

# ALSTER: A Local Spatio-Temporal Expert for Online 3D Semantic Reconstruction

Silvan Weder<sup>1</sup>

Francis Engelmann<sup>1,4</sup>

Johannes L. Schönberger<sup>2</sup>

Akihito Seki<sup>3</sup>

Marc Pollefeys<sup>1,2</sup>

Martin R. Oswald<sup>1,4</sup>

<sup>1</sup> ETH Zurich   <sup>2</sup> Microsoft   <sup>3</sup> Toshiba   <sup>4</sup> Google   <sup>4</sup> University of Amsterdam

## Abstract

We propose an online 3D semantic segmentation method that incrementally reconstructs a 3D semantic map from a stream of RGB-D frames. Unlike offline methods, ours is directly applicable to scenarios with real-time constraints, such as robotics or mixed reality. To overcome the inherent challenges of online methods, we make two main contributions. First, to effectively extract information from the input RGB-D video stream, we jointly estimate geometry and semantic labels per frame in 3D. A key focus of our approach is to reason about semantic entities both in the 2D input and the local 3D domain to leverage differences in spatial context and network architectures. Our method predicts 2D features using an off-the-shelf segmentation network. The extracted 2D features are refined by a lightweight 3D network to enable reasoning about the local 3D structure. Second, to efficiently deal with an infinite stream of input RGB-D frames, a subsequent network serves as a temporal expert predicting the incremental scene updates by leveraging 2D, 3D, and past information in a learned manner. These updates are then integrated into a global scene representation. Using these main contributions, our method can enable scenarios with real-time constraints and can scale to arbitrary scene sizes by processing and updating the scene only in a local region defined by the new measurement. Our experiments demonstrate improved results compared to existing online methods that purely operate in local regions and show that complementary sources of information can boost the performance. We provide a thorough ablation study on the benefits of different architectural as well as algorithmic design decisions. Our method yields competitive results on the popular ScanNet benchmark and SceneNN dataset.

## 1. Introduction

Humans not only require low-level spatial awareness of their surroundings to interact with the world, but also rely on a higher-level semantic understanding of its contents in

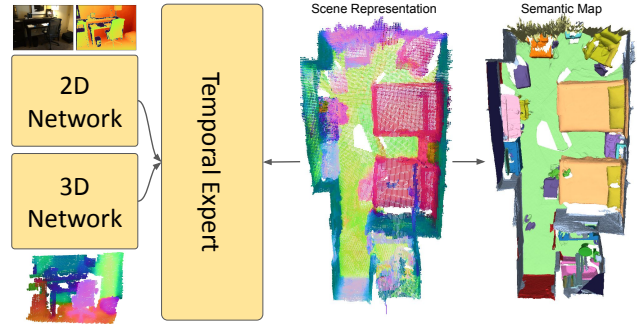


Figure 1. We propose an online semantic 3D reconstruction pipeline, which fuses RGB-D observations into a globally consistent semantic map. The key component is a local spatio-temporal expert network that fuses new observations into a learned scene representation. This temporal expert learns to select information from 2D, 3D, and previous steps using an attention mechanism.

real-time. In order to enable autonomous agents with similar capabilities, we aspire to model these processes using computational methods. Building spatial awareness through 3D reconstruction has been a long-standing topic in computer vision. More recently, and given the success in spatial reconstruction, the challenge of understanding the world on a higher-level has been approached.

One can understand the world on different levels – from low-level reconstruction of 3D geometric primitives to high-level scene classification. While high-level understanding (e.g., scene classification) is usually sufficient for planning and navigation, interaction also requires a fine-grain understanding with accurate semantic boundaries. As such, semantic segmentation is oftentimes at the heart of algorithms and pipelines that interact with the world, in which most approaches assign a discrete semantic class label to each reconstructed point in the scene.

In the past, there has been a large number of works in 3D scene understanding which take a point-cloud, a mesh, or a voxel-grid as input, and estimate a per point semantic label [3, 9, 17, 25]. While there has been impressive progress

with these works in recent years, a large majority of these have one major shortcoming that we aim to address. All these works require an a priori reconstruction of the scene and use this global information for the understanding task. Therefore, they are considered offline methods. However, autonomous agents (as well as humans) typically build a “mental” map of the environment in an incremental manner, that is, continuously update it over time as new information is collected. Thus, scene understanding must inherently be an iterative process, as an autonomous agent cannot assume all information known a priori. In this work, we investigate this particular problem, where we incrementally build a semantic map given a stream of posed RGB-D data that allows for online processing of the incoming data streams and can be integrated in real-time systems. This online processing is essential for enabling real-world applications, such as robotics and mixed reality, where an updated semantic map is required to solve complex tasks in the world.

Only few approaches tackle the problem specified above. The seminal works of Vineet *et al.* [33] and SemanticFusion [20] map 2D semantic predictions into 3D. One step further, PanopticFusion [24] predicts a semantic instance map in 2D that is mapped and aggregated in 3D. While these methods reason in 2D as well as 3D, the 3D reasoning is a CRF-based regularizer that is limited compared to modern neural networks. Further, the CRF requires global information of the entire scene that limits the scalability of the methods to small scenes. Similarly, INS-Conv [19] estimates semantic instance maps using 3D processing with a large UNet that requires global processing to avoid drifting errors. In contrast, other works [15, 37] perform 3D reasoning using point- or supervoxel convolutions in a local frame. However, they only store explicit labels that only encode per-point information, while our work uses learned features encoding low-, mid- and high-level information.

Our work is based on the observation that 2D and 3D information is complementary for the task of scene understanding. Some elements are better to be understood in 2D depending on context and geometry whereas others are easier to be segmented in 3D given their spatial structure. To this end, we present a novel attention-based aggregation mechanism that fuses 2D, 3D, as well as existing features into the scene. Our method only operates in a local region defined by the new measurement and integrates the updates into the learned global scene representation. Through this design, our method is independent of the scene size and can scale to large-scale scenes.

In an extensive experimental evaluation, we show that our method is competitive with existing approaches to online semantic 3D reconstruction while not requiring passes over the entire reconstruction as opposed to some other methods [19]. This is particularly important for online processing on mobile devices and agents that are constrained in

the amount of compute and memory available. We evaluate our method on ScanNet as well as SceneNN and present in-depth ablation studies to motivate our design choices. We will release the source code on acceptance of this paper to foster further research in this direction. In summary, the key contributions in this work are:

- We show that 2D and 3D information are complementary for the task of online 3D semantic reconstruction and improve the overall result.
- We propose a novel local fusion approach that leverages an attention mechanism to combine existing features with new 2D and 3D information in an online fashion. We evaluate our pipeline design on the well-known ScanNet [5] benchmark and show competitive results compared to existing online local reconstruction methods.

## 2. Related Work

**Offline vs. online processing.** Most existing 3D segmentation methods follow on offline approach: the 3D geometry of the scene and its corresponding features (color, normals, *etc.*) are known a-priori and then processed by the segmentation method. We first review prior work on *offline* semantic segmentation and then look at existing *online* methods in the context of incrementally building semantic 3D maps. **3D Semantic Segmentation** is the problem of assigning a class label to each point, voxel, or vertex of the 3D scene. It is central to many applications and pipelines that require some form of understanding. In recent years, many different methods tackled this problem. Semantic Stixels [31] predict 2D semantic labeling and stereo depth maps that are aggregated in a 3D stixel representation. While this representation can be sufficient for outdoor applications, it lacks representation power for indoor applications. Kundu *et al.* [17] address the problem of lack of context in the 2D views by rendering views from an already reconstructed mesh to have a larger field-of-view that improves the performance of 2D semantic segmentation. The predictions are afterwards aggregated again on the 3D mesh. As this approach is dependent on an already reconstructed mesh, it is not suitable for an online fusion approach. Atlas [23] jointly reconstructs a semantic and geometric map from visual inputs by learning multi-view fusion. As this approach needs to aggregate dense viewing frustums to solve the multi-view stereo problem, it is not suitable for fast online updates. SemanticNeRF [39] proposes the application of recently proposed neural radiance fields [22] to the problem of 3D semantic segmentation. While this approach shows impressive results, it is also not applicable to fast and accurate online updates of a semantic map. Mix3D [25] boosts the performance of 3D segmentation methods by proposing a novel data augmentation technique that combines different scenes. While this augmentation works for global methods, it cannot be applied to online fusion systems since we jointly learn the fu-

sion across time and segmentation of the scene. BPNet [12] couples 2D and 3D predictions of scene labels by proposing a bi-directional projection module. This boosts the performance on 3D semantic segmentation but is dependent on global processing and a-prior scene reconstructions. VM-Net [13] combines Euclidean and geodesic information to address the short-comings of voxel-only approaches. Yet, it also requires global processing to unfold its full potential. OccuSeg [9] enhances supervoxel-based geometric segmentation with learned features and refines them using graph-based clustering but requires a global receptive field, which makes them unnecessarily expensive for online processing.

**Online 3D Semantic Segmentation.** In contrast to the previously mentioned approaches, online methods iteratively process the scene making them better suitable to real-time applications, where agents are interacting with their environment such as robotics or mixed reality in unknown environments. There is a long line of work aiming at the real-time reconstruction of geometry and appearance [4, 26, 35, 36]. These works have been extended to scene understanding to enable agents with understanding capabilities. The approaches [33] and [20] proposed to fuse 2D semantic predictions into a global semantic map that is refined using a conditional random field (CRF). This idea has been extended by several works. SceneCode [38] stores a per-keyframe latent code encoding the semantic information of the scene that is optimized at test time. Meanwhile, MaskFusion [30] and Fusion++ [21] focus on 3D object segmentation while ignoring their semantic class. ProgressiveFusion [27] improves efficiency by clustering voxels into supervoxels and apply CRF on that level. SemanticReconstruction [16] follows a similar approach as [20], but shows that their scene representation can be used for downstream tasks such as scene completion and manipulation. PanopticFusion [24] estimates 3D semantic instance maps by predicting 2D semantic and instance segmentation using off-the-shelf networks, aggregates them in 3D, and also regularizes them using a CRF. While these works leverage 2D processing in combination with optimization-based 3D regularization, they all resort to traditional voxel fusion and do not utilize trainable 3D neural networks. This shortcoming has been addressed in SVCNN [15], which clusters voxels that store explicit semantic information into supervoxels, and then processes them using a special convolutional operator designed for supervoxels. However, [15] still resorts to an explicit fusion of 2D semantic information into voxels. An alternative is FusionAware [37] that represents scenes using efficient point cloud representations and uses point-convolutions to aggregate new information. More recently, Liu *et al.* [19] presented an online method predicting semantic instance maps using only 3D processing. Nevertheless, these two works disregard useful 2D information. In our work, we address these limitations by a) combining 2D

and 3D information in a temporal expert network leveraging both sources of information, and b) applying a powerful yet lightweight 3D network on the current viewing frustum.

### 3. Method

This section presents our method for online 3D semantic reconstruction. Firstly, we give an overview of our model (Fig. 2) and the 3D scene representation. Then, we describe the spatial-temporal expert that enables efficient local updates of the scene representation. Lastly, we discuss training protocols, loss functions and sequential optimization.

#### 3.1. Semantic 3D Reconstruction Pipeline

**Overview.** Our proposed model consists of three major components (Fig. 2) and a learned 3D scene representation. The first stage is a 2D encoder  $F^{2D}$  that extracts 2D feature maps from an incoming stream of RGB-D images. The second stage is a 3D encoder  $F^{3D}$  that incorporates 3D geometry into each feature map after lifting it to 3D using the given camera parameters and depth maps. The third stage is a new *temporal expert* network  $F^{\Delta T}$  that consolidates 3D scene representations using complementary information from 2D and 3D as well as the so-far reconstructed 3D scene.

**Scene Representation.** The backbone of every online reconstruction method is a suitable scene representation. Typically, the primary choice are voxels, points, meshes or implicit (neural) representation. Meshes and implicit representations are difficult to update with new observations, while points lack information about geometric connectivity. This is crucial for scene understanding where decisions about segmentation boundaries are oftentimes guided by geometric boundaries. Therefore, we represent scenes using a hybrid representation  $S$  combining learned and explicit features that are stored in a sparse voxel grid. Sparse voxel grids allow for efficient processing using neural networks. In particular, each voxel stores a learned feature  $F$  of dimension  $D_F = 40$  encoding the aggregated information about the scene content. This learned scene representation allows to store high-, mid-, and low-level information useful for the semantic segmentation task. This mitigates the need for expensive re-processing in deep neural networks at every time step to fuse existing and new information. The voxels also store the number of per-voxel observations, which is relevant for the subsequent fusion step.

**2D Encoder.** The aim of the first stage is to extract semantic features from incoming 2D RGB-D images using a 2D convolutional network  $F^{2D}$ . The 2D network  $F^{2D}$  with trainable parameters  $\theta^{2D}$  takes RGB-D frames  $(I_t, D_t)$  as input and predicts semantic features  $\tilde{f}_t^{2D}$  per frame:

$$\tilde{f}_t^{2D} = F^{2D}([I_t, D_t, N_t]; \theta^{2D}) \quad (1)$$

The normal map  $N_t$  is estimated from the depth map  $D_t$  and serves as additional input. The network consists

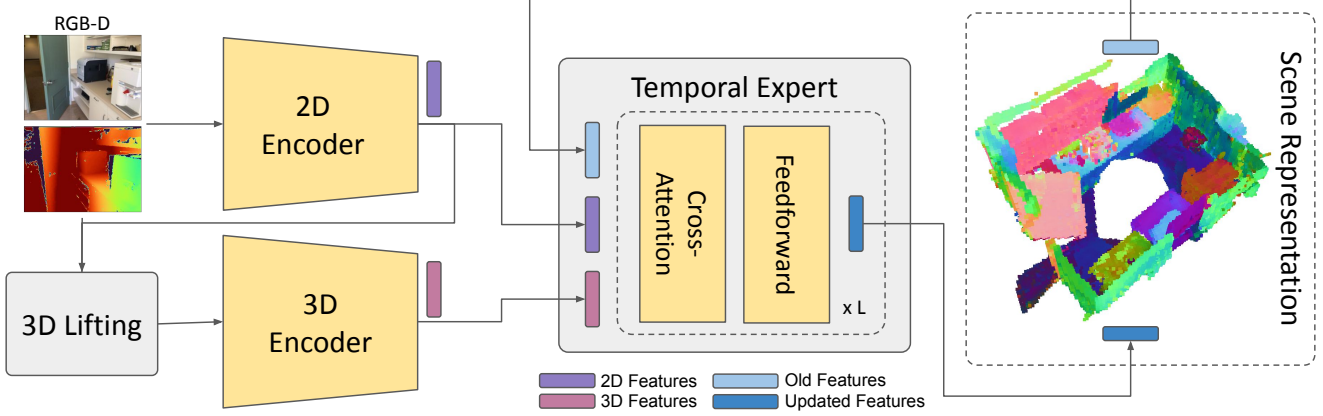


Figure 2. **Pipeline Overview:** Our pipeline consists of three main stages. The 2D encoder extracts information from incoming RGB-D imagery. This information is enhanced with 3D information using a light-weight 3D encoder. The information from these two sources is combined with existing information in the learned scene representation using the temporal expert.

of DeepLabV3+ [1] and uses, similar to [17], an Xception65 [2] encoder that is adjusted to handle RGB-D-N input data (color, depth, normals) since geometric information improves semantic segmentation [8, 10, 34]. The semantic features  $\tilde{f}_t^{2D}$  are directly obtained from the DeepLab decoder. However, the original dimension of the features  $\tilde{f}_t^{2D}$  is  $D_{\tilde{f}^{2D}} = 256$  which is too memory intensive for on-line processing of large indoor scenes. Instead, we project the feature maps to  $D^{2D} = 40$ . This compression allows on-line processing for the remaining of the pipeline while still retaining all relevant information needed for 3D semantic understanding. The 2D network  $F^{2D}$  is pre-trained on ImageNet [7] and fine-tuned on 2D training data from ScanNet [5]. During the training of the full pipeline, the 2D encoder is partially fine-tuned using an auxiliary semantic segmentation head enforcing consistent performance across frames and for regularization of the joint feature space.

**3D Encoder.** We additionally process the incoming information in 3D as geometry is complementary to 2D appearance. This processing is particularly motivated by the possible reasoning about hidden geometric object boundaries occluded in the current 2D frame. To this end, we lift the obtained 2D feature map  $f_t^{2D}$  to 3D point clouds by projecting the depth map  $D_t$  using the known, gravity-aligned camera orientation  $R \in SO(3)$  and intrinsics. Due to noisy depth estimates, we additionally filter out points that are more than 3 m away from the camera. The resulting *local* 3D feature volume is refined using a light-weight U-Net [29]  $F^{3D}$  yielding a 3D feature map  $f_t^{3D}$  with the same feature dimension  $D_F = 40$  as the 2D feature map  $f_t^{2D}$ .

**Spatio-Temporal Expert.** In the previous stages, 2D features are extracted and enhanced with 3D information. In the next step, this information is integrated into the existing global scene representation  $S_{t-1}$ . To this end, we propose the spatio-temporal expert network  $F^{\Delta T}$  with weights

$\theta^{\Delta T}$ . The task of the spatio-temporal expert network  $F^{\Delta T}$  is to update the features stored in the learned scene representation given the new information from  $f_t^{2D}$ ,  $f_t^{3D}$ , and the existing information  $f_{t-1}^{global}$ . The feature volume  $f_{t-1}^{global}$  is a crop of the relevant local sub-volume from the global scene representation  $S_{t-1}$  using the known camera pose  $[R|t] \in SE(3)$ . The overall mapping is computed as:

$$f_t^{global} = F^{\Delta T} \left( \left[ f_{t-1}^{global}, f_t^{3D}, f_t^{2D} \right]; \theta^{\Delta T} \right) \quad (2)$$

The resulting local volume  $f_t^{global}$  is then written back, using the inverse camera pose, to obtain the new global scene representation  $S_t$ . By providing access to both the 2D and 3D features in parallel, the expert network can learn where it is beneficial to rely more on 2D appearance features or where it is advantageous to trust the 3D geometry-based features, see Fig. 4 for an illustration. The 3D features reveal geometrical details while the 2D features provide textural information in flat areas with little geometric information.

$F^{\Delta T}$  is implemented as a Transformer consisting of cross-attention and feed-forward layers (see Fig. 3). The task of the cross-attention layer is to extract relevant information from the three sources of information ( $f_t^{2D}$ ,  $f_t^{3D}$ ,  $f_{t-1}^{global}$ ) using  $f_{t-1}^{global}$  as query features. The attention is defined as  $f = w_{f_{t-1}^{global}} v_{f_{t-1}^{global}} + w_{f_t^{2D}} v_{f_t^{2D}} + w_{f_t^{3D}} v_{f_t^{3D}}$  and the weights ( $w_{f_{t-1}^{global}}, w_{f_t^{2D}}, w_{f_t^{3D}}$ ) are defined as:

$$w_{f^k} = Q \left( f_{t-1}^{global} \right)^T K \left( f^k \right), \quad (3)$$

where  $f^k \in \{f_{t-1}^{global}, f_t^{3D}, f_t^{2D}\}$ . The values  $v_{f^k}$  are obtained using a linear projection layer  $v_{f^k} = V(f^k)$ .

The features extracted from the cross-attention layer are first normalized using layer norm and then refined using a standard feed-forward layer. Furthermore, both layers



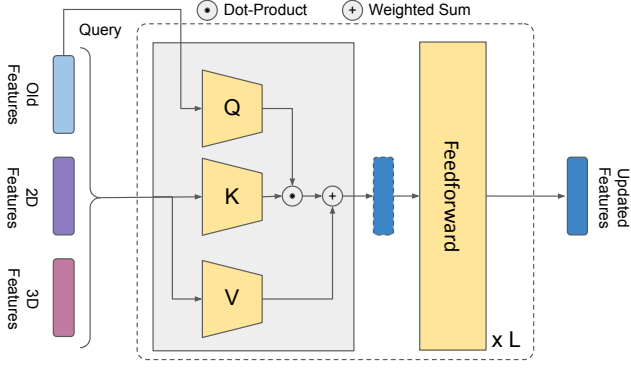


Figure 3. **Temporal expert network.** The temporal expert network takes the three features ( $f_{t-1}^{\text{global}}$ , 2D, and 3D) as input, and iteratively refines the old feature vector to obtain the update feature that can be stored in the scene representation. The old feature is used as the query in the attention mechanism.

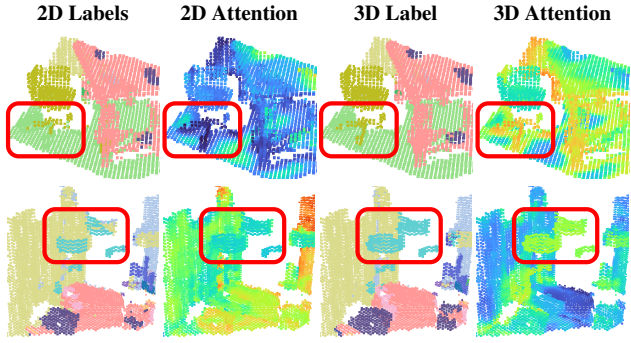


Figure 4. **Temporal expert attention maps.** We visualize the attention maps for the 2D and 3D input features in the expert network. The expert attends to the 3D features to refine the segmentation of fine details (legs of the chair, lamps), while it attends to the 2D features to predict the label for large areas (table, walls, etc.).

are augmented with a skip connection guaranteeing healthy gradients during training. The refined features that consist of information extracted from the three sources  $f_t^{2D}$ ,  $f_t^{3D}$ , and  $f_{t-1}^{\text{global}}$  are written back into the global scene representation. The temporal expert is also supervised by a point-wise segmentation loss, ensuring optimal segmentation given the currently available 3D scene information.

**Loss Function.** The pipeline is trained using focal loss [18] at several stages in the pipeline. These losses are applied after the 2D encoder  $F^{2D}$ , the 3D encoder  $F^{3D}$ , and the temporal expert network  $F^{\Delta T}$ . These auxiliary supervision signals are required to constrain the feature space that encodes the information throughout the entire pipeline. Further, these auxiliary losses ensure that each stage solves the task of semantic segmentation as good as possible for themselves providing the temporal expert network with valuable information. The overall loss is the sum of these losses:

$$\mathcal{L} = \lambda_{2D}\mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} + \lambda_{\text{Expert}}\mathcal{L}_{\text{Expert}} \quad (4)$$

where each term  $\mathcal{L}_{2D}$ ,  $\mathcal{L}_{3D}$ ,  $\mathcal{L}_{\text{Expert}}$  is a focal loss, defined as:

$$\mathcal{L} = (1 - \hat{y})^\gamma \text{CE}(\hat{y}, y) \quad (5)$$

As the loss is applied on a per-voxel level, the corresponding ground-truth labels first need to be mapped from the ground truth polygon mesh to a voxelized representation. To this end, we first voxelize all scenes using the target resolution and assign the label of the closest vertex of the mesh. Closest points are efficiently found using KD-tree-based nearest neighbor search.

**Sequential Training.** The temporal expert network needs to learn how to fuse new information into the existing scene representation based on sequential data. A key challenge is catastrophic forgetting, where the network forgets what it has learned during the beginning of a sequence and only focuses on the last few frames along a camera trajectory. To overcome this challenge, it is critically important to randomly select camera views along each video trajectory, *i.e.*, a permutation of the original frame order. Similarly, to avoid that the model only sees fully reconstructed scenes after some initial training time, we randomly reset the reconstructed scenes so that the model always sees scenes at varying levels of reconstruction.

## 4. Experiments

### 4.1. Implementation Details

We implement the proposed pipeline in PyTorch. We use the MinkowskiEngine [3] for the sparse 3D convolutions in the 3D encoder, and Pytorch3D [28] for the geometric projections. The entire pipeline is trained with the Adam optimizer and a OneCycle [32] learning rate scheduler. Due to memory constraints, the batch size is 4 but we obtain an effective batch size of 8 by aggregating the gradients across two batches. We set the maximum learning rate to 0.001 for the 3D and temporal expert networks, while the maximum learning rate for the pre-trained 2D encoder is set to  $1e-05$ . We equally weight the different terms in the loss function setting  $\lambda_{3D} = \lambda_{2D} = \lambda_{\text{Expert}} = 1$ . Further, we set the parameter of the focal loss  $\gamma = 1$ . We use five layers in the temporal expert transformer with a hidden dimension of  $D_{\text{hidden}} = 128$  in the feed-forward layers. The voxel grid resolution for the entire pipeline is set to 4 cm.

### 4.2. Online Methods in Comparison

**FusionAware [37].** Unlike our voxel-based representation, FusionAware is a *point*-based online 3D semantic segmentation method. The method aggregates measurements in 3D space using point convolutions and computes intra- and inter-frame features.

**SVCNN [15].** Similar to ours, Supervoxel Convolution (SVCNN) is another candidate from the space of voxel-based approaches. SVCNN uses dedicated convolutional

	Method	Processing	mIoU $\uparrow$	Bathtub	Bed	Bookshelf	Cabinet	Chair	Counter	Curtain	Desk	Door	Floor	Other Furniture	Picture	Fridge	Shower Curtain	Sink	Sofa	Table	Toilet	Wall	Window
Offline	Mix3D [25]	Global	78.1	96.4	85.5	84.3	78.1	85.8	57.5	83.1	68.5	71.4	97.9	59.4	31.0	80.1	89.2	84.1	81.9	72.3	94.0	88.7	72.5
	VirtualMVFusion [17]	Global	74.6	77.1	81.0	84.8	70.2	86.5	39.7	89.9	69.9	66.4	94.8	58.8	33.0	74.6	85.1	76.4	79.6	70.4	93.5	86.6	72.8
	Minkowski [3]	Global	73.6	85.9	81.8	83.2	70.9	84.0	52.1	85.3	66.0	64.3	95.1	54.4	28.6	73.1	89.3	67.5	77.2	68.3	87.4	85.2	72.7
Online	PanopticFusion [24]	Global	52.9	49.1	68.8	60.4	38.6	63.2	22.5	70.5	43.4	29.3	81.5	34.8	24.1	49.9	66.9	50.7	64.9	44.2	79.6	60.2	56.1
	INS-Conv [19]	Global	71.7	75.1	75.9	81.2	70.4	86.8	53.7	84.2	60.9	60.8	95.3	53.4	29.3	61.6	86.4	71.9	79.3	64.0	93.3	84.5	66.3
	FusionAware [37]	Local	63.0	60.4	74.1	76.6	59.0	74.7	50.1	73.4	50.3	52.7	91.9	45.4	32.3	55.0	42.0	67.8	68.8	54.4	89.6	79.5	62.7
	SVCNN [15]	Local	63.5	65.6	71.1	71.9	61.3	75.7	44.4	76.5	53.4	56.6	92.8	47.8	27.2	63.6	53.1	66.4	64.5	50.8	86.4	79.2	61.1
	ALSTER (Ours)	Local	66.8	82.2	77.1	49.6	65.1	83.3	54.1	76.1	55.5	61.1	96.6	48.9	37.0	38.8	58.0	77.6	75.1	57.0	95.6	81.7	64.6

Table 1. **3D Semantic Segmentation on ScanNet [5] Test.** Offline baselines predict semantic labels using a-priori 3D scene reconstructions and **global** passes over the entire scene. Online but **global** baselines do online reasoning but require global passes over the full scene. **Local** methods are online and reason only on local information within the viewing frustum and on currently updated points. Among local methods, our proposed approach improves over existing baselines by at least +3.3 mIoU.

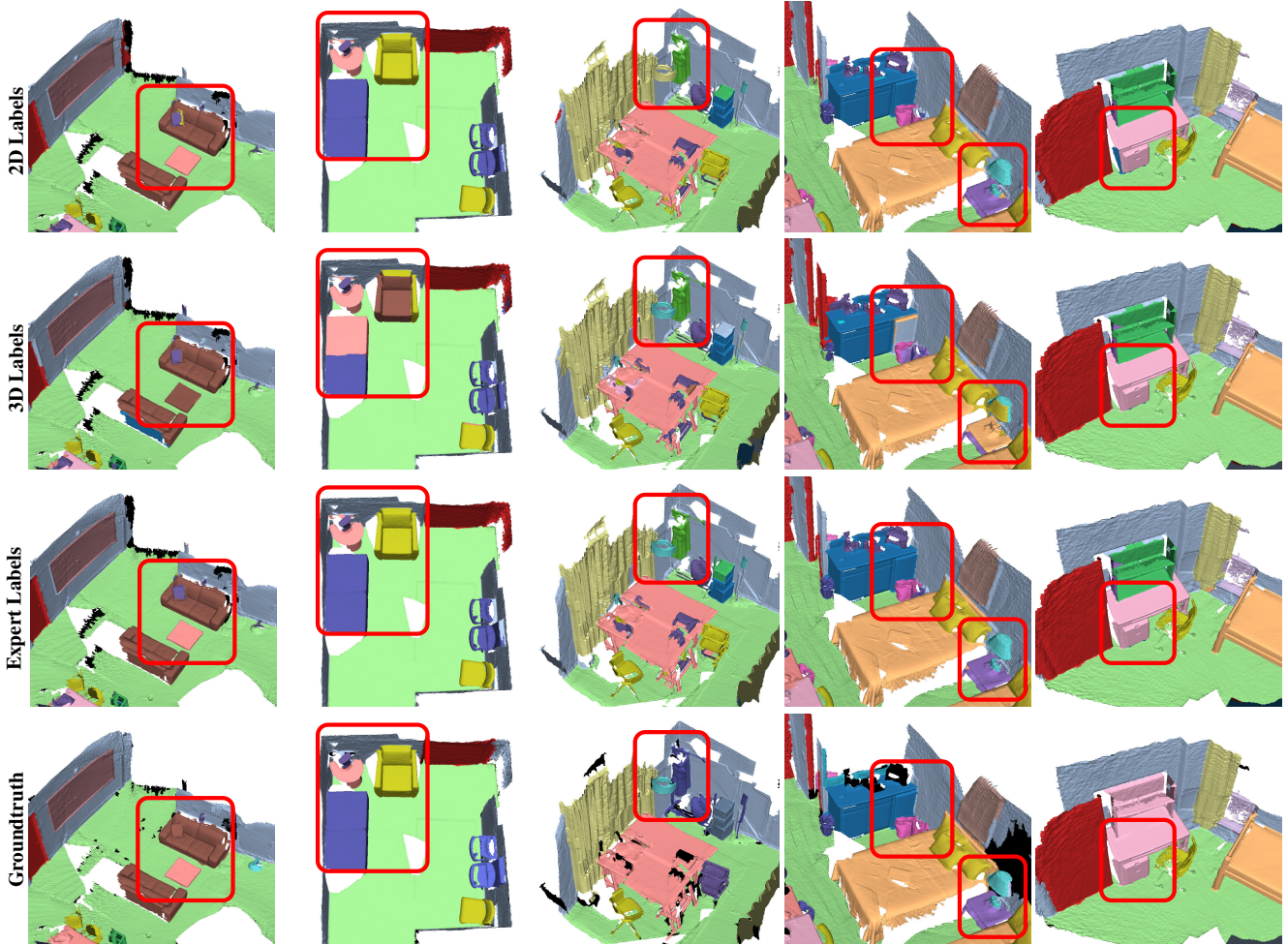


Figure 5. **Qualitative results of our proposed method and comparison between the different stages.** The expert successfully selects the correct information from the two encoders. The 2D encoder is focused on object-level decisions (e.g., table in column 1, bench in column 2) while the 3D information is used for fine details (e.g. lamp in column 3, desk in column 5).

operators that operate directly on supervoxels and aggregate multi-view features during online reconstruction.

### 4.3. Datasets and Metrics

**ScanNet [5]** consists of 1513 scans from 707 unique indoor scenes containing 2.5M RGB-D frames. All scenes provide dense 3D semantic annotations mapped to the NYU40 class labels. Each scene is recorded up to three times using an iPad equipped with an Occipital depth sensor. The camera poses and dense reconstruction of the scenes are obtained using BundleFusion [6]. The 3D labels are projected into all 2D frames to provide the 2D labels.

**SceneNN [14].** SceneNN consists of 76 scenes with semantic annotations and corresponding posed RGB-D data. We follow [15, 19] and demonstrate the generalization capabilities of our method by training on ScanNet and evaluating on the 76 SceneNN scenes.

**Metrics.** We follow the standard metrics of the ScanNet and SceneNN datasets. In particular, we compute the mean and per-class intersection over union (IoU) on ScanNet, as well as the mean accuracy (mAcc) and weighted intersection over union (wIoU) on SceneNN.

### 4.4. 3D Semantic Segmentation

Table 1 reports 3D semantic segmentation scores of our and recent methods on ScanNet [5]. We compare online and offline methods, as well as local and global methods. While offline methods rely on a pre-computed 3D scene reconstruction in the form of a point cloud or polygon mesh, online methods are able to reconstruct the 3D scene on the fly as new frames become available. This functionality is attractive for online applications in robotics or AR/VR devices, however they cannot rely on the full scene context which makes semantic reasoning harder and semantic scores are generally higher for offline methods [3, 17, 25]. In the group of online methods, our approach improves over the existing local methods like SVCNN [15] and Fusion-Aware [37] by at least +3.3 mIoU. Local methods operate on a local window defined by the newly incoming frames and are therefore memory and computationally efficient, both attractive qualities for real-time processing. Global methods require global passes over the reconstructed scenes either by CRF regularization or neural network processing. This step takes increasingly more time as the reconstructed scene becomes larger in size. These methods are therefore less applicable for real-time applications, since no upper bound on the processing time can be guaranteed.

In Table 3, we compare our method to existing baselines on the ScanNet validation set as well as SceneNN. For ScanNet, we report the mIoU over all benchmark classes. For SceneNN, we compute the weighted intersection over union (wIoU) and the mean accuracy (mAcc) for all annotated NYU40 classes. In addition to the quantitative met-

rics, we also report the voxel resolution for all methods where it is available. A smaller voxel size generally results in better scores since finer details can be represented, however this comes at increased memory costs. Our proposed method performs best among all local methods on both ScanNet and SceneNN (with a close second on the wIoU metric). When also compared to global methods, INS-Conv [19] performs only marginally better on ScanNet, even when using a voxel resolution that is twice as high, highlighting the memory efficiency of our method.

### 4.5. Ablation Studies

**Does the temporal expert network improve upon the individual 2D and 3D networks?** In Table 2, we report the numbers for the different stages in our pipeline. For each stage, we compute the per-class intersection over union on the ScanNet validation set. The per-stage labels are obtained from the auxiliary heads used during training to constrain the joint feature space and aggregated using a simple voting mechanism. These labels are then mapped to the ScanNet ground-truth and evaluated using their evaluation pipeline. The numbers show that the expert is consistently better than the individual branches (Ours - 2D and Ours - 3D). Further, the fact that sometimes the 2D labeling is better than the 3D labels and the significant margin between 3D and expert indicate that 1) bypassing the 2D information around the 3D encoder is useful and 2) our attention-based fusion mechanism allows better reasoning over time than simple voting. We also show the differences between the different stages in Figure 5, where one can see the benefits of selecting information from the two different encoders.

**What does the temporal expert network attend to?** In Figure 4, we visualize the attention maps for different frames during the fusion process together with the corresponding predicted labels. We qualitatively show that they learn to leverage the two different encoder according to their individual strengths. The temporal expert network attends to the 3D network for fine-details usually refining edges and geometric details (*e.g.*, legs of a chair, edge of a table) while it attends to the 2D feature for information about large regions. Further, we observe that the fusion with the old scene representation happens in later layers while earlier layers combine the 2D and 3D information.

**What is the impact of the feature dimension  $D_F$ ?** The dimension  $D_F$  of the features stored in the learned scene representation is a key hyperparameter of the pipeline. Thus, we evaluate its impact on the overall performance in Table 2. One can see that for  $D_F$  below 40 (the default value), the performance is slightly deteriorated due to the required compression of the semantic information. For  $D_F = 64$  the main reason for the slight performance drop is the increased overfitting to the training data due to the increased capacity of the features.



Method	mIoU $\uparrow$	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bookshelf	Picture	Counter	Desk	Curtain	Fridge	Shower Curt.	Toilet	Sink	Bathtub	Other Furn.
Ours - 2D	68.2	83.4	96.3	<b>60.1</b>	<u>75.0</u>	<u>85.8</u>	<b>77.0</b>	68.7	65.7	60.3	<b>66.7</b>	28.2	64.8	<u>56.1</u>	68.3	61.8	58.8	86.6	62.7	86.0	52.1
Ours - 3D	<u>69.0</u>	<u>85.0</u>	96.8	59.2	73.9	<b>87.3</b>	<u>75.8</u>	69.7	<b>69.5</b>	<u>61.3</u>	62.9	<u>33.1</u>	<u>65.7</u>	<u>56.1</u>	68.4	61.2	<u>60.2</u>	<u>89.0</u>	<u>64.1</u>	87.0	<u>52.8</u>
Ours - Temp	<b>70.6</b>	<b>85.6</b>	<b>96.9</b>	<u>59.6</u>	<b>77.1</b>	<b>87.3</b>	75.0	<b>71.9</b>	<u>68.4</u>	<b>65.8</b>	<u>65.9</u>	<b>36.6</b>	<b>67.2</b>	<b>61.1</b>	<b>69.4</b>	<b>62.6</b>	<b>65.0</b>	<b>89.8</b>	<b>65.2</b>	<b>87.1</b>	<b>53.8</b>
$D_F = 64$	67.9	84.4	<u>96.2</u>	59.4	76.9	85.6	73.0	69.5	<u>66.3</u>	63.1	36.4	24.8	<u>65.6</u>	<b>61.6</b>	<b>71.3</b>	<u>61.5</u>	<b>67.1</b>	89.1	64.7	<b>89.4</b>	52.9
$D_F = 40$	<b>70.6</b>	<b>85.6</b>	<b>96.9</b>	<u>59.6</u>	77.1	<b>87.3</b>	75.0	<b>71.9</b>	<b>68.4</b>	<b>65.8</b>	<b>65.9</b>	<b>36.6</b>	<b>67.2</b>	<u>61.1</u>	69.4	<b>62.6</b>	<u>65.0</u>	89.8	<u>65.2</u>	87.1	<u>53.8</u>
$D_F = 32$	<u>68.2</u>	83.0	94.9	<b>60.8</b>	<b>78.5</b>	<u>86.3</u>	<b>76.0</b>	69.8	61.6	58.7	<u>63.6</u>	24.6	64.5	<u>61.1</u>	<u>70.0</u>	60.9	50.4	<u>90.6</u>	<b>67.5</b>	<u>87.5</u>	53.0
$D_F = 16$	68.1	83.2	95.2	58.5	<u>77.9</u>	84.9	<u>75.8</u>	69.4	61.5	<u>63.8</u>	52.9	<u>33.0</u>	60.1	58.2	64.9	<u>61.5</u>	64.7	<b>90.8</b>	63.9	<u>85.9</u>	<b>55.0</b>
Xception	<b>70.6</b>	<b>85.6</b>	<b>96.9</b>	<b>59.6</b>	<b>77.1</b>	<b>87.3</b>	<b>75.0</b>	<b>71.9</b>	<b>68.4</b>	<b>65.8</b>	<b>65.9</b>	36.6	<b>67.2</b>	61.1	<b>69.4</b>	<b>62.6</b>	<b>65.0</b>	<b>89.8</b>	<b>65.2</b>	<b>87.1</b>	<b>53.8</b>
MobileNet	66.0	81.2	95.9	57.8	73.9	83.1	70.8	68.0	60.3	50.2	63.9	<b>37.3</b>	63.1	<b>61.9</b>	63.4	56.3	50.1	88.6	58.5	86.2	50.0

Table 2. **Ablating different aspects of our pipeline on ScanNet [5] validation set.** We show that the expert selects valuable information from the two encoders and the existing scene representation by evaluating the individual pipeline outputs. The expert network consistently improves upon the two other stages in terms of IoU. We also evaluate the impact of the stored feature dimension  $D_F$  on the overall result. While the smaller feature sizes suffer from compression due to limited capacity the larger features ( $D_F = 64$ ) suffer from slight overfitting. Finally, we compare the Xception 2D encoder to the lighter MobileNet ( $\times 20$  fewer parameters). Unsurprisingly, the smaller encoder leads to a slight deterioration of performance, but the results indicate potential for runtime-accuracy tradeoffs in time-critical applications.

3D Semantic Segmentation			ScanNet		SceneNN	
	Processing	Res. [cm]	val. mIoU	wIoU	mAcc	
SemanticFusion [20]	Global	N/A	42.3	<b>47.1</b>	<u>58.5</u>	
PanopticFusion [24]	Global	2.4	<u>53.1</u>	—	—	
InsConv [19]	Global	2	<b>72.4</b>	—	<b>79.5</b>	
SemanticReconstruction [16]	Local	N/A	44.0	—	—	
ProgressiveFusion [27]	Local	0.8	55.0	52.2	61.6	
FusionAware [37]	Local	N/A	67.2	63.9	71.7	
SVCNN [15]	Local	N/A	<u>68.3</u>	<b>69.0</b>	<b>76.9</b>	
ALSTER (Ours)	Local	4	<b>70.6</b>	<u>67.8</u>	<b>76.9</b>	

Table 3. **3D Semantic Segmentation on ScanNet and SceneNN.** Scores are mean intersection over union (mIoU) on ScanNet [5] validation, the mean accuracy (mAcc) and weighted IoU (wIoU) on SceneNN [27]. All other scores are as reported in [15] and [19].

**How fast is our method?** An average step through our entire (non-optimized) pipeline takes 116.1 ms (8.6 FPS) on an NVIDIA RTX 2080 and 3.6 GHz Intel CPU i9-9900K. To identify the main bottleneck, we analyse the runtime of the individual components in Fig. 6. One pass through the 2D network DeepLabV3 [1] plus lifting takes on average 83.5 ms. One pass through our light-weight 3D UNet takes 28.6 ms on average, and the temporal expert network operates at 4 ms per frame. These numbers reveal the 2D DeepLabV3 as the main bottleneck. Thus, we also report the performance of MobileNet [11] in Tab. 2 instead of the standard Xception encoder. This reduces the 2D processing time to 45.2 ms and boosts the overall runtime to 12.9 FPS.

**How many parameters does our pipeline have?** Our pipeline consists in  $51 \cdot 10^6$  parameters in total. The largest share is due to the DeepLabV3 (Xception) model, which consists of  $41 \cdot 10^6$  parameters. The 3D U-Net consists of

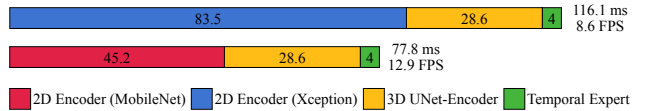


Figure 6. **Runtime analysis for different 2D backbones.** A smaller 2D encoder trades runtime for accuracy (cf. Tab. 2).

$10 \cdot 10^6$  parameters. Compared to both encoders, the expert model takes a relatively small share of  $92 \cdot 10^3$  parameters. This further justifies our architecture design, with a marginal increase in model size and runtime, we obtain a notable boost in performance (+1.6 mIoU, see Tab. 2)

## 5. Conclusion

We presented a novel pipeline for online joint geometric and semantic 3D reconstruction. The pipeline consists of three components and a learned scene representation that represents the scene as sparse voxel grid. In order to leverage the complementary nature of 2D and 3D information, the first two stages encode 2D RGB-D data and enhance the encoded features with 3D spatial information. At the heart of your pipeline sits a temporal expert fusion network, that sequentially updates the learned scene representation. This network attends to the 2D, 3D, and existing features to extract relevant information for the updates. We experimentally show that this design improves the performance on semantic segmentation upon the two individual branches.

**Acknowledgments.** This research is partially supported by Toshiba, the ETH AI Center PostDoc Fellowship, and an ETH Career Seed Award funded through the ETH Zurich Foundation.



## References

- [1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 4, 8
- [2] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4
- [3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 5, 6, 7
- [4] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996. 3
- [5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4, 6, 7, 8
- [6] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3):24:1–24:18, 2017. 7
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 4
- [8] Saurabh Gupta, Ross B. Girshick, Pablo Andrés Arbeláez, and Jitendra Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, pages 345–360. Springer, 2014. 4
- [9] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3
- [10] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusetnet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I*, pages 213–228. Springer, 2016. 4
- [11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 8
- [12] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional Projection Network for Cross Dimension Scene Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [13] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew Lan Tai. Vmnet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. In *International Conference on Computer Vision (ICCV)*, 2021. 3
- [14] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A Scene Meshes Dataset with Annotations. In *International Conference on 3D Vision (3DV)*, 2016. 7
- [15] Shi-Sheng Huang, Ze-Yu Ma, Tai-Jiang Mu, Hongbo Fu, and Shi-Min Hu. Supervoxel Convolution for Online 3D Semantic Segmentation. *ACM Transactions on Graphics (TOG)*, 2021. 2, 3, 5, 6, 7, 8
- [16] Junho Jeon, Jinwoong Jung, Jungeon Kim, and Seungyong Lee. Semantic Reconstruction: Reconstruction of Semantically Segmented 3D Meshes via Volumetric Semantic Fusion. In *Computer Graphics Forum*, 2018. 3, 8
- [17] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual Multi-view Fusion for 3D Semantic Segmentation. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 4, 6, 7
- [18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017. 5
- [19] Leyao Liu, Tian Zheng, Yun-Jou Lin, Kai Ni, and Lu Fang. INS-Conv: Incremental Sparse Convolution for Online 3D Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 6, 7, 8
- [20] John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *International Conference on Robotics and Automation (ICRA)*, 2017. 2, 3, 8
- [21] John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. Fusion++: Volumetric Object-Level SLAM. In *International Conference on 3D Vision (3DV)*, 2018. 3
- [22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [23] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [24] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019. 2, 3, 6, 8
- [25] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *International Conference on 3D Vision (3DV)*, 2021. 1, 2, 6, 7

- [26] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 3
- [27] Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Real-Time Progressive 3D Semantic Segmentation for Indoor Scenes. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2019. 3, 8
- [28] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5
- [29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 4
- [30] Martin Rünz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018. 3
- [31] Lukas Schneider, Marius Cordts, Timo Rehfeld, David Pfeiffer, Markus Enzweiler, Uwe Franke, Marc Pollefeys, and Stefan Roth. Semantic Stixels: Depth is not enough. In *IEEE Intelligent Vehicles Symposium (IV)*, 2016. 2
- [32] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR*, abs/1708.07120, 2017. 5
- [33] Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A Prisacariu, Olaf Kähler, David W Murray, Shahram Izadi, Patrick Pérez, et al. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 75–82. IEEE, 2015. 2, 3
- [34] Jinghua Wang, Zhenhua Wang, Dacheng Tao, Simon See, and Gang Wang. Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 664–679. Springer, 2016. 4
- [35] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. RoutedFusion: Learning Real-Time Depth Map Fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3
- [36] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. NeuralFusion: Online Depth Fusion in Latent Space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [37] Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Fusion-Aware Point Convolution for Online Semantic 3D Scene Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 5, 6, 7, 8
- [38] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J. Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [39] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place Scene Labelling and Understanding with Implicit Scene Representation. In *International Conference on Computer Vision (ICCV)*, 2021. 2