covering informative regions
Cluster the voxels $\{v = (q, n)\}$ into K regions $\{c_i, i = 1, 2, \ldots, K\}$;
Denote the total number of voxels whose information score is larger than τ_0 in the K clusters as N;
Count the number n_i of voxels whose information score is larger than τ_0 for each cluster c_i and calculate their total information score s_i;
Sort clusters in descending order $\{i_1, i_2, \cdots, i_K\}$ by information score s_i;
TMP = 0;
for k = 1 to K **do**
  TMP = TMP + n_{i_k}
  if TMP $> N \times \tau$ **then**
    break
  end if
end for
for j = 1 to k **do**
  Calculate the virtual camera parameters for cluster c_{i_j};
  Render n_{views} virtual 2D images $\{I_j\}$ with virtual camera parameters;
end for
return $\{\{I_j\}, j = 1, 2, \ldots, k\}$

For training, we keep or discard each region by Bernoulli sampling described in the previous section to get the reward, train our score net, and generate virtual views for all the kept regions. At the inference stage, we only select as few regions as possible to cover as many regions of high information as possible. This SCP problem can be approximately solved by a greedy method. Specifically, we first sort the clusters of voxels in a descending order according to the total information score of voxels in the cluster. Then we find the minimal k, such that the regions covered by the top k clusters surpass a given covering threshold τ. Finally, we render a virtual view for each of the top k clusters by setting the fixation direction as the opposite of the average normal of the region, with an FoV just covering the whole region. In practice, due to the noise in real-world scans and the reconstruction error on the 3D scene, instead of using the opposite normal view, we render n_{views} virtual views for each region with the fixation directions evenly surrounding the average normal of the region and at an angle of 30° to the normal. We also perform the virtual view generation only for voxels whose information score is larger than a threshold τ_0, for the consideration of efficiency and focusing on informative regions. The above process of generating virtual 2D views is also listed in Algorithm 1.

Fig. 2(a) demonstrates several virtual views rendered for selected locations indicated by the information score map. The segmentation results in both the red box and cyan box of Fig. 2(b) show that after performing the prediction

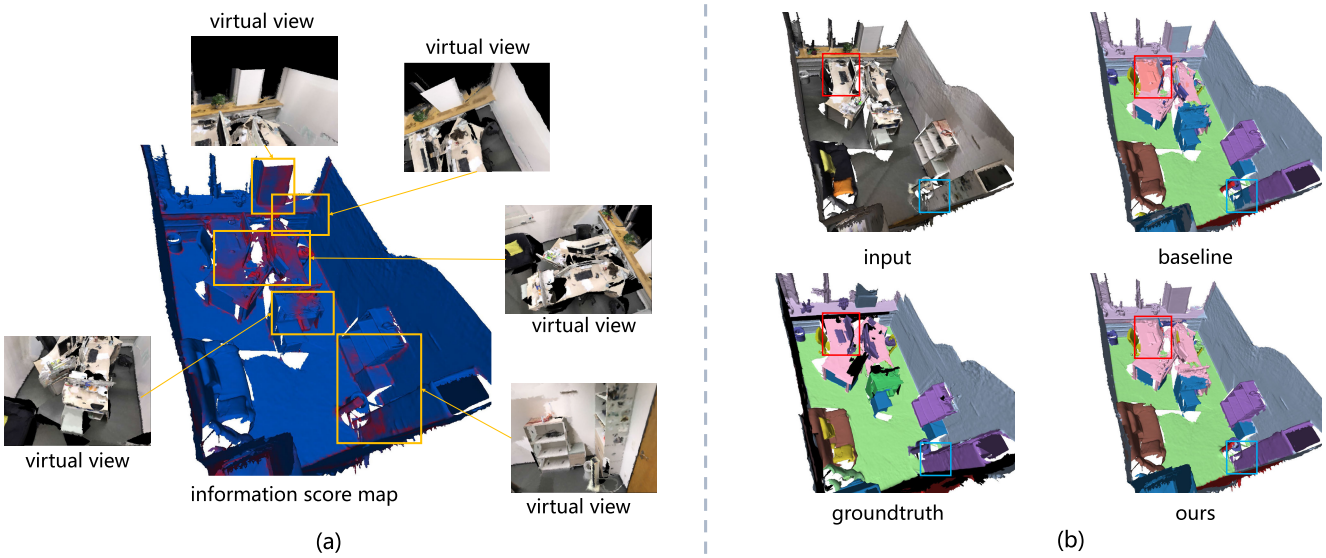


Fig. 2. (a) Virtual views rendered for highly informative regions in the scene, indicated by the yellow boxes. (b) the input 3D geometry, the groundtruth semantic segmentation, the prediction without virtual views (baseline), and the prediction with selected virtual views (ours). Note that we only present parts of the virtual views for clarity. Regions where significant improvements are achieved with our method are highlighted using colored boxes.

with selected virtual views, the segmentation performance of regions with high information scores is improved.

IV. Experiments

In this section, we first demonstrate that our method can serve as a general framework for joint 2D and 3D scene understanding (and for pure 3D methods by introducing a 2D semantic segmentation branch and 2D-3D fusion) by evaluating it with different base models and datasets. Then, we will justify the parameters of our method with ablation studies.

A. Implementation Details

1) Datasets: We train and evaluate our method on two public real-world datasets, i.e., ScanNet v2 [1] and the Stanford 3D Indoor Scene Dataset (S3DIS) dataset [2].

ScanNet v2 is built with manually recorded RGB-D video sequences and uses BundleFusion [76] to reconstruct 3D scenes with both voxel and triangular mesh representations. ScanNet v2 contains both 2D and 3D data, including about 2.5 million RGB-D images and 1,513 reconstructed 3D indoor scenes. The camera poses and camera intrinsic parameters of all RGB-D images are also annotated. The 1,513 scenarios are divided into 1,045 as the training set, 156 as the validation set, and 312 as the hidden test set.

S3DIS contains 6 reconstructed large-scale indoor areas from 271 different scans. The reconstructed 3D textured meshes and colored point clouds are generated based on RGB-D images collected by a Matterport camera. The point clouds are annotated with 13 semantic labels. We follow previous works which use Fold-1 split with Area5 as the test set.

2) Metric: We use the mean of class-wise intersection over union (mIoU) to evaluate the accuracy of 3D semantic segmentation, which is computed as the mean of 20 class IoUs for the ScanNet v2 dataset and the mean of 13 class IoUs for S3DIS dataset.

3) Other Details: The virtual 2D views are rendered using Python's Open3D module [77] with the scene's mesh model provided by the datasets. Since our method performs 3D semantic segmentation using virtual views, our 2D-3D joint base models also need to be trained with rendered virtual views. So, we sample one frame from every 20 frames and render virtual views with the same camera parameters as the original frames to re-train the base model (It takes 104 minutes and 26 minutes to render these virtual views for ScanNet v2 and S3DIS, respectively). This ensures that the base model is consistent with our method where virtual views are used. As we can later see in Table I, our re-trained base model performs almost the same as the original one trained with real views, showing that virtual views also work well for semantic segmentation.

The MLP for information score prediction has one hidden layer with the size of 16, and the size of the output layer is set to 1, ending with a sigmoid activation function. A is set to 10 to give a large reward or penalty. During the training of the score net, points in each scene need to be clustered into K groups. To achieve this, we first randomly down-sample the point cloud of each scan to 10,000 points for efficiency, and then cluster the down-sampled points into K groups using the k-means algorithm based on the Euclidean distance in the 6-dimensional space combining the coordinates and normal vectors of the points. Since the clustering is independent of the training process, this is pre-computed only once for each scene. It takes 42 seconds and 57 seconds on average to down-sample and cluster a scene in ScanNet v2 and S3DIS, respectively.

The training, rendering, and inference of the method described in this paper were all carried out in a Ubuntu server environment equipped with 8 Titan RTX GPUs.

B. Case Study

In this section, we conduct both qualitative and quantitative experiments by combining our virtual view generation

TABLE I

THE QUANTITATIVE SEMANTIC SEGMENTATION ACCURACY ON THE SCANNET V2 VALIDATION SET. "PT" REFERS TO THE PRETRAINED CHECKPOINT ON REAL SCANS PROVIDED BY THE ORIGINAL PAPER; "W/ VV" REFERS TO THE RETRAINED MODEL USING VIRTUAL VIEWS RENDERED WITH THE SAME CAMERA PARAMETERS AS ORIGINAL REAL SCANS; * INDICATES THE REPORTED SCORES FROM ORIGINAL PAPERS. + INDICATES THE RETRAINED MODEL ON REAL SCANS USING THE CODE PROVIDED BY THE ORIGINAL PAPER

Method	bathtub	bed	bookself	cabinet	chair	counter	curtain	desk	door	floor	otherfur.	picture	refrig.	curtain	sink	sofa	table	toilet	wall	window	mIoU
MVPNet w/ VV	80.7	70.5	60.8	58.2	93.1	59.6	41.8	60.2	54.9	91.9	51.3	14.7	49.8	53.4	63.4	75.0	83.6	86.3	80.9	50.2	64.0
MVPNet^{+}	80.5	69.9	62.1	59.2	92.7	60.1	43.4	60.3	55.2	91.9	52.2	15.3	49.4	55.2	62.5	74.8	82.5	86.1	79.5	51.1	64.2
MVPNet* [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
BPNet w/ VV	83.9	79.0	77.6	66.2	90.1	64.4	50.3	61.4	56.9	93.3	55.2	21.7	54.0	62.9	67.6	80.8	74.4	88.3	82.6	56.4	68.4
BPNet PT	83.4	78.6	79.6	64.0	91.2	63.2	52.7	61.6	57.4	93.3	53.6	22.2	55.0	63.7	69.0	79.2	74.5	88.0	83.9	54.9	68.5
BPNet* [4]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70.6
VMNet [44]	85.2	81.9	81.2	63.6	91.7	65.3	75.1	63.6	64.5	95.9	54.8	36.8	64.5	70.4	66.5	87.0	73.1	90.9	89.0	65.1	73.3
Virtual MVFusion [5]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.4
Ours (MVPNet)	82.4	71.8	63.7	59.8	93.1	59.4	46.7	61.8	59.3	92.5	51.2	21.2	52.6	56.0	64.5	77.8	82.7	88.3	83.3	54.9	66.2
Ours (BPNet)	84.4	80.3	79.8	67.1	90.9	66.0	56.1	62.2	60.8	93.6	57.5	24.2	57.7	65.6	69.9	81.4	76.7	88.3	86.0	58.5	70.4
Ours (VMNet)	86.3	83.9	80.3	64.8	90.4	67.1	75.7	66.2	67.6	96.0	57.9	40.6	66.5	67.9	69.1	89.2	74.1	90.1	90.8	67.9	74.6

technique with different state-of-the-art joint 2D-3D learning models (such as BPNet [4], MVPNet [6], and DeepViewAgg [67]) or pure 3D-input-based models (such as VMNet [44] and StratifiedFormer [48]) as the base model, to demonstrate that our proposed method can serve as a general framework to improve the 3D scene understanding with better 2D view selection. All results are obtained by performing the inference stage only once.

1) BPNet: To train the score network for BPNet [4] on a real scanned dataset, i.e., ScanNet v2, we first train BPNet for 100 epochs with point clouds and virtual views from the training set of the ScanNet v2 dataset. The 2D encoder network is initialized using the weights of ResNet34 [8] pretrained on ImageNet [78], and the 3D part is trained from scratch. In the BPNet training process, we set the voxel size as 0.05m and use the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and an initial learning rate of 0.01. The learning rate decays according to the Polynomial Learning Rate Policy with a power of 0.9. The resulting base model is referred to as "BPNet w/ VV", where "VV" means Virtual Views. We then train the score network with the policy gradient method on the training split of ScanNet v2 for 80 epochs. We use the Adam optimizer with a learning rate of 0.002, β_1 of 0.9, β_2 of 0.999 and ϵ of 1e-8.

Table I shows the quantitative comparison results of the baseline method BPNet and our method on the validation split of the ScanNet v2 dataset. We also list the result of BPNet produced by running the pre-trained checkpoint model[1] on real scans provided by the author (referred to as "BPNet PT") and the one reported in their original paper (referred to as "BPNet*"). As we can see, "BPNet PT" and "BPNet w/ VV" perform very similarly, demonstrating that the rendered virtual views and real scans with the same camera parameters are almost identical for learning. Our full method improves the IoU of most classes compared with the baseline model "BPNet w/ VV", and the improvement is more significant for objects like curtains, pictures, walls and windows whose 3D shapes are not particularly obvious, but their 2D texture features are comparatively distinct. Overall, our framework improves the performance of the base model "BPNet w/ VV" by 2% mIoU. Fig. 3 shows more qualitative comparison results between our method and the base model, demonstrating that the virtual views selected by our method help to correct the semantics of regions, such as the tables and curtains, that are previously wrongly predicted.

2) MVPNet: Similarly, we first train MVPNet [6] on ScanNet v2 training split with pre-processed virtual views and then train the score network with the deep reinforcement learning method. We first train its 2D CNN part with virtual views (the resulting base model is referred to as "MVPNet w/ VV") for 80,000 iterations with a batch size of 64. We employ the SGD optimizer with a learning rate of 0.005 and weight decay of 1e-4. Then, the weights of the 2D feature extractor network are frozen, while the 3D part of MVPNet is trained for another 40,000 iterations using the Adam optimizer with a learning rate of 0.02 and other hyper-parameters the same as BPNet. The process of training the score network of MVPNet is identical to BPNet.

The quantitative evaluation of MVPNet baseline "MVPNet w/ VV" and ours is listed in Table I. We also list the reported result of MVPNet in their original paper (referred to as "MVPNet*") and the retrained result on real scans using the code[2] provided by the original paper (referred to as "MVPNet^{+}") here. Again, we can see that "MVPNet^{+}" and "BPNet w/ VV" perform very similarly, and our framework improves over the baseline model "MVPNet w/ VV" for almost all classes, with the mIoU increased by 2.2%.

More qualitative comparison results between our method and MVPNet base model are shown in Fig. 3, demonstrating that the virtual views selected by our method help to correct the semantics of regions that are previously wrongly predicted, such as the tables and curtains.

3) VMNet: VMNet [44] is a network architecture that operates on the voxel and mesh representations leveraging both Euclidean and geodesic information. It is a pure 3D-input-based method. To make it compatible with our framework, we introduce a 2D segmentation network CMX [79] and use the fusion of VMNet and CMX predictions as our final result.

[1]https://github.com/wbhu/BPNet

[2]https://github.com/maxjaritz/mvpnet

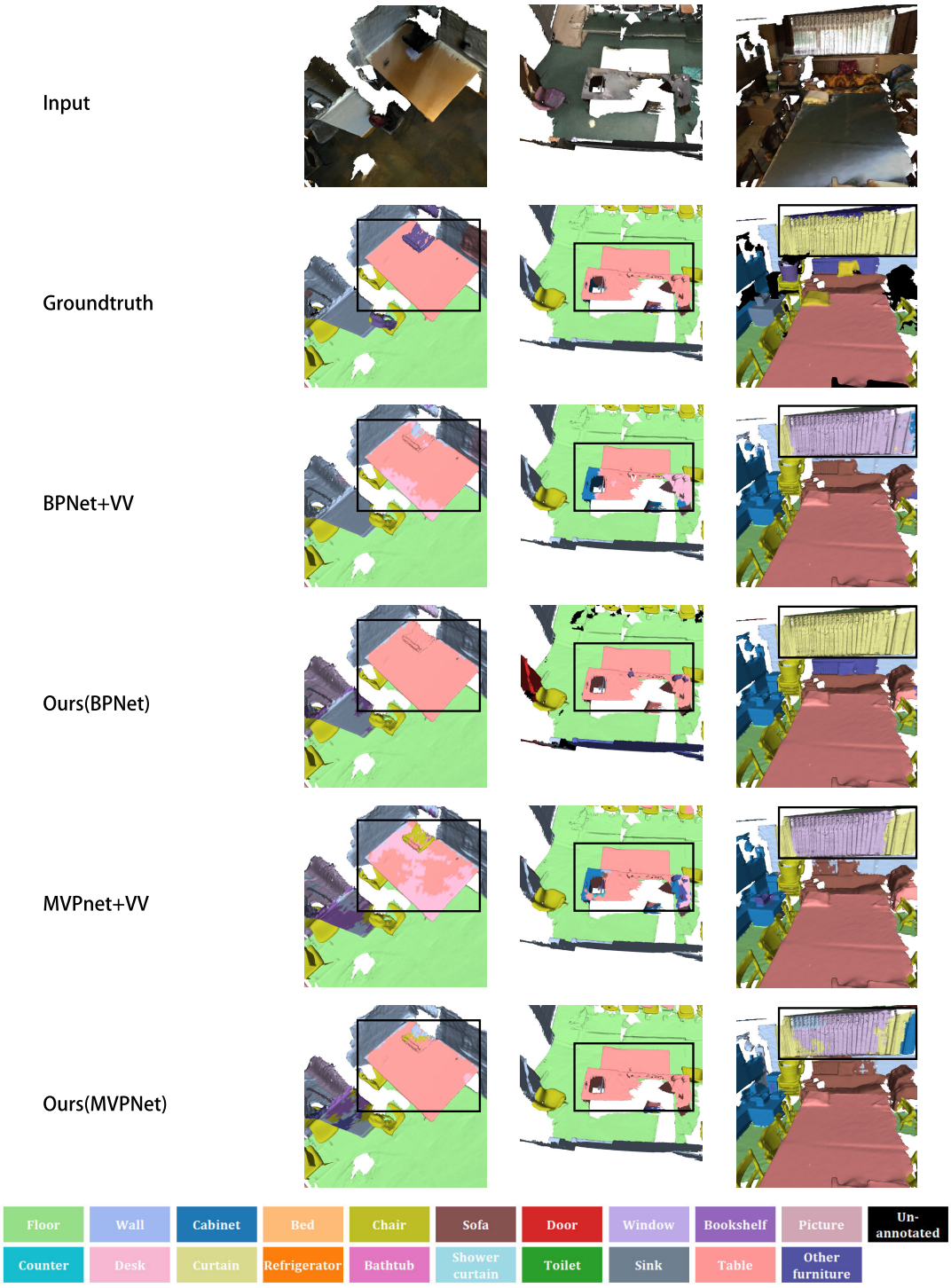


Fig. 3. More qualitative results of our method on the ScanNet v2 validation set. Our method improves semantic segmentation results of the base models ("+VV" means trained with virtual views). For each scene (column), please note the regions indicated by black boxes.

CMX is a vision-transformer-based cross-modal fusion method for RGB-X semantic segmentation. CMX uses an RGB image and another X modal image as input.

To train CMX on ScanNet v2, we render virtual views with the same camera pose as the original scan images. Then, we calculate the correspondence of pixels and voxels with the camera parameters. For each pixel, we record the corresponding xyz coordinates. The xyz-coordinates, together with the depth of each pixel, form the other modalities of the CMX network.

We train the 2D semantic segmentation network with rendered virtual views. We use MiT-B5 [80] as the backbone and use 1/20 of frames of rendered views as the training set. We train VMNet as the base model for 200 epochs with scenes of the ScanNet v2 dataset. In the training process, we keep the same settings as the VMNet paper. Input meshes are voxelized

TABLE II
THE QUANTITATIVE 3D SEMANTIC SEGMENTATION RESULTS ON SCANNET V2 HIDDEN TEST SET

Method	Publication	mIOU(%)
Mix3D [81]	3DV'21	78.1
OccuSeg [38]	CVPR'20	76.4
O-CNN [37]	SIGGRAPH'17	76.2
VMNet [44]	ICCV'21	74.6
LPRNet [39]	CVPR'23	74.2
MinkowskiNet [52]	CVPR'19	73.6
PointTransformerV2 [46]	NeurIPS'22	75.2
PointConvFormer [47]	CVPR'23	74.9
StratifiedFormer [48]	CVPR'22	74.7
KPConv [30]	ICCV'19	68.4
PointASNL [32]	CVPR'20	66.6
PointNet [25]	CVPR'17	55.7
Virtual MVFusion [5]	ECCV'20	74.6
BPNet [4]	CVPR'21	74.9
CSC [82]	CVPR'21	73.8
MVPNet [6]	ICCV'19	64.1
SVNet [69]	TOG'21	63.5
Ours(MVPNet)	/	66.4
Ours(BPNet)	/	75.7
Ours(VMNet)	/	76.1

at a resolution of 2 cm with data augmentation. The network is trained end-to-end by minimizing the cross entropy loss using Momentum SGD with Poly scheduler decaying from learning rate 0.1.

During the fusion of 2D prediction results and 3D results, each voxel can be projected to pixels of several virtual views. However, some of these views are not suitable for predicting the semantic label of the voxel, e.g., when the corresponding pixel is located at the edge of the view, or the view from that perspective is difficult to identify the classification of the object the voxel belongs to. So, for each voxel, we need to select some of the views that can better predict the label of its corresponding pixels.

Specifically, for each voxel, we calculate the cross entropy of all the 2D prediction label probabilities of its corresponding pixels and select the top 5 of those pixels (or views). After averaging the top 5 result probability vectors, we perform the fusion between the 2D averaged label probability vector and 3D semantic segmentation label probability vector by a 2-layer MLP.

Finally, we train the score net for baseline VMNet with the policy gradient method on the training split of ScanNet v2 for 100 epochs. We use the Adam optimizer with a learning rate of 0.002.

Table I shows the quantitative comparison results of the baseline method VMNet and our method on the validation split of the ScanNet v2 dataset. Our method again boosts the base model by 1.3% for mIoU.

4) Comparison With Other Methods: We also compare our method with other methods using different types of convolutions, including point-based convolutions [25], [30], [32], [46], [47], [48], pure 3D convolutions [37], [38], [39], [44], [52], [81], 2D convolutions [5] and joint 2D-3D convolutions [4], [6], [69], [82], on the ScanNet v2 hidden test set. Some results are directly drawn from [5] and [69], and listed in Table II. As we can see, our method improves the mIoU of the base model VMNet by 1.5%. The performance of our method based on VMNet exceeds most of other methods except Mix3D, a data augmentation method mixing scenes to train a base model. Our method is currently incompatible with Mix3D because of the low quality of virtual views rendered from 2 overlapped scenes. Our method achieves comparable performance as two other pure 3D methods, i.e., O-CNN [37] and OccuSeg [38] (the semantic segmentation version of the implementation is unavailable), which could be further improved with our virtual views. Although the results of BPNet and MVPNet base models re-trained with virtual views are below their reported scores in their original papers on the validation set of ScanNet v2, on the hidden test set of ScanNet v2, our framework improves BPNet and MVPNet by 0.8 and 2.3 percent mIoU, respectively.

Different from ours, Virtual MVFusion selects denser virtual views (typically 100∼200 views per scene) to fully cover the scene in a heuristic manner, while ours can still achieve more accurate segmentation results by only covering 60% of the highly informative voxels (with information score larger than 0.6). Moreover, our framework surpasses Virtual MVFusion by 1.5 percent mIoU on the test set, despite being dropped by 1.8 percent mIoU on the validation set, demonstrating that our virtual view selection based on learned information score can consistently improve joint 2D and 3D semantic segmentation.

5) Comparison on S3DIS Dataset: To further verify the effectiveness of our method, we tested our proposed framework on another real-world 3D scene dataset S3DIS [2]. Most of the hyper-parameters in Section IV-C for S3DIS are the same as ScanNet v2 dataset.

Similarly, we first generate virtual images for each scan according to the pose of RGB-D images. Then we train BPNet and MVPNet using the rendered virtual views and obtain the baseline models, referred to as "BPNet w/ VV" and "MVPNet w/ VV". For pure 3D-input method VMNet [44], we train the base model using the mesh and our version of VMNet with additional virtual views. The agent is trained on this dataset with deep reinforcement learning, so that at the inference stage, we can get the desired virtual views according to the information score map produced by the score network. We also applied our framework to new state-of-the-art models, including DeepViewAgg [67] and StratifiedFormer [48].

The quantitative comparison results on Area5 are listed in Table III, showing that our framework can also consistently boost (+2.7/+2.1/+1.5/+1.2/1.0 mIoU) the performance of the base models (MVPNet/BPNet/DeepViewAgg/VMNet/StratifiedFormer) for large-scale 3D scenes. We also compare our method with other state-of-the-art methods using point-based convolutions [25], [28], [32], [35], [46], 3D convolutions [39], [52], and 2D convolutions [5]. Our framework using StratifiedFormer [48] as the base model outperforms all other competitors and achieves new top accuracy for joint 2D and 3D scene semantic segmentation on S3DIS. We believe Retro-FPN [35] (a pure 3D-input-based model) can also benefit from our framework.[3]

[3]Retro-FPN [35] is not open-sourced yet by the submission time of our work.

TABLE III

THE QUANTITATIVE SEMANTIC SEGMENTATION ACCURACY ON THE S3DIS TEST SET (AREA 5). FOR EACH CLASS, THE IOU IS REPORTED AND THE NUMBERS IN **BOLDFACE** INDICATE THE BEST PERFORMANCE.CONV CATEGORY: (I) POINT-BASED CONVOLUTION, (II) 3D CONVOLUTION, (III) 2D CONVOLUTION, (IV) JOINT 2D AND 3D CONVOLUTION

Method	Conv.	mIoU
PointNet [25]	I	41.1
PointCNN [28]		57.3
PointASNL [32]		62.6
PointTransformerV2 [46]		71.6
StratifiedFormer [48]		72.0
Retro-FPN [35]		**73.0**
MinkowskiNet [52]	II	65.35
VMNet [44]		65.4
LPRNet [39]		69.1
Virtual MVFusion [5]	III	65.38
MVPNet [6] w/ VV	IV	58.2
BPNet [4] w/ VV		60.9
DeepViewAgg [67]		67.2
Ours (MVPNet)	IV	60.9
Ours (BPNet)		63.0
Ours (VMNet)		66.9
Ours (DeepViewAgg)		68.4
Ours (StratifiedFormer)		**73.0**

C. Ablation Studies

During inference, several parameters affect the performance of our method. The virtual view selection policy determines how and where to render the virtual views; the threshold τ_0 controls the total number of voxels to be covered by the virtual views; the covering threshold τ influences the actual amount of 2D virtual views to be integrated with 3D. The parameters of voxel clustering and the number of virtual views per region affect the generated virtual views. The number of inference cycles affects the accuracy and efficiency of the method. To justify the choice of these parameters, we perform ablation studies on the ScanNet v2 validation set with the base model set as VMNet [44].

1) The Effect of Virtual View Selection Policy: To verify the effectiveness of selecting virtual views according to the information scores predicted by our method, a comparative experiment is designed. We compare our selection policy with three other view selection policies. The first one is to randomly select regions to render virtual views. The second one is to select views according to the information entropy of the prediction vectors provided by the base model. The last one is to use real scans instead of virtual views. Specifically, for each selected region, we choose n_{views} real scans which cover the most part of the region instead of generating n_{views} virtual views. The quantitative comparison results are shown in Table IV, and the qualitative comparison between different virtual view selection policies is illustrated in Fig. 4. As we can see, the policy that randomly selects regions for virtual view rendering has a poor performance, and the segmentation performance of entropy policy is lower than the result of virtual view generation according to the information scores. Thus, the score network trained by reinforcement learning in this paper does play a significant role in improving the semantic segmentation performance. Using real scans is still inferior to our virtual views due to the incapability of covering many regions of high information caused by the limited FoV of real scans. Compared to the simple entropy-based baseline, the increase in training time and memory mainly comes from the score net. Taking BPNet as an example, the training time per epoch on ScanNet v2 for the baseline model is 1,713 seconds, and the extra training time for our score net is 954 seconds. The training memory is 9,236MB and 4,682MB, respectively. For MVPNet, the training time per epoch for the baseline model on ScanNet dataset is 1,472 seconds, and the extra training time for our score net is 897s. The training memory is 7,582MB and 3,894MB respectively. Since our work is not an online method, it is worthwhile to take extra training time to achieve a better segmentation performance, which is essential for further applications, such as robot grasping and manipulation.

TABLE IV

THE QUANTITATIVE ACCURACY COMPARISON BETWEEN DIFFERENT VIRTUAL VIEW SELECTION POLICIES ON THE SCANNET V2 VALIDATION SET

Policy	Random	Entropy	Real Scans	Ours
mIOU(VMNet)/%	71.89	74.27	73.75	74.62
mIOU(BPNet)/%	66.79	68.83	67.95	70.38

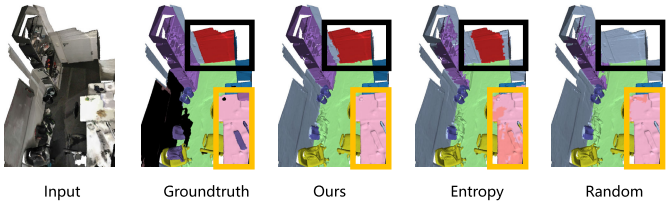


Fig. 4. The comparison of different virtual view selection policies. In this scene, the virtual views generated by our method help to correct the semantic segmentation on the door and the table. Note the regions indicated by the black and golden boxes.

2) The Effect of Information Score Threshold τ_0: The output of our agent (score net) serves as the action selection probability. During the inference stage, when generating virtual views, we only consider those having an information value larger than the given threshold τ_0. Fig. 5 shows the performance of different values of τ_0 in $[0.0, 1.0]$ with a step of 0.1. As we can see, when a small value is set for τ_0, the selected virtual views might not be precisely targeted at the voxels of high information, which will reduce the semantic segmentation performance. On the other hand, when the threshold is too high, the proportion of highly informative voxels covered by virtual views will be small, which also limits the potential of the method. Finally, we achieve a trade-off by selecting τ_0 as 0.6.

3) The Effect of the Covering Threshold τ: At the inference stage, we select regions with high total information scores, which contain sufficient numbers of voxels that surpass a given covering threshold τ, to generate virtual views. The total actual number of virtual views for each scan is thus determined by this parameter. Since the virtual views are rendered during the inference period and will be used as the input to the baseline model, each additional region will increase the prediction time and space consumed. Fig. 6 shows the relationship between the covering rate τ and the accuracy of segmentation prediction results. As we can see, the mIoU

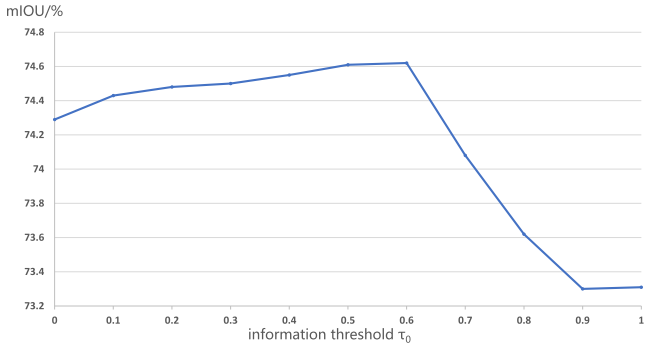


Fig. 5. The quantitative accuracy of our method on the ScanNet v2 validation set w.r.t. different τ_0 values.

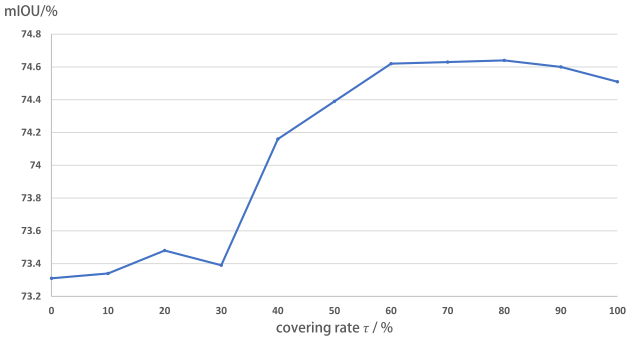


Fig. 6. The semantic segmentation mIOU w.r.t. the covering rate τ on ScanNet val set.

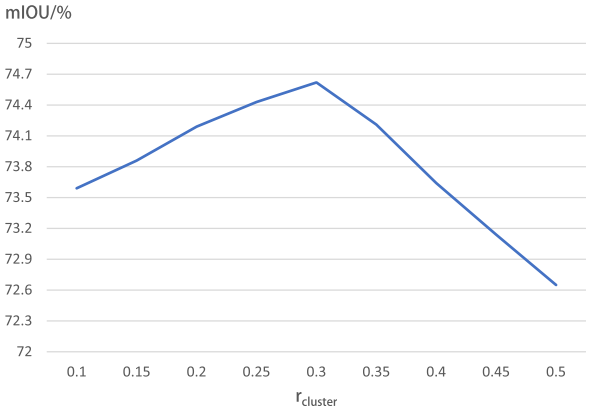


Fig. 7. The semantic segmentation accuracy w.r.t. $r_{cluster}$.

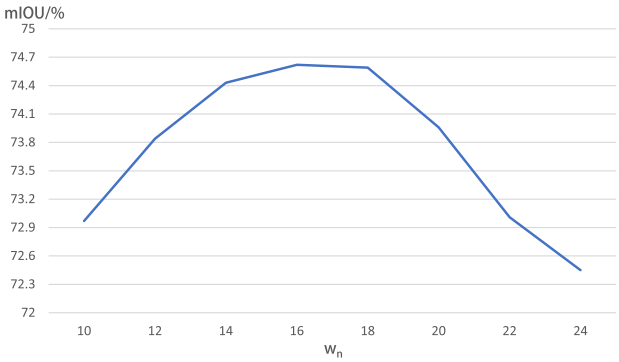


Fig. 8. The semantic segmentation accuracy w.r.t. the weight of normal similarity w_n.

of the prediction increases as the covering rate rises. However, when the rate τ surpasses 0.6, the rate of performance gain in the prediction becomes much slower. This is because a larger covering rate leads to a higher proportion of regions covered by virtual views, and when the covered regions are large enough, most inaccurate segmentation regions will be associated with enough 2D information. So, further increasing the number of virtual views, equivalently increasing the covering rate, will not bring new and useful information to the network, even introducing redundant and contradictory information, and causing performance degradation. Therefore, we set the covering rate as 0.6 to obtain the final 3D segmentation results, making a balance between accuracy and efficiency. As a result, our method usually selects 6 regions to render virtual views for most scenes.

4) The Effect of the Clustering Parameters: During the virtual view selection, we cluster the entire scene into K parts. The number of clusters is linearly proportional to the area of the scene A_{scene} (measured in m^2) with ratio $r_{cluster}$. Specifically, the number of clusters K equals $r_{cluster} \cdot A_{scene}$, and if it is worked out as less than 20, K is set to 20. The mIoUs of our method w.r.t. different values of $r_{cluster}$ are shown in Fig. 7. We set $r_{cluster} = 0.30m^{-2}$ to ensure clustered regions are sufficiently fine-grained but not overly fragmented to facilitate virtual view selection.

We perform the voxel clustering in the normal-sensitive space (a 3-dimensional position and a 3-dimensional normal), where the weight of the normal similarity is controlled by w_n. The position parameter is the xyz-coordinates of each voxel using meters as the unit, and the normal vector is normalized to a unit vector. Different w_n will affect the clustering quality and further influence the final semantic segmentation result, as shown in Fig. 8. As we can see, as w_n increases, the clustered region indeed has more consistent normals, helping generate good virtual views. However, when w_n gets too large, the clustered region would be cluttered voxels, making the generated virtual views less informative. To ensure each cluster covered by one virtual view, voxels in one cluster should be close and share similar normal directions. To achieve this, we choose a relatively large value for the weight of normal similarity as 16.

5) The Effect of the Number of Virtual Views: For each region, a proper number of virtual views should be generated to cover it. The virtual views need to provide enough information while avoiding redundancy. Different numbers of views for each region n_{views} can affect the final segmentation result, as shown in Fig. 9. As we can see, the performance increases evidently when n_{views} is below 4 and is almost constant when n_{views} is above 4. To balance the performance and efficiency, we choose 4 virtual views for each selected region, resulting in 24 views for most scenes. Although original base methods usually require 3-5 images, their performance would not increase when fed with more real scans. Besides, our method still requires fewer images than Virtual MVFusion [5].

6) The Number of Inference Cycles and Timings: Our inference process could be repeated by iteratively selecting virtual views to update the model. The extra computational cost of

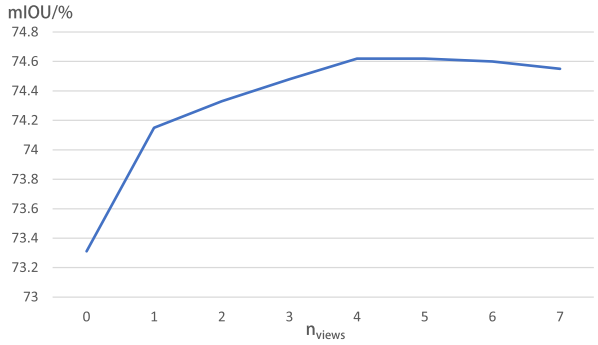


Fig. 9. The semantic segmentation accuracy w.r.t. n_{views}.

TABLE V
THE TIMINGS AND ACCURACY W.R.T. THE NUMBER OF INFERNCE CYCLES

Methods	Params/M	time per scene/s	mIOU/%
VMNet	569.4	1.7	73.31
Ours(VMNet)	807.5	2.9	74.62
Ours(VMNet) 2cycles	807.5	3.9	74.56
Ours(VMNet) 3cycles	807.5	4.9	74.53
Ours(VMNet) 4cycles	807.5	5.4	74.58

repeating the inference cycles consists of the running of the extra round of the base model (plus the 2D semantic network for pure 3D-input models), the score net, and the rendering of virtual views. We perform the inference stage with different numbers of cycles and list the number of network parameters, average inference time per scene, and the mIoU of semantic segmentation results in Table V. As we can see, the mIoU is almost unchanged. This is because most highly informative regions have been covered with virtual views after the first cycle of inference. To balance the performance and efficiency, we perform the inference stage only once.

V. CONCLUSION

In this paper, we introduce a general framework to select virtual views for 3D semantic segmentation. Compared to originally captured images, virtual views are free of FoV limitation and occlusion, which helps to associate more informative 2D features for 3D semantic segmentation. Deep reinforcement learning has been employed to train a score network to predict the information map of the scene, guiding the selection of virtual views with a greedy strategy. Comprehensive experiments on two real-world datasets, i.e., ScanNet v2 and S3DIS datasets show that our method can consistently boost the performance of different base models and achieves the best result in 3D semantic segmentation compared to other 2D-3D joint or pure 3D learning methods.

One limitation of our method is that it does not work well for sparse point clouds. In this case, the rendered virtual view would be filled with many holes, leading to poor 2D feature extraction. One possible solution is to complete the sparse point cloud before rendering virtual views.

We believe the region selection policy based on information score maps can be applied to the 3D domain to achieve better performance in scene understanding. In future, we plan to further introduce our module to another important scene understanding task, i.e., 3D instance segmentation.

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers for their helpful suggestions for improving this article.

REFERENCES

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[2] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2D–3D-semantic data for indoor scene understanding," 2017, *arXiv:1702.01105*.

[3] S. Yang, B. Li, M. Liu, Y.-K. Lai, L. Kobbelt, and S.-M. Hu, "HeteroFusion: Dense scene reconstruction integrating multi-sensors," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 11, pp. 3217–3230, Nov. 2020.

[4] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.-T. Wong, "Bidirectional projection network for cross dimension scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14373–14382.

[5] A. Kundu et al., "Virtual multi-view fusion for 3D semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, Aug. 2020, pp. 518–535.

[6] M. Jaritz, J. Gu, and H. Su, "Multi-view PointNet for 3D scene understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3995–4003.

[7] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. ICCV*, Dec. 2015, pp. 1520–1528.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[9] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[11] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1239–1285, May 2020.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[13] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[14] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.

[15] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4628–4635.

[16] J. Jeon, J. Jung, J. Kim, and S. Lee, "Semantic reconstruction: Reconstruction of semantically segmented 3D meshes via volumetric semantic fusion," *Comput. Graph. Forum*, vol. 37, no. 7, pp. 25–35, Oct. 2018.

[17] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4205–4212.

[18] M. Runz, M. Buffier, and L. Agapit, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE Int. Symp. Mix. Augment. Real. (ISMAR)*, Oct. 2018, pp. 10–20.

[19] Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Real-time progressive 3D semantic segmentation for indoor scenes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1089–1098.

[20] K. Genova et al., "Learning 3D semantic segmentation with only 2D image supervision," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 361–372.

[21] D. Huang, S. Peng, T. He, H. Yang, X. Zhou, and W. Ouyang, "Ponder: Point cloud pre-training via neural rendering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 16089–16098.

[22] H. Zhu et al., "PonderV2: Pave the way for 3D foundation model with a universal pre-training paradigm," 2023, *arXiv:2310.08586*.

[23] Y. Zhang, S. Song, P. Tan, and J. Xiao, "PanoContext: A whole-room 3D context model for panoramic scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 668–686.

[24] M. Rong, H. Cui, and S. Shen, "Efficient 3D scene semantic segmentation via active learning on rendered 2D images," *IEEE Trans. Image Process.*, vol. 32, pp. 3521–3535, 2023.

[25] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5099–5108.

[27] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[28] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. NeurIPS*, Dec. 2018, pp. 828–838.

[29] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[30] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6410–6419.

[31] Y. Lin et al., "FPConv: Learning local flattening for point convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4292–4301.

[32] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5588–5597.

[33] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Fusion-aware point convolution for online semantic 3D scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4533–4542.

[34] H. Shuai, X. Xu, and Q. Liu, "Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 4973–4984, 2021.

[35] P. Xiang, X. Wen, Y.-S. Liu, H. Zhang, Y. Fang, and Z. Han, "Retro-FPN: Retrospective feature pyramid network for point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17826–17838.

[36] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6620–6629.

[37] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, 2017.

[38] L. Han, T. Zheng, L. Xu, and L. Fang, "OccuSeg: Occupancy-aware 3D instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2937–2946.

[39] X.-L. Li, M.-H. Guo, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Long range pooling for 3D large-scale scene understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10300–10311.

[40] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN: A network with an edge," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 90:1–90:12, 2019.

[41] S.-M. Hu et al., "Subdivision-based mesh convolution networks," *ACM Trans. Graph.*, vol. 41, no. 3, pp. 1–16, Jun. 2022.

[42] X.-L. Li, Z.-N. Liu, T. Chen, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Mesh neural networks based on dual graph pyramids," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 7, pp. 4211–4224, Jul. 2024.

[43] H.-Y. Peng, M.-H. Guo, Z.-N. Liu, Y.-L. Yang, and T.-J. Mu, "MWFormer: Mesh understanding with window-based transformer," *Comput. Graph.*, vol. 115, pp. 382–391, Oct. 2023.

[44] Z. Hu et al., "VMNet: Voxel-mesh network for geodesic-aware 3D semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15468–15478.

[45] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[46] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer V2: Grouped vector attention and partition-based pooling," in *Proc. NeurIPS*, Nov. 2022, pp. 1–13.

[47] W. Wu, L. Fuxin, and Q. Shan, "PointConvFormer: Revenge of the point-based convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21802–21813.

[48] X. Lai et al., "Stratified transformer for 3D point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8500–8509.

[49] P.-S. Wang, "OctFormer: Octree-based transformers for 3D point clouds," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–11, Aug. 2023.

[50] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D semantic segmentation with superpoint transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 17195–17204.

[51] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9224–9232.

[52] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3075–3084.

[53] G. Liu, O. van Kaick, H. Huang, and R. Hu, "Active self-training for weakly supervised 3D scene semantic segmentation," *Comput. Vis. Media*, vol. 10, no. 3, pp. 425–438, Jun. 2024.

[54] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "RangeViT: Towards vision transformers for 3D semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5240–5250.

[55] A. Xiao et al., "3D semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9382–9392.

[56] J. Li, H. Dai, H. Han, and Y. Ding, "MSeg3D: Multi-modal 3D semantic segmentation for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21694–21704.

[57] G. Puy, A. Boulch, and R. Marlet, "Using a waffle iron for automotive point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 3379–3389.

[58] J. Sanchez, J.-E. Deschaud, and F. Goulette, "Domain generalization of 3D semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 18077–18087.

[59] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3050–3057.

[60] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 11214. Cham, Switzerland: Springer, Sep. 2018, pp. 458–474.

[61] Y. Lu, M. Zhen, and T. Fang, "Multi-view based neural network for semantic segmentation on 3D scenes," *Sci. China Inf. Sci.*, vol. 62, no. 12, Dec. 2019, Art. no. 229101.

[62] Ò. Lorente, I. Riera, and A. Rana, "Scene understanding for autonomous driving," 2021, *arXiv:2105.04905*.

[63] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3D for autonomous agents: A survey," *IEEE Access*, vol. 7, pp. 1859–1887, 2019.

[64] L. Roldao, R. de Charette, and A. Verroust-Blondet, "3D semantic scene completion: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 8, pp. 1–28, 2022.

[65] J. Zhao et al., "The fusion strategy of 2D and 3D information based on deep learning: A review," *Remote Sens.*, vol. 13, no. 20, p. 4029, Oct. 2021.

[66] R. Zhang, G. Li, T. Wunderlich, and L. Wang, "A survey on deep learning-based precise boundary recovery of semantic segmentation for images and point clouds," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 102, Oct. 2021, Art. no. 102411.

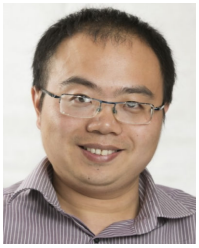
[67] D. Robert, B. Vallet, and L. Landrieu, "Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5565–5574.

[68] J.-X. Cai, T.-J. Mu, Y.-K. Lai, and S.-M. Hu, "LinkNet: 2D–3D linked multi-modal network for online semantic segmentation of RGB-D videos," *Comput. Graph.*, vol. 98, pp. 37–47, Aug. 2021.

[69] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "Supervoxel convolution for online 3D semantic segmentation," *ACM Trans. Graph.*, vol. 40, no. 3, pp. 1–15, Jun. 2021.
[70] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[71] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[72] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI*, Feb. 2018, pp. 7582–7589.
[73] J. Han, L. Yang, D. Zhang, X. Chang, and X. Liang, "Reinforcement cutting-agent learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9080–9089.
[74] W. Dong, Z. Zhang, and T. Tan, "Attention-aware sampling via deep reinforcement learning for action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jan. 2019, pp. 8247–8254.
[75] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. MICCAI*, in Lecture Notes in Computer Science, vol. 9901, Oct. 2016, pp. 424–432.
[76] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 3, pp. 24:1–24:18, Jul. 2017.
[77] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[78] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.
[79] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14679–14694, Dec. 2023.
[80] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Sys. (NeurIPS)*, vol. 34, Dec. 2021, pp. 12077–12090.
[81] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann, "Mix3D: Out-of-context data augmentation for 3D scenes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 116–125.
[82] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3D scene understanding with contrastive scene contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15582–15592.

Tai-Jiang Mu received the bachelor's and Ph.D. degrees from the Department of Computer Science and Technology, Tsinghua University, in 2011 and 2016, respectively. He is currently an Assistant Researcher with the Department of Computer Science and Technology, Tsinghua University. His research interests include visual media learning, computer graphics, and image processing.

Ming-Yuan Shen received the B.S. degree in computer science from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2021, where he is currently pursuing the master's degree. His research interests include computer graphics, machine learning, and image processing.



Yu-Kun Lai (Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science and Informatics, Cardiff University, U.K. His research interests include computer graphics, geometric processing, computer vision, and image processing. He is on the editorial boards of IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS and The Visual Computer. For more information, visit https://users.cs.cf.ac.uk/Yukun.Lai/.

Shi-Min Hu (Fellow, IEEE) received the Ph.D. degree from Zhejiang University in 1996. He is currently a Professor in computer science with Tsinghua University, Beijing, China. He has published more than 100 papers in journals and refereed conferences. His research interests include geometry processing, image and video processing, rendering, computer animation, and CAD. He is on the editorial boards of several journals, including *Computer-Aided Design* and *Computer & Graphics*. He is the Editor-in-Chief of *Computational Visual Media*.