Supervoxel Convolution for Online 3D Semantic Segmentation

SHI-SHENG HUANG, Tsinghua University, China
ZEYU MA*, Princeton University, USA
TAI-JIANG MU, Tsinghua University, China
HONGBO FU, City University of Hong Kong, China
SHI-MIN HU†, Tsinghua University, China

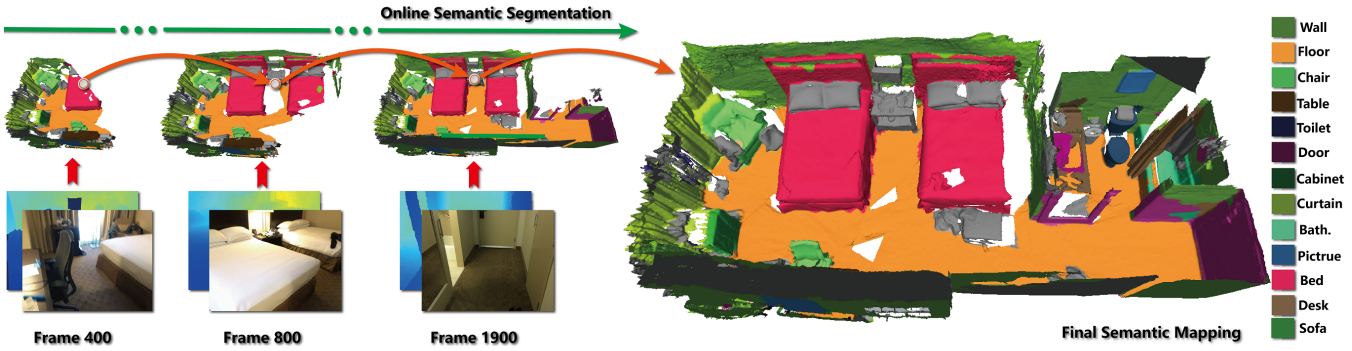

Fig. 1. We present an effective supervoxel convolution operation (*SVConv* for short) and apply it to 2D-3D joint learning for semantic mapping, which performs online dense semantic segmentation as well as scene reconstruction. Our approach strikes a significantly better balance between efficiency and segmentation accuracy than the existing online 3D semantic segmentation techniques.

Online 3D semantic segmentation, which aims to perform real-time 3D scene reconstruction along with semantic segmentation, is an important but challenging topic. A key challenge is to strike a balance between efficiency and segmentation accuracy. There are very few deep learning based solutions to this problem, since the commonly used deep representations based on volumetric-grids or points do not provide efficient 3D representation and organization structure for online segmentation. Observing that on-surface supervoxels, i.e., clusters of on-surface voxels, provide a compact representation of 3D surfaces and brings efficient connectivity structure via supervoxel clustering, we explore a supervoxel-based deep learning solution for this task.

To this end, we contribute a novel convolution operation (SVConv) directly on supervoxels. SVConv can efficiently fuse the multi-view 2D features and 3D features projected on supervoxels during the online 3D reconstruction, and leads to an effective supervoxel-based convolutional neural network, termed as *Supervoxel-CNN*, enabling 2D-3D joint learning for 3D semantic prediction. With the *Supervoxel-CNN*, we propose a *clustering-then-prediction* online 3D semantic segmentation approach. The extensive evaluations on the public 3D indoor scene datasets show that our approach significantly outperforms the existing online semantic segmentation systems in terms of efficiency or accuracy.

CCS Concepts: • **Computing methodologies** → *Computer Graphics*; Artificial intelligence; *Machine Learning*.

Additional Key Words and Phrases: Depth Fusion; Semantic Mapping; Supervoxel Clustering; Supervoxel Convolution; Deep Learning.

ACM Reference Format:
Shi-Sheng Huang, Zeyu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. 2021. Supervoxel Convolution for Online 3D Semantic Segmentation . *ACM Trans. Graph.* 1, 1 (March 2021), 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online 3D semantic segmentation together with on-the-fly 3D reconstruction has become urgent and crucial for applications involving instant scene understanding such as service robotics and autonomous driving [Dong et al. 2019; Liu et al. 2018; Zheng et al. 2019]. Directly segmenting progressively fused 3D geometry is often time-consuming. A common solution is to perform the 3D segmentation in a 2D-to-3D semantic mapping manner [Jeon et al. 2018;

*Work conducted in Tsinghua University.
†Shi-Min Hu is the corresponding author.

Authors' addresses: Shi-Sheng Huang, BNRist, Department of Computer Science and Technology, Tsinghua University, Room 3-507, 3-523, Information Technology Building (FIT), Beijing, China, shishenghuang.net@gmail.com; Zeyu Ma, Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540-5233, USA, mzy16@mails.tsinghua.edu.cn; Tai-Jiang Mu, BNRist, Department of Computer Science and Technology, Tsinghua University, Room 3-507, 3-523, Information Technology Building (FIT), Beijing, China, taijiang@tsinghua.edu.cn; Hongbo Fu, School of Creative Media, City University of Hong Kong, China, hongbofu@cityu.edu.hk; Shi-Min Hu, BNRist, Department of Computer Science and Technology, Tsinghua University, Room 3-507, 3-523, Information Technology Building (FIT), Beijing, China, shimin@tsinghua.edu.cn.

0730-0301/2021/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

McCormac et al. 2017a; Narita et al. 2019; Pham et al. 2019], i.e., mapping the 2D semantics from multiple views to a progressively reconstructed 3D surface. However, these traditional solutions usually fuse the 2D semantics in a naïve Bayesian style, which constrains 3D segmentation results to a low level of accuracy.

Recently, deep neural networks have achieved promising results for various 3D shape analysis and understanding tasks, such as 3D object classification [Graham et al. 2018; Qi et al. 2017a,b; Sedaghat et al. 2017; Zhao et al. 2019], 3D scene semantic segmentation [Choy et al. 2019; Dai and Nießner 2018; Wu et al. 2019], instance segmentation [Han et al. 2020; Hou et al. 2019], scene completion [Dai et al. 2018; Song et al. 2017] and object localization [Avetisyan et al. 2019a,b, 2020; Chen et al. 2020; Wald et al. 2019]. However, most of these attempts are performed in an offline manner, assuming that the inputs are already complete 3D scenes or objects. Online scene understanding especially for semantic mapping has not benefited too much from the advancement of deep learning techniques.

Directly apply the volumetric-based convolution [Graham et al. 2018; Hou et al. 2019], point-based convolution [Qi et al. 2017a,b; Wu et al. 2019], or even sparse convolution [Choy et al. 2019] to online 3D semantic segmentation by simply converting densely reconstructed scenes to volumetric-grids or a set of points is inefficient due to the huge amounts of data to be handled (e.g., a 5m $\times$ 5m $\times$ 3m room would result in 600 million voxels at a voxel size of 5mm). The joint learning by fusing 3D geometry features and 2D features from multi-view frames [Graham et al. 2018; Hou et al. 2019] could be even more difficult for online 3D semantic segmentation due to the lack of an efficient organization structure for the 2D-3D data association, though it has more potential to improve the semantic segmentation accuracy. Very recently, Zhang et al. [2020] proposed probably the first deep learning method for online 3D segmentation. However, they can only predict semantics for a relatively small number of 3D points (typically 512) per view with a moderate processing rate of 10fps. The key challenges still remain to be solved for the online 3D semantic segmentation via deep neural networks, i.e., how to re-organize the inherently unstructured 3D data in a structured representation, and design effective deep neural networks on such structured representation to balance the efficiency and accuracy for online 3D segmentation.

Our key observation is that not all voxels are meaningful to represent a progressively reconstructed 3D geometry, and only those on the geometry surface make sense. Based on this insight, we only track the on-surface voxels and cluster them into boundary-preserving supervoxels (with almost equal supervoxel size) via progressive supervoxel clustering, instead of randomly sampling points in each view as in [Zhang et al. 2020]. Supervoxel is a compact representation for 3D geometry with a much less number of units (100$\times$ less compared with the number of voxels). Besides, the supervoxel clustering step brings a very efficient on-surface connectivity structure between supervoxels, and this structure enables a very efficient convolution operation, leading to deep convolution neural networks for online 3D semantic segmentation.

However, how to perform the convolution on supervoxels has not been explored to the best of our knowledge and is nontrivial, since the convolutional kernel function and the neighborhood required for convolution are not well defined. In this paper, we propose a feasible convolution operation on supervoxels, named *SVConv*, and make such a supervoxel convolution effective with a carefully designed 2D-3D joint learning. Benefited from SVConv, we propose a deep convolution neural network, *Supervoxel-CNN*, with which a *clustering-then-prediction* semantic mapping approach is built for the online 3D semantic segmentation task. To the best of our knowledge, our work is the first to introduce such an effective supervoxel-based deep convolution neural network for the online 3D semantic segmentation task (Fig. 1).

We have extensively evaluated the efficiency and accuracy of our approach on the public 3D indoor benchmark, i.e. ScanNet v2 [Dai et al. 2017a] and SceneNN [Hua et al. 2016] datasets, compared to the state-of-the-art online and offline 3D semantic segmentation techniques. Our system significantly boosts the segmentation accuracy (see in Sec. 5) compared with the traditional Bayesian-style 2D-to-3D semantic mapping systems (e.g., SemanticFusion [McCormac et al. 2017a] and ProgressiveFusion [Pham et al. 2019] with more than 10% mIoU accuracy improvement. Although our approach outperforms the very recent deep learning based approach [Zhang et al. 2020] only with a slightly higher segmentation accuracy, we perform the online 3D segmentation at about 20fps, which is 2$\times$ faster than their system, demonstrating the efficiency of our proposed Supervoxel-CNN.

We summarize our technical contributions as: 1) We for the first time contribute a feasible convolution operation directly on supervoxels, i.e. SVConv, making it possible for an efficient 2D-3D joint learning for the online semantic segmentation task. 2) We propose a Supervoxel-CNN network and a *clustering-then-prediction* semantic mapping approach, which efficiently segments a progressively reconstructed 3D surface, achieving state-of-the-art online semantic segmentation accuracy.

2 RELATED WORK

In recent years we have seen great progress in real-time 3D scene reconstruction in computer vision, computer graphics, and robotics. Significant efforts have also been put to semantic scene understanding in either 2D or 3D. A full review of such topics is beyond the scope of this work. Below we discuss works mostly related to ours.

RGB-D Depth Fusion. Since the pioneer work of KinectFusion [Newcombe et al. 2011], a lot of efforts have been put to achieve real-time 3D scene reconstruction. To enable large-scale 3D reconstruction, many efficient data structures, such as Voxel-Hashing [Nießner et al. 2013] and Scalable-VoxelHashing [Chen et al. 2013], have been proposed for depth fusion based on a truncated signed distance function (TSDF) [Curless and Levoy 1996]. Besides, several works like RGB-D SLAM [Whelan et al. 2015], InfiniTAM [Kähler et al. 2015], ElasticFusion [Nießner et al. 2013], BundleFusion [Dai et al. 2017b] etc. achieve more accurate RGB-D depth fusion by using bundle adjustment and deformable loop closure. Wang et al. [2017] introduced a robust feature-based real-time 3D reconstruction approach by tracking the RGB features of 3D points from all frames. Recently, Cao et al. [2018] proposed a precise depth fusion approach to reduce the depth noise influence using noise-aware bundle adjustment. However, most of the current depth fusion systems have focused on 3D geometry reconstruction and

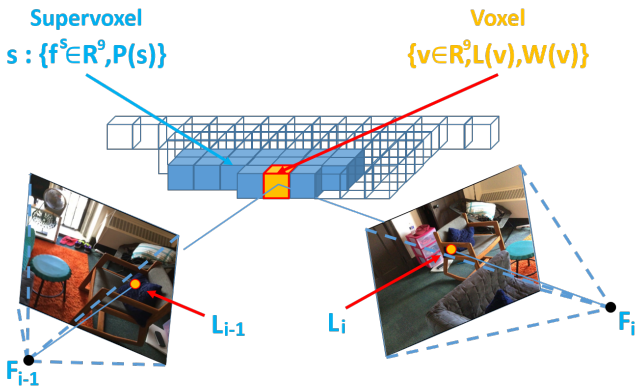

Fig. 2. Illustration of a supervoxel (containing highlighted voxels), and its feature and label representations.

very few of them support 3D semantic segmentation along with on-the-fly depth fusion.

The semantic information has been shown useful for object [Hu et al. 2018] or 3D scene reconstruction [McCormac et al. 2018; Yang et al. 2019; Zheng et al. 2019]. Some previous 3D depth fusion approaches have introduced semantic information to online 3D reconstruction by joint structure and semantic analysis [Zhang et al. 2015], object identification [Xu et al. 2016], object clustering [Nan et al. 2012], etc. Unlike these works, ours is based on the state-of-the-art depth fusion techniques like VoxelHashing [Nießner et al. 2013] and aims to improve the accuracy of semantic mapping but not geometry quality.

Semantic Segmentation. Due to the fast development of deep neural networks, scene understanding, especially in the tasks of 2D scene labeling, scene segmentation, object classification, has advanced significantly. A lot of techniques (e.g., DeconvNet [Noh et al. 2015], FCN [Shelhamer et al. 2017], Mask-RCNN [He et al. 2017], SSMA [Valada et al. 2020]) have been proposed for 2D segmentation. Although they produce impressive 2D segmentation for individual images, they often fail to provide consistent segmentation between consecutive RGB frames. A straightforward integration of them into a 2D-to-3D semantic mapping framework would cause uncertain association.

3D scene understanding based on deep 3D geometry learning has also advanced rapidly in recent years. For example, PointNet [Qi et al. 2017a] and its variations [Qi et al. 2018, 2017b] provide a powerful deep learning method to learn rotation- and translation-invariant deep features for unorganized point clouds. PointConv [Wu et al. 2019] introduces a point convolution operation with translation-invariant and permutation-invariant convolution learning for any point set. SparseConvNet [Graham et al. 2018] and MinkowskiNet [Choy et al. 2019] extend the convolution operation to high dimensional data with sparse convolution. 3DMV [Dai and Nießner 2018] and 3D-SIS [Hou et al. 2019] introduce a 2D-3D joint feature learning by projecting 2D features to 3D voxels with a volumetric-based convolution. FPConv [Lin et al. 2020] introduced a local flattening with a learnt weight map to project 3D points onto a 2D grid, thus enabling regular 2D convolution for efficient feature learning. To further improve the spatial consistency of 3D semantic segmentation, Hu et al. [2020] proposed dynamic region growing and data-driven context analysis with multi-scale processing for patch partition and classification.

Although these 3D learning methods achieve nice segmentation results, their inputs are a complete set of 3D points or a fully reconstructed scene. Our experiments will show that although some of those methods like MinkowskiNet [Choy et al. 2019] might achieve a higher segmentation accuracy than ours, they are not efficient enough for online 3D segmentation. The recent work OccuSeg [Han et al. 2020] proposes an occupancy-aware *learning-then-clustering* approach for 3D instance segmentation. Except from the different goals, our approach contributes a feasible 3D representation via supervoxel *clustering* for efficient 2D-3D joint online semantic segmentation *learning*, instead of clustering the surface patches with embedded deep features like OccuSeg.

Online Semantic Segmentation. The semantic segmentation in company with the on-the-fly 3D reconstruction performs the 3D scene geometry reconstruction and semantic understanding at the same time, and is deemed to be more suitable for online applications in robotics and virtual reality, thus receiving hot research attentions these years.

SemanticFusion [McCormac et al. 2017a] provides a pioneer solution for online 3D semantic segmentation with a Bayesian style 2D-to-3D mapping based on surfels. To reconstruct watertight surfaces, the subsequent works such as Semantic Reconstruction [Jeon et al. 2018] and PanopticFusion [Narita et al. 2019], extend the 2D-to-3D mapping framework to TSDF voxels. For better efficiency, ProgressiveFusion [2019] adopts to over-segment the voxels into supervoxel then performs 2D-to-3D semantic mapping based on supervoxels. A fundamental problem for these 2D-to-3D semantic mapping approaches is that they rely on 2D learning but lack 3D learning, thus limiting the further accuracy improvement. Compared with these approaches, our approach performs supervoxel-based 2D-3D joint learning to get much better segmentation results.

Very recently, Zhang et al. [2020] introduce an impressive point convolution for time-varying geometric data, and their approach achieves the state-of-the-art accuracy for online 3D semantic segmentation. However, this *point*-based segmentation approach is still not efficient, since the performance of its neighborhood management would descend rapidly with the increasing number of 3D points. An efficient data structure to organize 3D voxels is still urgently needed in the online semantic segmentation task. Compared to their approach, our supervoxel-based solution has two advantages. First, the supervoxel number is mainly influenced by the underlying geometry surface being reconstructed itself but not the number of camera views. Thus the number of supervoxels needed to be processed is significantly less than the number of points from multi-view as in Zhang et al. [2020]. Second, the supervoxel neighborhoods can be easily managed via supervoxel clustering, which does not need extra time-consuming management of neighborhoods.

There also exist relevant works [Valentin et al. 2015], which aim at interactive online 3D semantic segmentation. In contrast, our approach performs 3D segmentation automatically without any user intervention [Thanh Nguyen et al. 2017; Yang et al. 2017].

3 SUPERVOXEL CONVOLUTION

Towards an efficient and effective deep convolution network for online 3D semantic segmentation, we first introduce our progressive supervoxel clustering with boundary-preserving property, and then present a novel convolution operation on supervoxels.

3.1 Progressive Supervoxel Clustering

Supervoxel clustering is a technique for over-segmenting voxels into a graph of connected supervoxels, in which the voxel-to-supervoxel indexing and supervoxel-to-supervoxel neighborhood information can be efficiently extracted [Papon et al. 2013]. Besides, the number of supervoxels is significantly less than that of the original voxels.

On-surface Supervoxel Clustering. Although a depth fusion system like VoxelHashing [Nießner et al. 2013], BundleFusion [Dai et al. 2017b] et al. often needs to allocate a huge number of voxels, we only keep track of the on-surface voxels, denoted as $V = \{v_i = (p_i, n_i, c_i)\}$ with p_i, n_i, c_i encoding the position, normal and color of the voxel v_i, and cluster them into a set of supervoxels $\mathcal{S} = \{s_k, s_k \subset V\}$, as illustrated in Fig. 2. Since voxels across boundaries often belong to different objects thus with different semantic labels, we cluster voxels along object boundaries such that all the voxels inside a supervoxel can be assigned with the same label. To this end, we follow the latest technique proposed by Lin et al. [2018] to perform boundary-preserving supervoxel clustering. However, directly applying Lin et al. [2018]'s method to our online 3D semantic labeling task is not suitable since the original clustering implementation is too time-consuming with slow convergence.

To address this issue, we make two modifications. First, we relax their fusion-based minimization function using a much larger initialization parameter λ (with 8× larger). Second, we terminate the exchange-based minimization once the number of voxels to be exchanged is lower than a certain threshold (2,000 in all our experiments). Please refer to Lin et al. [2018] for more technical details. In this way, the clustering can converge much faster to keep pace with 3D depth fusion. Besides, we constrain the number of voxels in each clustered supervoxel such that the clustered supervoxels have a nearly equal number of voxels. This constraint is meaningful for the derivation of supervoxel convolution in Sec. 3.2.

Progressive Clustering. Since the previously reconstructed surface would not influence the current reconstruction during incremental 3D reconstruction, we can perform supervoxel clustering in a progressive way to further improve the efficiency of the whole system. Specifically, we track the latest on-surface voxels and divide them into two subsets: those overlapping with the previously clustered regions, U, and the remaining U'. We further identify the voxels $S \subset U$, satisfying that at least one of its siblings in the same supervoxel 1) is updated due to the depth fusion from the latest frame, or 2) is neighboring to U'. Finally, we only cluster the voxels in $S \cup U'$ and leave the the rest of the voxels unchanged. Since voxels in S have already been clustered in the previous round, we adopt to transfer the previous clustering centers to these voxels and perform the supervoxel clustering only for the newly added voxels in U'. In most of the cases, since the voxel number in U' is far less than S between consecutive frames during 3D reconstruction, our progressive clustering can be performed very efficiently. In this way, we efficiently organize all the tracked on-surface voxels with supervoxels during the online processing. Sometimes, the voxel number of U' would not be enough to form a desired supervoxel. Since this scenario seldom occurs, we cluster such voxels into a new supervoxel and send it to the subsequent prediction.

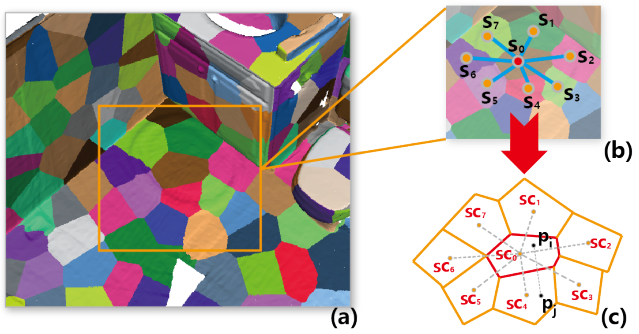

Fig. 3. Illustration of supervoxel convolution. For a set of supervoxels (a), the convolution for a supervoxel s_0 with its neighbors s_k, $k = \{1, ..., 7\}$ (b) is performed by estimating the supervoxel weight function $W_{sv}(\cdot)$ defined on the supervoxel centroid displacement $sc_0 - sc_k$ (c). This is an approximation of the convolution weight function $W(\cdot)$ defined on the voxel position displacement $p_i - p_j$ in PointConv convolution (c).

Properties of Supervoxels. During the on-the-fly depth fusion, our progressive supervoxel clustering approach organizes all 3D voxels as a set of supervoxels such the following properties are preserved. 1) The resulting supervoxels are located on the reconstructed geometry surface since the associated 3D voxels are all on-surface. 2) Those supervoxels have a nearly equal number of voxels, benefiting the definition of supervoxel convolution. 3) The neighboring relations among supervoxels are kept such that the neighbors of a supervoxel can be fetched in O(1) time given a small number of supervoxels, without the need of using extra time-consuming tree-based data structures as done in [Zhang et al. 2020]. Besides, the voxel-to-supervoxel indexing information within a supervoxel is also held, which can benefit the feature aggregation of supervoxel convolution in Sec. 3.2. 4) Supervoxels are boundary-preserving such that the voxels associated with each supervoxel can be predicted with the same label as the supervoxel. These properties make it possible to define a convolution operation directly on supervoxels for effective learning on 3D.

3.2 Supervoxel Convolution

Convolution operations have been proven to be very efficient in 3D geometry learning, such as mesh-based convolution [Hanocka et al. 2019], volumetric-based convolution [Dai and Nießner 2018; Hou et al. 2019], and point-based convolution [Wu et al. 2019]. Although a voxel could be regarded as a point in our case, the convolution operation on supervoxels is undefined. The core problems faced are: 1) how to define a supervoxel's position representation with which the convolution weight function can be estimated, and 2) how to define a supervoxel's feature representation. In this subsection, we present *SVConv*, which extends convolution for points [Wu et al.

2019] to supervoxels, as a feasible convolution operation directly on supervoxels.

From PointConv to SVConv. PointConv [Wu et al. 2019] introduces a novel convolution operation over 3D points as:

$$F_{out}(p, W, F_{in}) = \sum_{p_j \in \Omega} W(p - p_j) F_{in}(p_j),$$

where Ω is a set of neighboring points of a point p, $W(\cdot)$ is the local convolution weight function, and $F_{in}(\cdot)$ and $F_{out}(\cdot)$ are the input and output features of each point, respectively. Following PointConv, we consider a supervoxel s_0 with its neighbors $\{s_k\}$, $k = \{1, 2, ..., K\}$ (e.g., $K = 7$ in Fig. 3(b)) and convolve the input feature F_{in} of each voxel in s_0 by the weight function $W(\cdot)$ defined on the voxels' position displacement. Specifically, for two voxels v_i and v_j, their position displacement is defined as $p_i - p_j$. Then we formulate the PointConv for voxel $v_i \in s_0$ as:

$$F_{out}(v_i) = \sum_{k=0}^{K} \sum_{v_j \in s_k} W(p_i - p_j) F_{in}(v_j). \tag{1}$$

Since the supervoxel's size is almost fixed due to almost the same number of voxels in each supervoxel, we can approximate the position displacement $p_i - p_j$ ($v_i \in s_0, v_j \in s_k$) as the displacement of their corresponding supervoxel centroids $sc_k = \sum_{p_j \in s_k} p_j / |s_k|$ using Taylor expansion: $p_i - p_j \approx sc_0 - sc_k + o(sc_0 - sc_k)$ with $o(\cdot)$ being the Peano remainder. Following the Taylor's theorem, we approximate the weight function $W(p_i - p_j) \approx W(sc_0 - sc_k + o(sc_0 - sc_k)) \approx W(sc_0 - sc_k) + o(sc_0 - sc_k)^T \frac{\partial W}{\partial (sc_0 - sc_k)}$. By defining a new convolution weight function as $W_{sv}(sc_0 - sc_k) = |s_k|(W(sc_0 - sc_k) + o(sc_0 - sc_k)^T \frac{\partial W}{\partial (sc_0 - sc_k)})$, we can re-write Equation (1) as:

$$F_{out}(v_i) \approx \sum_{k=0}^{K} W_{sv}(sc_0 - sc_k) \{ \frac{1}{|s_k|} \sum_{v_j \in s_k} F_{in}(v_j) \}. \tag{2}$$

By averaging the output features of all voxels v_i in supervoxel s_0 and considering the nearly equal size of all supervoxels $|s_0| \approx |s_k|, k = \{1, 2, ..., K\}$ (Sec. 3.1), we can obtain:

$$\frac{1}{|s_0|} \sum_{v_i \in s_0} F_{out}(v_i) \approx \sum_{k=0}^{K} W_{sv}(sc_0 - sc_k) \{ \frac{1}{|s_k|} \sum_{v_j \in s_k} F_{in}(v_j) \}. \tag{3}$$

Equation (3) shows that we can define a new convolution operation directly on supervoxels. For each supervoxel s_k, if we utilize the centroid $sc_k = \sum_{p_j \in s_k} p_j / |s_k|$ as its position representation, the input feature $SV_{in}(s_k)$ as the average of its associated voxels' input features $SV_{in}(s_k) = \frac{1}{|s_k|} \sum_{v_j \in s_k} F_{in}(v_j)$ and the output feature $SV_{out}(s_k)$ as the average of its associated voxels' output features $SV_{out}(s_k) = \frac{1}{|s_k|} \sum_{v_j \in s_k} F_{out}(v_j)$, we can define a supervoxel-based convolution operation (SVConv) for each supervoxel s_0 with its neighbors $\{s_k\}$ as:

$$SV_{out}(s_0) = \sum_{k=0}^{K} W_{sv}(sc_0 - sc_k) SV_{in}(s_k), \tag{4}$$

In this way, our SVConv can be viewed as an approximation of PointConv in the supervoxel level, which is also effective for 3D classification.

Position Representation. From the definition of SVConv, we calculate the centroid of a surpervoxel as its position, and then estimate the convolution weight function $W_{sv}(\cdot)$ in its local neighbors. Similar to PointConv [Wu et al. 2019], we approximate this weight function using a set of Multi-Layer Perceptrons (MLPs).

Feature Representation for 2D-3D Joint Learning. Since a supervoxel's feature representation is the average of its voxels' features, descriptive features are required for voxels. The works of 3DMV [Graham et al. 2018], 3D-SIS [Hou et al. 2019] and Zhang et al. [2020] have shown that the fusion of 3D features with 2D deep features from multi-view is effective for 3D semantic prediction. However, this 2D-3D fusing operation is not suitable for our *dense* setting. First, the mutlti-view 2D-3D management of each 3D voxel would be too time-consuming to be company with the 3D reconstruction system. More importantly, storing the 2D deep feature for each voxel during the dense reconstruction would result in memory explosion.

To address this issue, we choose to fuse the relatively solidified 2D semantic probability distribution instead of the original 2D deep feature. What's more, to avoid storing an $M-$dimensional probability distribution (M is the number of labels) for each voxel, we only store a single label along with its confidence and update the label and confidence in a *max-pooling* way during the dense reconstruction. More specifically, as shown in Fig. 2, for a voxel v with label $L(v)$ and confidence $W(v)$, if a newly detected 2D label L_i equals $L(v)$, we keep v's label unchanged and increase its confidence weight by one: $W(v) \leftarrow W(v) + 1$. Otherwise, we decrease its weight by one, and in case of a negative weight, we replace its label with L_i and reset its weight: $W(v) \leftarrow 0$. Then as for the supervoxel s, we calculate the supervoxel prediction probability distribution $P(s) = (p^s(l_1), ..., p^s(l_M))$ by averaging over all the associated voxels for l_i, weighted by their associated confidence weights: $p^s(l_i) = \frac{1}{\sum_{v_j \in s} W(v_j)} \sum_{v_j \in s} W(v_j) \cdot I(l(v_j) = l_i)$, with $I(\cdot)$ the indictor function. For the supervoxel's 3D features, we adopt the voxel's 9-dimensional 3D geometry features and perform the 3D feature averaging with $f^s = \frac{1}{|s|} \sum_{v_i \in s} v_i$.

Finally, we concatenate the supervoxel's 3D geometry features and the 2D prediction probability distribution together, yielding a fused 2D-3D feature $SV(s) = (f^s, P(s)) \in \mathbf{R}^{9+M}$. As shown in Sec. 5, our fused 2D-3D features for supervoxels are effective for supervoxel convolution as a 2D-3D joint learning dedicated to semantic prediction during the online dense 3D reconstruction, balancing the time complexity and memory usage.

Properties of SVConv. Although SVConv is derived from PointConv, it is distinct from PointConv with at least three major benefits for the online semantic segmentation task: (1) Feature aggregation. The supervoxel's compact representation stores the voxel-to-supervoxel indexing information, thus making it possible to compute the supervoxel prediction probability distribution (see the feature representation above) for the 2D-3D feature aggregation. We show that without such feature aggregation, a naïve PointConv on uniformly sampled voxels is not very effective to achieve a high accuracy of semantic segmentation (see the evaluation in Sec. 5.3). (2) Supervoxel size constraint. Our SVConv is only feasible when the supervoxels have a nearly equal size (in term of voxel numbers) (see

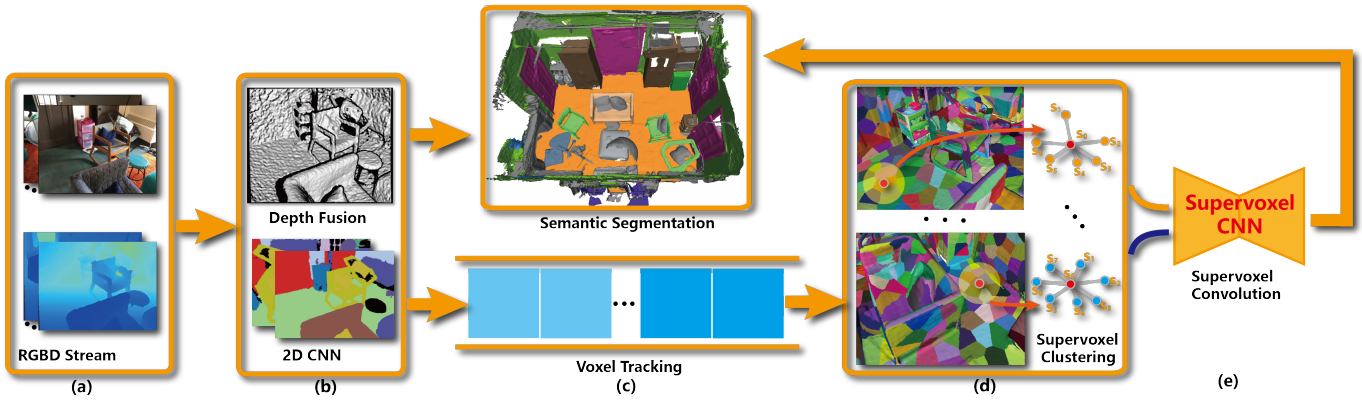

Fig. 4. System overview of online 3D semantic segmentation based on Supervoxel-CNN. It takes the RGB-D stream as input (a) and performs in the loop depth-fusion-based scene reconstruction with semantic labels projected from the 2D CNN semantic prediction on keyframes (b). A voxel tracking module (c) is performed to collect the latest-ready 3D voxels. We run supervoxel clustering to progressively cluster the latest-ready 3D voxels into supervoxels with their neighborhoods managed efficiently (d). Those clustered supervoxels are then fed to Supervoxel-CNN with supervoxel convolution for 2D-3D joint learning of semantic prediction (e), leading to a reconstructed 3D scene with dense semantic segmentation.

Equation 3). Without this constraint, the semantic segmentation accuracy will decrease, as shown in Sec. 5.4 ("Supervoxel-CNN w/o BP"). (3) On-surface neighbors. Based on supervoxel clustering, our SVConv performs the convolution operation on on-surface neighbors efficiently. In contrast, PointConv could not ensure the on-surface neighbors, thus missing the on-surface neighborhood benefits.

4 SUPERVOXEL-CNN FOR ONLINE 3D SEMANTIC SEGMENTATION

Based on the progressively clustered supervoxels and the defined *SVConv*, we build a deep neural network with all convolution layers, called Supervoxel-CNN. With Supervoxel-CNN, we demonstrate a *clustering-then-prediction* approach for online 3D semantic segmentation with the state-of-the-art efficiency and accuracy.

4.1 Approach

The pipeline of our online 3D semantic segmentation enabled by Supervoxel-CNN is shown in Fig. 4. It contains an online reconstruction module with VoxelHashing [Nießner et al. 2013] for camera pose estimation and dense 3D geometric reconstruction. Alongside the online reconstruction, we extract the 2D semantic label information by using a 2D CNN (i.e., SSMA [Valada et al. 2020] in our implementation) on keyframes. In parallel with the semantic reconstruction, we implement a voxel tracking module to track the on-surface voxels, whose geometry or semantic label changes during scene reconstruction. The tracked voxels are arranged in a buffer and then sent to supervoxel clustering. We perform boundary-preserving supervoxel *clustering* in a progressive way to organize the on-surface 3D voxels as supervoxels, whose voxel-to-supervoxel index and supervoxel-to-supervoxel neighbors are also efficiently maintained. The latest supervoxels and their neighbors are then fed to Supervoxel-CNN to *predict* the final semantic labels for a reconstructed 3D scene. Thereafter each voxel $v \in s$ is automatically assigned with the same label as s, and updated back to the reconstruction process. The online segmentation results at frame 400, 800, 1900, and the final results of scene0435_01 from the ScanNet v2 validation set are presented in Fig. 1. Our proposed Supervoxel-CNN achieves highly accurate segmentation results, at a processing rate of about 20fps.

4.2 Supervoxel-CNN

The aforementioned supervoxel convolution makes it possible for us to build a deep convolution neural network to directly learn the semantic labels of supervoxels. The backbone of our Supervoxel-CNN for 3D semantic segmentation is illustrated in Fig. 5. It takes as input a set of supervoxels $\mathcal{S} = \{s_i, i = 1, ..., n\}$, with each supervoxel s_i affiliated with the centroid displacements between its K neighborhoods (a $K \times 3$ vector) and their corresponding fused 2D-3D features (a $K \times (9 + M)$ vector). The input centroid displacement vector is fed to a set of MLPs with sharing weights, batch normalization and ReLU activation, and then convoluted (matrix multiply) with the input fused 2D-3D features, followed by *Conv2D*, Reshape and MLPs. Finally, a softmax layer is applied to generate the final semantic probability prediction for supervoxels. Besides, we also apply a CRFasRNN [Zheng et al. 2015] layer on the output of the backbones for further smoothing the semantic prediction results.

4.3 Training Details

Preparing Training Data. We train Supervoxel-CNN on the ScanNet v2 dataset [Dai et al. 2017a], which contains in total 1,513 real world RGB-D sequences with annotated 3D scenes, 1,201 sequences for training and 312 sequences for validation. Since the annotation data in ScanNet is based on voxels instead of supervoxels, we need to adapt their data for our purpose. To create the supervoxel-based training data, we implement a semantic reconstruction system without the Supervoxel-CNN part, called *SV-SemanticFusion* for short).

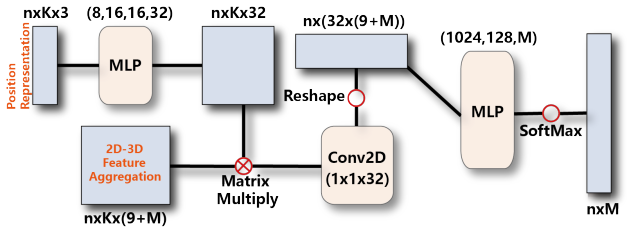

Fig. 5. The backbone of our Supervoxel-CNN architecture.

We use *SV-SemanticFusion* to collect supervoxels for training. Specifically, the output of *SV-SemanticFusion* for the i-th sequence data is a set of supervoxels with K neighbors for each supervoxel, denoted as $\mathcal{S}_i = \{s_j, j = 1, ...k_i\}$ with a corresponding position representation set $\mathcal{F}_i = \{f^s | s \in \mathcal{S}_i\}$ and a feature representation set $\mathcal{P}_i = \{P(s) | s \in \mathcal{S}_i\}$ computed as described in Sec. 3.2. For each computed supervoxel $s_j \in \mathcal{S}_i$, we seek a ground-truth semantic label $\hat{l}_j \in \mathcal{L}$ from the corresponding 3D annotated mesh and denote the ground-truth label set for $\mathcal{S}_i$ as $\hat{\mathcal{L}}_i = \{\hat{l}_j, j = 1, ..., k_i\}$. By collecting all the 1,201 training sequences, we obtain the training data as $\mathcal{T} = \{\bigcup_i \mathcal{F}_i, \bigcup_i \mathcal{P}_i, \bigcup_i \hat{\mathcal{L}}_i\}$. In the *SV-SemanticFusion* system, we also adopt the SSMA [Valada et al. 2020] to compute the labels.

Training Parameters. When training our Supervoxel-CNN, we randomly select about 90% of the training data as a training dataset and the rest for cross validation. We use the Adam optimizer with the initial learning rate set as $l_r = 10^{-3}$. The batch size is set as 16 and the number of epochs is set as 10. We use the *categorical-hinge* loss and set the clip value as 0.5 to avoid gradient explosion. We train the network on a platform with NVIDIA Titan RTX (24G GPU) configuration. It takes about a few hours to train the network to reach an accuracy of 93% on the validation set.

5 EVALUATION AND ANALYSIS

To further demonstrate the advantages of our Supervoxel-CNN and the Supervoxel-CNN based *clustering-then-prediction* online 3D semantic segmentation approach (short for our approach) over traditional 2D-to-3D mapping and other learning based methods, we first conduct extensive qualitative (Sec. 5.1) and quantitative evaluations (Sec. 5.2) on two public 3D datasets (Scannet v2 [Dai et al. 2017a] and SceneNN [Hua et al. 2016]), and compare our approach with some representative offline 3D CNN approaches (Sec. 5.3). Then we perform an ablation study of the core part of our approach, i.e the Supervoxel-CNN (Sec. 5.4). Thereafter, we give a comprehensive time efficiency analysis of our approach (Sec. 5.5) and perform a time *vs* accuracy evaluation (Sec. 5.6) to see how our approach behaves in striking the balance between time efficiency and segmentation accuracy. In Section 5.7, we summarize our main limitations and discuss some meaningful directions to improve our approach.

5.1 Qualitative Evaluation

We first qualitatively compare our approach with two semantic mapping approaches, i.e., SemanticFusion (SF) [McCormac et al. 2017a] and ProgressiveFusion (PsF) [Pham et al. 2019]. SF is a representation for naïve Bayesian based semantic mapping methods such as Semantic Reconstruction (SR) [Jeon et al. 2018] and PanopticFusion (PF) [Narita et al. 2019]. Similar to ours, PsF is the only semantic mapping method that adopts supervoxels to organize voxels. Since the original SF implementation is based on surfel-based depth fusion, we re-implemented it based on TSDF-based depth fusion for a fair comparison. Besides, we also implemented the PsF system. For fairness, we run all the methods using the same camera trajectory for each sequence. Fig. 6 shows two representative semantic mapping results generated by these methods on the ScanNet v2 validation set. Benefiting from the effective joint 2D-3D learning on supervoxels, our approach achieves more accurate 3D segmentation results than both SF and PsF. Please refer to our supplementary material to see more qualitative comparison results.

5.2 Quantitative Evaluation

In this experiment, we adopt the same setting as the aforementioned qualitative evaluation and compare the semantic segmentation accuracy using two commonly used metrics, i.e., mAcc and mIoU, as described by the ScanNet v2 dataset. In addition to the four non-learning based methods, i.e., SF, PsF, SR, and PF, we also include the recent work of Zhang et al. [2020] (referred to as FA-PConv), which is also a deep learning based online 3D segmentation method like ours.

Comparisons with Non-3D-learning Methods. As shown in Table 1, our approach achieves significantly higher IoU than all the four non-3D-learning based semantic mapping methods in all 20 classes. In total, our approach significantly boosts the mAcc and mIoU with a large margin of 9.3% and 13.3%, respectively, compared to PsF, which performs the best among the four traditional methods. This confirms that our approach serves as a practical and effective learning approach for online 3D semantic segmentation.

Since our approach performs semantic segmentation in an online manner, we also present the mAcc and mIoU accuracy values of our system for every 100 frames in comparison with SF and PsF. As shown in Fig. 7, our approach helps achieve more consistent and accurate results than the other two traditional approaches.

Comparison with FA-PConv. Since FA-PConv performs convolution directly on 3D points, to achieve online performance, this method randomly samples a much smaller number of points (typically only 512 points for each frame) as input and thus only predicts labels for this sparse set of points. A post-processing is thus needed to 'transfer' labels from sparse points to dense points, by assigning the label of each dense point with that of its nearest sparse point, to perform the comparison. As shown the comparison results on the Scannet v2 validation set in Table 1, our approach outperforms FA-PConv in 8 classes and achieves nearly equal (within 1%) accuracy in 3 classes in terms of IoU. In total, our approach achieves slightly higher accuracy than PA-PConv with an improvement of 2.6% and 1.1% for mAcc and mIoU, respectively.

Although our approach performs only slightly better than FA-PConv in terms of quantitative evaluation, our approach achieves visually more consistent segmentation results. This is partially due to that the compact representation (supervoxel) our Supervoxel-CNN

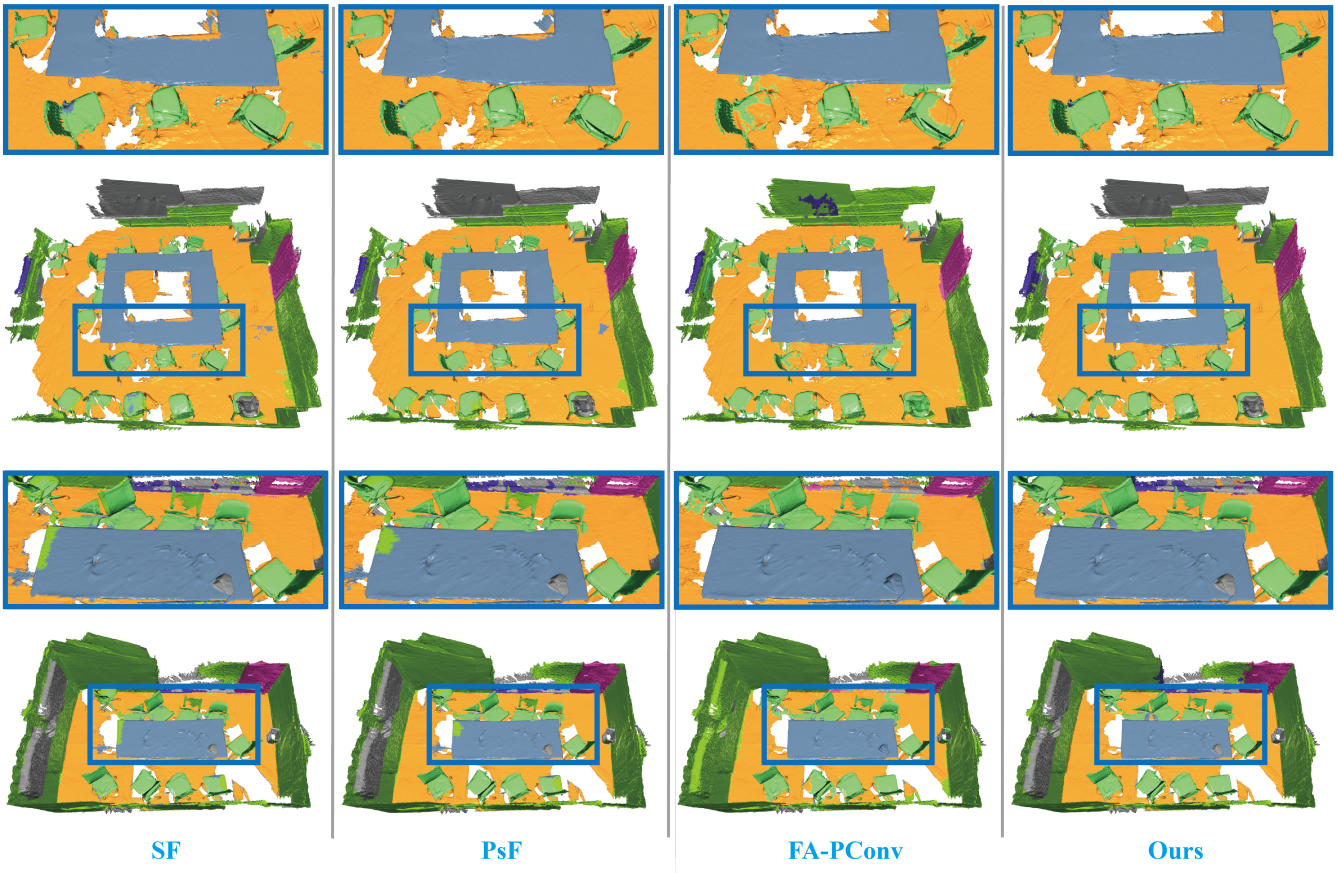

Fig. 6. Qualitative comparison of semantic segmentation results by different approaches, including SemanticFusion (SF), ProgressiveFusion (PsF), FA-PConv and ours. The two examples are from scene0430 (top) and scene0575 (bottom) in the ScanNet v2 validation set. Please refer to our supplementary material for more comparison results.

Table 1. The quantitative accuracy comparison with different online 3D semantic segmentation approaches on the ScanNet v2 validation set, including SemanticFusion (SF) [McCormac et al. 2017a], Semantic Reconstruction (SR) [Jeon et al. 2018], PanopticFusion (PF) [Narita et al. 2019], Progressive Fusion (PsF) [Pham et al. 2019], and FA-PConv (FPC) [2020]. For each class, the IoU is reported. '↑' means 'the larger, the better' for the underlying metrics(%), and the numbers in **boldface** indicate the best performance.

Md.	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	mAcc↑	mIoU↑
SF	58.3	72.2	35.7	46.8	46.9	43.7	37.8	35.7	29.5	32.0	21.7	33.2	34.7	46.9	34.3	28.8	65.5	47.2	59.8	34.7	47.4	42.2
SR	66.9	80.6	31.7	52.6	64.0	58.3	51.6	30.9	21.1	31.2	7.3	24.0	26.1	30.3	56.3	23.6	73.3	46.2	69.7	33.3	65.6	44.0
PF	65.7	81.0	44.8	67.0	66.7	61.3	54.8	35.6	45.9	48.4	30.2	35.8	42.0	53.3	47.0	50.8	82.1	52.6	57.5	40.3	68.7	53.1
PsF	70.6	87.1	48.6	61.2	68.4	59.8	54.0	47.5	47.0	65.7	25.7	41.7	48.9	54.9	41.8	34.5	78.9	53.4	65.6	43.7	70.3	55.0
FPC	**83.8**	**91.9**	**60.9**	**82.3**	75.1	**77.9**	**68.9**	**64.8**	56.3	64.0	**40.6**	56.0	58.2	64.8	**64.2**	51.7	**87.0**	**63.5**	**85.4**	46.4	77.0	67.2
Ours	80.5	91.1	60.5	78.5	**80.6**	72.6	64.4	60.5	**61.7**	**79.1**	35.0	**59.3**	**59.9**	**70.4**	57.5	**75.2**	86.4	61.3	73.4	**57.8**	**79.6**	**68.3**

adopted has the advantage of overcoming inconsistent segmentation. Fig. 6 shows several representative visual comparison examples evaluated on the Scannet v2 validation set. The segmentation results of FA-PConv are noisy in some object areas (such as 'chair' areas in Fig. 6). In contrast, our results are more consistent. Please refer to our supplementary material for more visual comparisons.

Besides, we also evaluate our approach on the Scannet v2 hidden test in the Scannet benchmark[1]. Our approach (termed as

[1]http://kaldir.vc.in.tum.de/scannet_benchmark/, 21st Oct. 2020

'Supervoxel-CNN') achieves an mIoU of 63.5, ranked as the state-of-art online 3D semantic segmentation approach (till the time of this submission). Please refer to our supplementary material to see the detailed comparison results.

Evaluation on SceneNN. Besides the ScanNet v2 dataset, we also perform an evaluation on the SceneNN dataset [Hua et al. 2016] to test the generalization ability of our approach. The SceneNN dataset is a public indoor 3D dataset consisting of various indoor scenes, e.g., offices, dormitory, classrooms, pantry etc., with 50 scans for the training and 26 scans for the test. Since only the 50 scans in the training set have public ground-truth semantic annotations, we perform the evaluation on those 50 scans. To evaluate the generalization ability of our approach, we only use the weights of Supervoxel-CNN pre-trained on the ScanNet v2 training dataset, without fine-tuning on SceneNN scans. The FA-PConv [Zhang et al. 2020] results are obtained using its publicly released code[2] with the default parameters settings. The segmentation accuracies of SF and PsF are fetched from the original paper of PsF [Pham et al. 2019]. Besides, we also adopt the segmentation accuracy metrics (i.e., mAcc and wIoU) used in PsF. Table 2 shows the average mAcc and wIoU of the four compared approaches evaluated on the SceneNN. Our method significantly outperforms SF and PsF in both the mAcc and wIoU accuracy, with more than 15% improvement. Our method also achieves a higher accuracy than FA-PConv with about 5% accuracy improvement in both mAcc and wIoU. Please refer to our supplementary materials for the detailed accuracy results on each scan and the visual comparison results on the SceneNN dataset.

Table 2. Comparison of the segmentation accuracy (average mAcc(%) and wIoU(%)) on the SceneNN dataset with the compared approaches, i.e., SemanticFusion (SF) [McCormac et al. 2017a], ProgressiveFusion (PsF) [Pham et al. 2019], FA-PConv(FPC) [Zhang et al. 2020], and ours. Please refer to the supplementary material for the detailed segmentation accuracy.

Metrics	SF	PsF	FPC	Ours
mAcc$^{\uparrow}$	58.50	61.60	71.70	**76.93**
wIoU$^{\uparrow}$	47.13	52.21	63.88	**69.00**

5.3 Comparisons with 3D CNNs

Our approach performs online semantic segmentation, which is different from many existing 3D CNN-based methods for offline 3D scene segmentation. However, to evaluate the segmentation accuracy of our method against the state-of-the-art methods, we also compare our approach with these offline segmentation methods by evaluating the semantic segmentation accuracy on a final reconstructed surface. For the 3D CNNs, we choose three state-of-the-art methods, i.e., PointNet++ [Qi et al. 2017b], 3DMV [Dai and Nießner 2018] and MinkowskiNet [Choy et al. 2019], and use their open-source code (or accuracy reported in the original papers) for the comparison. The evaluation is again performed on the ScanNet v2 validation set. Table 3 shows the mAcc accuracy of the final segmentation results using the four compared methods. Our approach achieves higher mAcc accuracy than PointNet++ and 3DMV, but lower than MinkowskiNet.

Although MinkowskiNet achieves better segmentation accuracy than our approach, it is nontrivial to directly apply MinkowskiNet to our online task. One important issue is the time efficiency. In the online semantic segmentation task, a huge number of 3D voxels (about millions) make the voxel-based semantic prediction very time-consuming with MinkowskiNet. This is verified by the timing evaluation made in Sec. 5.5. An alternative solution is to progressively predict the voxels' semantic labels for a partially reconstructed scene. However, as shown in Fig. 8, MinkowskiNet often fails to make accurate semantic prediction for partially reconstructed 3D objects. This suboptimal performance for partially reconstructed 3D objects is a common issue for the 3D geometry learning approaches such as PointNet, PointNet++, and MinkowskiNet. In contrast, our approach performs better, mainly because of our adopted 2D-3D joint learning strategy, which makes use of more descriptive 2D deep features for partially reconstructed 3D objects. Please refer to our supplementary materials for more visual results.

SVConv versus PointConv. As aformentioned in Sec. 3.2, the design of our SVConv has several advantages over the original PointConv [Wu et al. 2019] for the online 3D semantic segmentation task. To verify our claims, we build a *PointConv-Uniform* model, which performs a progressive semantic prediction on the uniformly sampled on-surface voxels with PointConv. For an efficient implementation, we use the supervoxels' centroids as the uniformly sampled on-surface voxels, since our supervoxel clustering can be seen as a nearly uniform voxel sampling. For feature representations, since those centroids are isolated voxels without neighboring information, we use the label distribution projected from 2D CNN as the feature representation fed to PointConv. Table 4 shows that our approach also outperforms this alternative solution with an improvement of 4.5% and 2.9% in mAcc and mIoUrespectively, demonstrating that our proposed SVConv is more effective than PointConv.

5.4 Ablation Study

Boundary-preserving Supervoxel. To investigate the impact of the boundary-preserving supervoxels on the 3D semantic segmentation accuracy, we trained Supervoxel-CNN using non-boundary-preserving supervoxels (generated by VCCS [Papon et al. 2013], as shown in Fig. 9), referred to as "Supervoxel-CNN w/o BP". As shown in Table 4, compared to Supervoxel-CNN trained on boundary-preserving supervoxels, this non-boundary-preserving Supervoxel-CNN drops off 4.3% and 2.6% in mAcc and mIou, respectively. One main reason would be that the non-boundary-preserving supervoxels are inherently more likely to mis-classify the voxels close to object boundaries. Fig. 10 shows a visual comparison between semantic segmentation using "Supervoxel-CNN FULL" and "Supervoxel-CNN w/o BP".

2D/3D Features. A descriptive feature is crucial for learning. To verify the importance of the fused 2D-3D features, we train Supervoxel-CNN using only the 3D geometric feature f^s or the 2D semantic prediction probability feature $P(s)$ as the feature representation for every supervoxel s, called "Supervoxel-CNN w/o 3D" and "Supervoxel-CNN w/o 2D", respectively. As shown in Table 4, the

[2]https://github.com/jzhzhang/FusionAwareConv

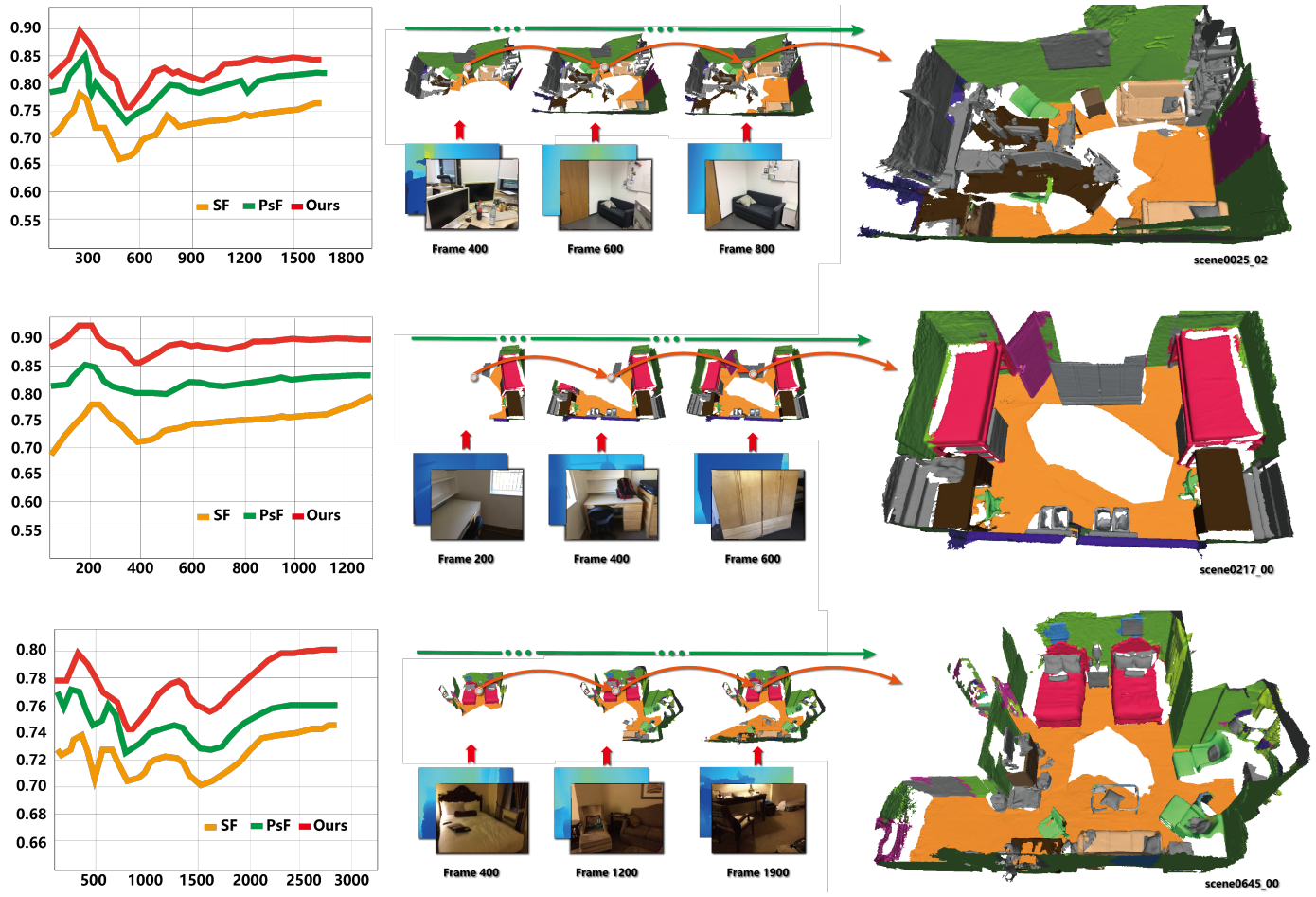

Fig. 7. The progressive mAcc accuracy values (1st Column) by three compared methods, including SemanticFusion (SF), ProgressiveFusion (PsF), and ours, tested on sequences scene0025 (top), scene0217 (middle) and scene645 (bottom) from the ScanNet v2 validation set. 2nd and 3rd Column: the progressive and final 3D segmentation results by our method. Please refer to the supplementary material for more results.

Table 3. The quantitative accuracy comparison of the final semantic segmentation results between our approach and offline semantic segmentation methods on the ScanNet v2 validation set, including PointNet++ (PN++) [Qi et al. 2017b], 3DMV (with 5-views) [Dai and Nießner 2018], and MinkowskiNet (MkNet) [Choy et al. 2019]. Note that the result values of PN++ and 3DMV are adopted from the original paper of 3DMV with only the Acc reported for each class. So the results of MkNet and ours also report the Acc for each class in the table for easy comparison.

Md.	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	mAcc↑
PN++	89.5	**97.8**	39.8	69.7	86.0	68.3	59.6	27.5	23.7	84.3	0	37.6	66.7	48.7	54.7	85.0	84.8	62.8	86.1	30.7	60.2
3DMV	73.9	95.6	69.9	80.7	85.9	75.8	67.8	86.6	61.2	88.1	**55.8**	31.9	73.2	**82.4**	74.8	82.6	88.3	72.8	**94.7**	58.5	75.0
MkNet	**93.5**	97.6	**78.9**	**85.7**	**92.9**	**85.8**	**77.8**	**88.6**	74.2	89.1	45.3	**75.9**	**85.6**	75.1	**85.7**	**93.0**	91.8	**77.8**	91.6	**71.2**	**82.9**
Ours	90.5	95.1	76.0	83.8	89.0	85.0	77.4	75.8	**74.7**	**89.7**	47.8	74.3	77.6	79.2	67.5	90.1	**93.9**	73.1	80.2	70.6	79.6

fused 2D-3D feature representation indeed contributes to a higher semantic segmentation accuracy, and the 2D semantic feature is more descriptive than the 3D geometric feature. This is reasonable since the 2D feature has been solidified for prediction through the SSMA [Valada et al. 2020], while the 3D feature is still raw.

Besides, to study the influence of SSMA, we have also implemented the SF and PsF using SSMA for 2D semantic prediction, termed as *SF-SSMA* and *PsF-SSMA*, respectively. As shown in Table 4, though SSMA helps improve the accuracy of SF and PsF, our Supervoxel-CNN still outperforms PsF-SSMA and SF-SSMA with

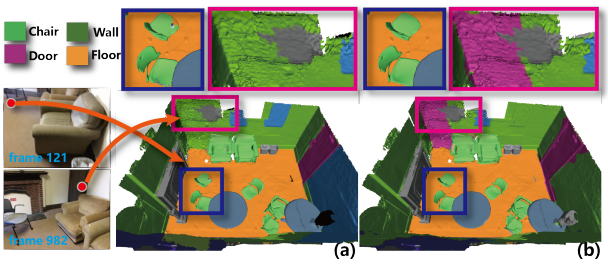

Fig. 8. Visual comparison of the segmentation results of scene616 (ScanNet v2) between our method (b) and MinkowskiNet (a) adapted for online segmentation. Note that MinkowskiNet makes wrong prediction of a partially observed chair (frame 121) and a door (frame 982) in the online task.

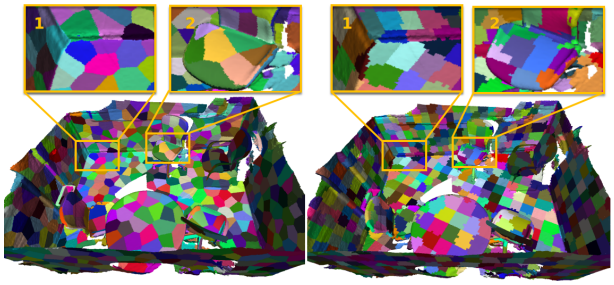

Fig. 9. Visual comparison between boundary-preserving supervoxel clustering (Left) and VCCS [Papon et al. 2013] (Right) on the scene0427 from the ScanNet v2 validation set. Note that our method better respects the underlying object boundaries.

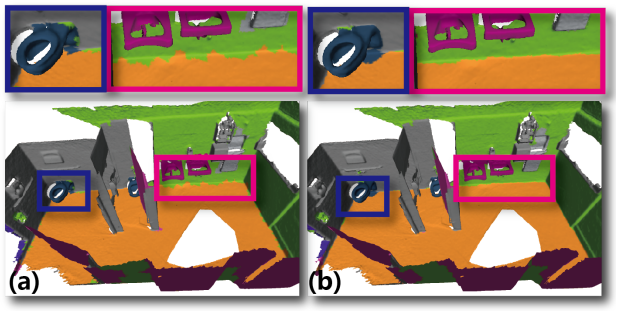

Fig. 10. Visual comparison between the semantic segmentation results of scene0609 of the ScanNet v2 validation set, using Supervoxel-CNN FULL (a) and Supervoxel-CNN w/o BP (b).

an improvement of 7.2% in mAcc and 6.9% in mIoU w.r.t. PsF-SSMA and even larger w.r.t SF-SSMA.

CRFasRNN Inference. We inspect how Supervoxel-CNN performs without CRFasRNN inference, referred to as "Supervoxel-CNN w/o CRFasRNN" in Table 4. CRFasRNN is a data-driven approach that leads to spatially consistent semantic prediction by reformulating the CRF inference as RNN inference in a deep leaning way [Zheng et al. 2015]. We thus adopt the CRFasRNN module to encourage more consistent semantic predictions between neighboring supervoxels. Although CRFasRNN inference does not significantly

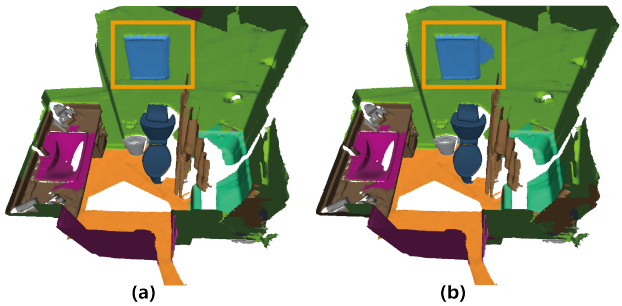

Fig. 11. Visual semantic segmentation results of Supervoxel-CNN for scene scene0406 from the ScanNet v2 validation set, with (a) or without (b) CRFasRNN inference.

Table 4. The 3D semantic segmentation accuracy (mAcc and mIoU) of different models (or methods) evaluated on the validation set of the Scannet v2 dataset. Please refer to our supplementary materials for the statistic details for both the mAcc and mIoU metrics.

Model	mAcc	mIoU
SF-SSMA	65.4	59.1
PsF-SSMA	68.6	61.4
PointConv-Uniform	75.1	65.4
Supervoxel-CNN w/o BP	75.3	65.7
Supervoxel-CNN w/o 3D	74.0	64.2
Supervoxel-CNN w/o 2D	70.1	62.4
Supervoxel-CNN w/o CRFasRNN	78.9	68.1
Supervoxel-CNN FULL	**79.6**	**68.3**

boost the semantic segmentation accuracy number, it makes the final results spatially more consistent, as shown in Fig. 11. Besides, we show more visual results in the supplementary materials.

Supervoxel Neighborhood. The number of supervoxel neighborhoods, i.e. K in Sec. 3.2, used in *SVConv* also influences the performance of our Supervoxel-CNN. To study how the number of supervoxel neighborhoods effects our Supervoxel-CNN, we train the Supervoxel-CNN on the Scannet v2 training set with different values of K for the *SVConv*, and evaluate the segmentation accuracy (mAcc and mIoU) on the Scannet v2 validation set. Fig. 12 shows the average mAcc and mIoU accuracy over the 312 scans of Scannet v2 validation set. In general, the segmentation accuracy increases with the increasing number of supervoxel neighborhoods when $K <= 10$ and doesn't take much effects when $K > 10$. So in all of our experiments we set $K = 10$ for best segmentation accuracy of our Supervoxel-CNN.

5.5 Time Analysis

Complexity Analysis. Our approach performs online 3D semantic segmentation following the clustering-then-prediction fashion. For supervoxel *clustering*, the time complexity is $O(Nlog(N))$ with N being the voxel number. Note that in our progressive clustering, the voxel number N in region U' (Sec. 3) is a relatively small number (around 4000 in our experiments). The time complexity of

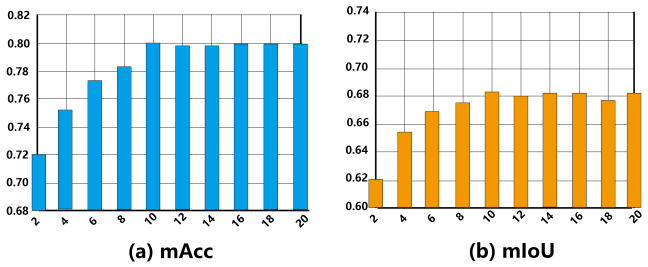

Fig. 12. The average mAcc (a) and mIoU (b) accuracy statistics of our Supervoxel-CNN with the respect to the number of supervoxel neighborhoods K, tested on the Scannet v2 validation set.

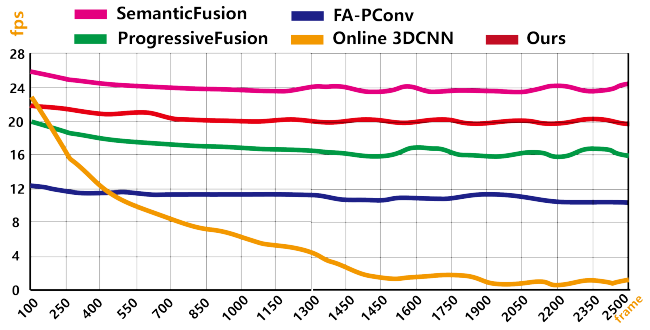

Fig. 13. The semantic prediction rate curves of different online methods. Here, the online 3DCNN is implemented using MinkowskiNet.

Supervoxel-CNN *prediction* is $O(M)$ with M being the supervoxel number. The supervoxel number M is also a small number in the latest on-surface region (Sec. 3). Besides, the supervoxel neighborhood searching during the *SVConv* of our Supervoxel-CNN is $O(1)$, which is simply performed by querying the neighborhood array stored for each supervoxel, benefiting from the connectivity structure after the supervoxel clustering.

System Timing. We evaluate the time efficiency between different online semantic segmentation methods, including SemanticFusion (SF), ProgressiveFusion (PsF), FA-PConv, and ours. Since the approaches of SF, PsF, and ours consist of both the 3D dense reconstruction component and semantic segmentation component, while FA-PConv only focus on the semantic segmentation component without a dense 3D reconstruction component, for a fair comparison, we focus only on the semantic prediction component for all the four systems. Specifically, we calculate the average semantic prediction rate when the number of frames increases during the 3D reconstruction on the Scannet v2 validation set.

As shown in Fig. 13, SF is the most efficient approach and achieves average 25fps prediction rate. PsF achieves average 18fps prediction rate and our approach achieves average 20fps prediction rate. Specifically, our progressive supervoxel clustering step takes average 18ms and Supervoxel-CNN prediction takes average 32ms only, since the number of supervoxels is small during the online task. Please note that the prediction time depends on multiple factors including the scene size, supervoxel size, etc. (see the evaluation in Sec. 5.6). FA-PConv achieves average 10fps prediction rate, which is slower than our approach partially due to the time-consuming neighborhood management scheme during their fusing-aware point convolution. Note that FA-PConv would slow down further if it is integrated into a dense 3D reconstruction system.

Compared to these approaches, our approach achieves comparable online prediction rate (near real-time) with SF (slightly faster than PsF) but with significantly better accuracy, and at least 2× faster prediction rate than FA-PConv with better segmentation results, which strikes the best balance between efficiency and accuracy. We also evaluate the processing time of an online 3DCNN approach (adopting MinkowskiNet to predict all voxel's semantic labels at each timestep). Please refer to the accompanying video for live demonstration of both the data from the ScanNet dataset and real shot scenes.

GPU Storage Usage. Our system consumes GPU memory storage mainly in three modules: the SSMA 2D semantic prediction module, the clustering-then-prediction 3D semantic prediction module, and the 3D reconstruction module. The SSMA module with default parameters setting costs about 1G GPU memory storage. In our 3D semantic prediction module, the *clustering* (including the voxel tracking, progressive supervoxel clustering, etc.) costs about 0.7G GPU memory storage, and Supervoxel-CNN *prediction* costs about 1G GPU memory storage. The 3D reconstruction module consumes the most, and uses about 10G GPU memory storage with a voxel size of 0.01m.

5.6 Timing vs Accuracy Evaluation

The semantic prediction timing and segmentation accuracy in our approach would be mainly influenced by two factors: (1) the supervoxel size R, which plays a crucial role in both the *clustering* and *prediction* stages and directly influences the prediction timing and segmentation accuracy. Note that supervoxels are clustered into a similar pattern once the supervoxel size is configured [Lin et al. 2018]; (2) the scene size, which affects the CPU/GPU computation and storage for the entire system, thus influencing the prediction timing and segmentation accuracy indirectly. To evaluate how our approach behaves under different supervoxel sizes and scene sizes in terms of the prediction timing and segmentation accuracy, we perform another evaluation of our system on the ScanNet v2 validation set.

For the ScanNet v2 validation set, we found the floor area of all the 312 indoor scenes varies from about $3m^2$ to $60m^2$, with intervals $[21m^2, 60m^2]$, $[13m^2, 21m^2]$ and $[3m^2, 13m^2]$ each containing about $\frac{1}{3}$ of the scenes. We thus randomly select 10 scenes for each of the floor area intervals, and mark them as 'Large' scenes, 'Moderate' scenes, and 'Small' scenes, respectively. For all the 30 selected scenes, we test our full system with supervoxel size R varying from 0.1m to 0.5m (a step of 0.05m), and compute the corresponding semantic prediction rate (in fps) and segmentation accuracy (in mAcc). For comparison, we also test the SemanticFusion and ProgressiveFusion systems on these scenes with the same supervoxel size configurations. Note that supervoxel size is a parameter for supervoxel clustering [Papon et al. 2013], we also use it to control the clustering

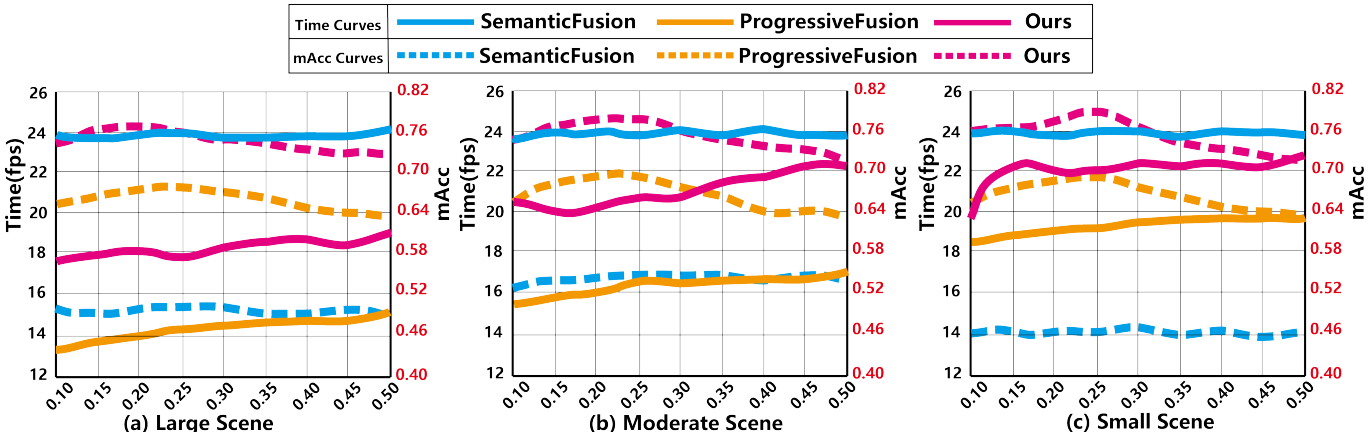

Fig. 14. The average semantic prediction rate and segmentation accuracy curves for large scenes (a), moderate scenes (b) and small scenes (c) with different supervoxel's sizes (horizontal axis) respectively, by three online semantic segmentation methods (SemanticFusion, ProgressiveFusion and ours) evaluated on the ScanNet v2 dataset. Please refer to the supplementary material for the detailed timing and accuracy curves of each individual scene.

in ProgressiveFusion though their intermediate clustering results would not guarantee boundary-preserving like ours.

Fig. 14 shows the average semantic prediction rates and segmentation accuracy values for Large, Moderate and Small scenes, with respect to different supervoxel sizes. Our approach's semantic prediction rate increases slightly with the increasing superovoxel size R (about 2fps increment from 0.1m to 0.5m). This makes sense since when the supervoxel size become larger, the total number of supervoxels to be fed to our system becomes smaller, thus making the overall system faster. For the segmentation accuracy, the mAcc first increases with the supervoxel size varying from 0.1m to 0.3m and then decreases from 0.3m to 0.5m (with peak at around 0.25m). The scene size mainly influences the semantic prediction rate, with 18-20fps in large scenes, 20-22fps in moderate and small scenes, but has little influence on the segmentation accuracy. This is partially due to that large CPU/GPU computation and storage in large scenes will slow down the performance of our supervoxel clustering and Supervoxel-CNN prediction.

To strike a trade-off between efficiency and accuracy, we set the supervoxel size as 0.25m with the highest segmentation accuracy and around 20fps average semantic prediction rate of all scene size. For SemanticFusion, the different supervoxel size does not affect the semantic prediction rate or the segmentation accuracy, since SemanticFusion does not adopt supervoxel-based prediction. Note that there are small variations in the prediction rate and accuracy curves of SemanticFusion due to multiple tests. ProgressiveFusion behaves similarly to our approach but has a lower prediction rate and segmentation accuracy.

5.7 Limitations and Discussion

A main limitation of our approach is that we could not correct wrong 2D semantic labels from the 2D CNN. As shown in Fig. 15, since SSMA wrongly detects a monitor (marked '1' in the figure) as part of a wall, our approach fails to correct it. Another limitation is that our approach could not segment small objects very well, partially due to the unbalanced 3D annotation dataset for training. A small object would be wrongly assigned as a semantic label of its neighboring region or others (e.g., the three small objects marked '2' in the figure are labeled wrongly). One feasible solution to overcome these limitations might be to design a more effective end-to-end 3D-to-3D semantic mapping deep neural network trained on balanced a sufficiently annotated 3D dataset. Besides, the 3D reconstruction quality would influence the final semantic segmentation results, though 3D reconstruction quality is not the focus of our approach.

We assume that the reconstruction system can provide reliable camera poses for the online task. The challenging cases raised by camera tracking lost, re-localization or loop closure are not seriously explored in our approach yet. Our Supervoxel-CNN performs the 2D-3D joint learning by aggregating the 2D-3D features only. However, we have not explored how to improve the 2D feature learning (such as the SSMA module), thus enhancing the 2D-3D feature aggregation in turn. Our current implementation determines the final prediction for each supervoxel based on the latest prediction by our Supervoxel-CNN, which might lead to gradually non-stabilized predictions. A more robust solution might be to determine the label of each supervoxel based on the last N predictions for temporally more consistent predictions. Besides, the performance of 2D CNNs would be effected by camera views [Kundu et al. 2020] thus influencing our 2D-3D feature aggregation. This could be improved by introducing a view consistent constraint to the 2D-3D feature aggregation for better 3D segmentation consistency, which is not explored in our current solution yet. Lastly, a comprehensive re-training with both synthetic and real-world data for the 2D CNN module and Supervoxel-CNN could further improve the generalization capability of our approach, though our preliminary evaluation found that re-training Supervoxl-CNN on the ScanNet v2 dataset augmented with the synthetic data from SceneNetRGB-D [McCormac et al. 2017b] did not lead to significant improvement in terms of segmentation accuracy. We think these above mentioned points are interesting points to explore in the future to improve our system's robustness and segmentation accuracy.

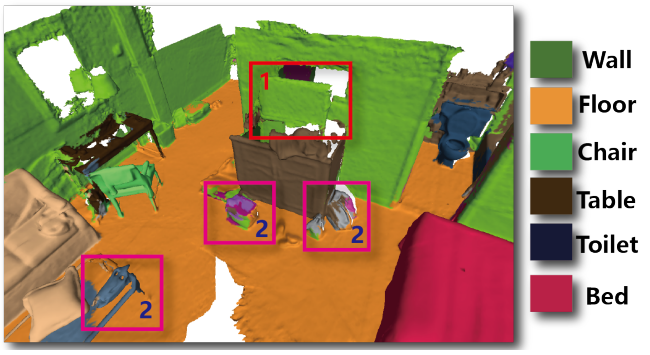

Fig. 15. A failure case of our approach evaluated on scene0645 of the ScanNet v2 dataset.

6 CONCLUSION

In this paper, we have introduced *SVConv*, the first attempt to perform convolution on supervoxels for efficient and effective learning on a dense 3D representation. Based on SVConv, we present a supervoxel-based deep neural network with fully convolution layers, i.e. *Supervoxel-CNN*, and propose a *clustering-then-prediction* online dense 3D semantic segmentation approach based on the *Supervoxel-CNN*, which transforms the complicated 2D-to-3D semantic mapping for the dense voxels to a novel, learnable and effective 3D semantic prediction with joint 2D-3D learning on progressively clustered supervoxels. Our approach outperforms other online 3D semantic segmentation methods, achieving the state-of-the-art semantic segmentation results. We hope that our work can inspire more efficient solutions using deep neural networks for the important 3D semantic mapping task in computer graphics, computer vision, and robotics communities. In the future, we would like to apply SVConv for other 3D geometry learning tasks like semantic instance segmentation [Hou et al. 2019].

7 ACKNOWLEDGEMENT

We would like to thank all the anonymous reviewers for their constructive suggestions. This work was supported by grants from the China Postdoctoral Science Foundation (Grant No.: 2019M660646), Natural Science Foundation of China (Grant No.: 61902210, 61521002), Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11208721), City University of Hong Kong (Project No. 7005590), and the Centre for Applied Computing and Interactive Media (ACIM) of School of Creative Media, CityU.

REFERENCES

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. 2019a. Scan2CAD: Learning CAD Model Alignment in RGB-D Scans. In *IEEE CVPR*. 2614–2623.

Armen Avetisyan, Angela Dai, and Matthias Nießner. 2019b. End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans. In *IEEE ICCV*. 2551–2560.

Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. 2020. SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans. In *ECCV*.

Yan-Pei Cao, Leif Kobbelt, and Shi-Min Hu. 2018. Real-time High-accuracy Three-Dimensional Reconstruction with Consumer RGB-D Cameras. *ACM Trans. Graph.* 37, 5 (2018), 171:1–171:16.

Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. 2020. ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. In *ECCV*.

Jiawen Chen, Dennis Bautembach, and Shahram Izadi. 2013. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.* 32, 4 (2013), 113:1–113:16.

Christopher B. Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE CVPR*. 3075–3084.

Brian Curless and Marc Levoy. 1996. A Volumetric Method for Building Complex Models from Range Images. In *ACM SIGGRAPH*. 303–312.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. 2017a. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *IEEE CVPR*. 2432–2443.

Angela Dai and Matthias Nießner. 2018. 3DMV: Joint 3D-Multi-view Prediction for 3D Semantic Scene Segmentation. In *ECCV (Lecture Notes in Computer Science, Vol. 11214)*. 458–474.

Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017b. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. *ACM Trans. Graph.* 36, 3 (2017), 24:1–24:18.

Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. 2018. ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. In *IEEE CVPR*. 4578–4587.

Siyan Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. 2019. Multi-robot collaborative dense scene reconstruction. *ACM Trans. Graph.* 38, 4 (2019), 84:1–84:16.

Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 2018. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *IEEE CVPR*. 9224–9232.

Lei Han, Tian Zheng, Lan Xu, and Lu Fang. 2020. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *IEEE CVPR*. 2937–2946.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: a network with an edge. *ACM Trans. Graph.* 38, 4 (2019), 90:1–90:12.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *IEEE ICCV*. 2980–2988.

Ji Hou, Angela Dai, and Matthias Nießner. 2019. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *IEEE CVPR*. 4421–4430.

Ruizhen Hu, Cheng Wen, Oliver Van Kaick, Luanmin Chen, Di Lin, Daniel Cohen-Or, and Hui Huang. 2018. Semantic Object Reconstruction via Casual Handheld Scanning. *ACM Trans. Graph.* 37, 6 (2018), 219:1–219:12.

Shi-Min Hu, Junxiong Cai, and Yu-Kun Lai. 2020. Semantic Labeling and Instance Segmentation of 3D Point Clouds Using Patch Context Analysis and Multiscale Processing. *IEEE TVCG* 26, 7 (2020), 2485–2498.

Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. 2016. SceneNN: A Scene Meshes Dataset with aNNotations. In *3DV*. 92–101.

Junho Jeon, Jinwoong Jung, Jungeon Kim, and Seungyong Lee. 2018. Semantic Reconstruction: Reconstruction of Semantically Segmented 3D Meshes via Volumetric Semantic Fusion. *Comput. Graph. Forum* 37, 7 (2018), 25–35.

Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip H. S. Torr, and David William Murray. 2015. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE TVCG* 21, 11 (2015), 1241–1250.

Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David A. Ross, Brian Brewington, Thomas A. Funkhouser, and Caroline Pantofaru. 2020. Virtual Multi-view Fusion for 3D Semantic Segmentation. In *ECCV*.

Yangbin Lin, Cheng Wang, Dawei Zhai, Wei Li, and Jonathan Li. 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS* 143 (2018), 39 – 47.

Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. 2020. FPConv: Learning Local Flattening for Point Convolution. In *IEEE CVPR*. 4292–4301.

Ligang Liu, Xi Xia, Han Sun, Qi Shen, Juzhan Xu, Bin Chen, Hui Huang, and Kai Xu. 2018. Object-aware guidance for autonomous scene reconstruction. *ACM Trans. Graph.* 37, 4 (2018), 104:1–104:12.

John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. 2018. Fusion++: Volumetric Object-Level SLAM. In *3DV*. 32–41.

John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. 2017a. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *IEEE ICRA*. 4628–4635.

John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. 2017b. SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?. In *IEEE ICCV*.

Liangliang Nan, Ke Xie, and Andrei Sharf. 2012. A *search-classify* approach for cluttered indoor scene understanding. *ACM Trans. Graph.* 31, 6 (2012), 137:1–137:10.

Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. 2019. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *IEEE IROS*. 4205–4212.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*. 127–136.

Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph.* 32, 6 (2013), 169:1–169:11.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning Deconvolution Network for Semantic Segmentation. In *IEEE ICCV*. 1520–1528.

Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Wörgötter. 2013. Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds. In *IEEE CVPR*. 2027–2034.

Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. 2019. Real-Time Progressive 3D Semantic Segmentation for Indoor Scenes. In *IEEE WACV*. 1089–1098.

Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. 2018. Frustum PointNets for 3D Object Detection From RGB-D Data. In *IEEE CVPR*. 918–927.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE CVPR*. 77–85.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NIPS*. 5099–5108.

Nima Sedaghat, Mohammadreza Zolfaghari, Ehsan Amiri, and Thomas Brox. 2017. Orientation-boosted Voxel Nets for 3D Object Recognition. In *BMVC*.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE T-PAMI* 39, 4 (2017), 640–651.

Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. 2017. Semantic Scene Completion from a Single Depth Image. In *IEEE CVPR*. 190–198.

Duc Thanh Nguyen, Binh-Son Hua, Lap-Fai Yu, and Sai-Kit Yeung. 2017. A Robust 3D-2D Interactive Tool for Scene Segmentation and Annotation. *IEEE TVCG* (2017).

Abhinav Valada, Rohit Mohan, and Wolfram Burgard. 2020. Self-Supervised Model Adaptation for Multimodal Semantic Segmentation. *Int. J. Comput. Vis.* 128, 5 (2020), 1239–1285.

Julien P. C. Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Nießner, Antonio Criminisi, Shahram Izadi, and Philip H. S. Torr. 2015. SemanticPaint: Interactive 3D Labeling and Learning at your Fingertips. *ACM Trans. Graph.* 34, 5 (2015), 154:1–154:17.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. 2019. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *IEEE ICCV*. 7657–7666.

Chao Wang and Xiaohu Guo. 2017. Feature-based RGB-D camera pose optimization for real-time 3D reconstruction. *Comput. Vis. Media* 3, 2 (2017), 95–106.

Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice F. Fallon, John J. Leonard, and John McDonald. 2015. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *I. J. Robotics Res.* 34, 4-5 (2015), 598–626.

Wenxuan Wu, Zhongang Qi, and Fuxin Li. 2019. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *IEEE CVPR*. 9621–9630.

Kai Xu, Yifei Shi, Lintao Zheng, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 2016. 3D attention-driven depth acquisition for object identification. *ACM Trans. Graph.* 35, 6 (2016), 238:1–238:14.

Sheng Yang, Zheng-Fei Kuang, Yan-Pei Cao, Yu-Kun Lai, and Shi-Min Hu. 2019. Probabilistic Projective Association and Semantic Guided Relocalization for Dense Reconstruction. In *IEEE ICRA*. 7130–7136.

Sheng Yang, Jie Xu, Kang Chen, and Hongbo Fu. 2017. View suggestion for interactive segmentation of indoor scenes. *Comput. Vis. Media* 3, 2 (2017), 131–146.

Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. 2020. Fusion-Aware Point Convolution for Online Semantic 3D Scene Segmentation. In *IEEE CVPR*.

Yizhong Zhang, Weiwei Xu, Yiying Tong, and Kun Zhou. 2015. Online Structure Analysis for Real-Time Indoor Scene Reconstruction. *ACM Trans. Graph.* 34, 5 (2015), 159:1–159:13.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3D Point Capsule Networks. In *IEEE CVPR*. 1009–1018.

Lintao Zheng, Chenyang Zhu, Jiazhao Zhang, Hang Zhao, Hui Huang, Matthias Nießner, and Kai Xu. 2019. Active Scene Understanding via Online Semantic Reconstruction. *Comput. Graph. Forum* 38, 7 (2019), 103–114.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional Random Fields as Recurrent Neural Networks. In *IEEE ICCV*. 1529–1537.