Temporally Consistent Semantic Segmentation using Spatially Aware Multi-view Semantic Fusion for Indoor RGB-D videos

Fengyuan Sun Sezer Karaoglu Theo Gevers
University of Amsterdam & 3DUniversum
Amsterdam, The Netherlands
fengyuansun2000@gmail.com s.karaoglu@3duniversum.com Th.Gevers@uva.nl

Abstract

The task of performing image semantic segmentation faces challenges in achieving consistent and robust results across a sequence of video frames. This problem becomes more prominent for indoor scenes where small camera movement can lead to drastic appearance changes, occlusions, and loss of global context information.

To overcome these challenges, this paper proposes a novel approach that combines multi-view semantic fusion with spatial reasoning to produce view-invariant semantic features for temporally consistent semantic segmentation for indoor RGB-D videos.

The experiments are conducted on the ScanNet dataset, showing that the proposed spatially aware multi-view fusion mechanism significantly improves the state-of-the-art image semantic segmentation methods Mask2Former and ViT-Adapter. In particular, the proposed pipeline offers improvements of 5%, 9.9%, and 14.4% in 2D mIoU, cross-view consistency, and temporal consistency, respectively, when compared to Mask2Former. Similarly, when compared to ViT-Adapter, the proposed mechanism offers enhancements of 4.8%, 8.9%, and 10.9% in the same metrics.

1. Introduction

While there has been considerable progress in image semantic segmentation [36, 24, 4], there has been relatively little research focused on achieving consistent results across a sequence of video frames. However, the need for temporally coherent semantic segmentation has become increasingly critical in many fields, including robotics, virtual reality, and augmented reality.

The task of performing image semantic segmentation faces challenges in achieving consistent results across a sequence of video frames, where viewpoint changes can cause inconsistent predictions between views. In a single image, the appearance is encoded through the relationship between the scene and the camera viewpoint. However, in a video sequence, the viewpoint changes over time, resulting in a continuous variation in the appearance. This means that small changes in camera viewpoint can cause significant appearance changes, particularly in indoor scenes where the scene is in close proximity to the camera. These appearance changes can be attributed to the variation in the scene's perspective and the occlusion of objects in the scene.

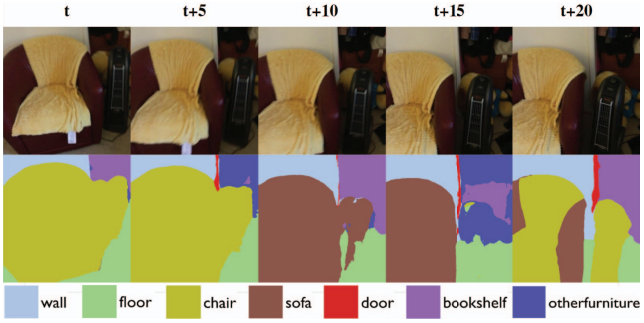

Figure 1: The qualitative results of state-of-the-art image-based semantic segmentation algorithm, ViT-Adapter, on a selected sequence of data from ScanNet. Despite the relatively stable viewpoint and appearance, the algorithm encounters difficulties in predicting consistent and reliable labels over time.

In addition, indoor scenes may present more challenging appearances since the scene is often very close, and certain views can lose global context information. For example, when recording a corridor, the camera is often positioned very close to the wall, resulting in a loss of global context information.

Most existing methods on video semantic segmentation exploit the temporal relationship between frames by propagating features with optical flow [16, 22] or use attention mechanisms to establish temporal relations [34, 14]. These methods rely on the assumption of feature consistency across frames to establish temporal relationships be-

tween them. However, this assumption represents a significant challenge in the case of indoor scenes, where rapid changes in viewpoint and high rates of occlusion can result in a significant appearance change. Consequently, these methods have limitations in their capacity to effectively exploit the complete temporal relationship between frames for indoor scenes.

Motivated by the more readily accessible RGB-D data, a new approach is taken to video semantic segmentation to circumvent the difficulties on indoor scenes. This task can be re-framed into a multi-view semantic fusion problem by carrying the predictions from 2D to 3D space and fusing them into a unified and consistent segmentation result. Then, by re-projecting the predictions to 2D, a temporally consistent video segmentation can be acquired. This approach reduces the need for expensive temporal consistency modeling in video segmentation networks, and can generate temporally and spatially consistent segmentation results from single-image segmentation networks. Additionally, by applying the multi-view fusion method directly on 2D pixel-wise semantic labels, the proposed method can process a whole video sequence at once, independent of the 2D segmentation network.

This approach has been demonstrated to improve segmentation accuracy, particularly in cases where individual views may lose global context information due to close proximity to the scene, significant appearance change between frames or occlusion of objects. However, existing methods that employ multi-view semantic fusion strategies have not yet fully exploited the geometric and the view-dependent features of the data, which can help to resolve ambiguous predictions in 2D.

To address these challenges, our proposed approach combines multi-view semantic fusion with spatial reasoning to produce view-invariant semantic features for indoor video semantic segmentation. By leveraging the rich information provided by multiple views, our method produces more robust and accurate semantic features, which are refined using geometric data through spatial reasoning. The refined segmentation map is projected from 3D to 2D, resulting in improved video semantic segmentation accuracy for indoor scenes.

The proposed method has following contributions:

- A novel multi-view semantic fusion with spatial reasoning pipeline is proposed. The proposed pipeline is tested using two off-the-shelf state-of-the-art image semantic segmentation algorithms. The results show that spatially-aware fusion helps predicting temporally consistent and more robust semantic labels for indoor RGB-D videos.

- The proposed method reaches state-of-the-art 2D mIoU performance on semantic segmentation on ScanNet validation set. In addition, it reaches better 3D mIoU performance than the previous best-performing multi-view methods.

- To the best of our knowledge, we are the first to propose temporal coherency scores on ScanNet dataset. The proposed method establishes a baseline for further research and achieves the highest cross-view consistency and temporal consistency scores on ScanNet dataset.

- The proposed method is flexible and can be used with any image semantic segmentation pipeline.

2. Related work

Multi-view semantic fusion. As a different approach to directly processing 3D data, other research has focused on segmenting images in 2D and projecting the segmentation scores onto 3D space. Early works commonly extract pixel-wise semantic features and aggregate them using weighted averaging [33] and Bayesian fusion [26, 13] followed by a Conditional Random Field model to regularize the 3D segmentation, or employed a label diffusion method [25] to unify both steps. These fusion methods, however, deal with difficulties due to occlusion, illumination, and camera pose inaccuracies present in RGB-D data. Some approaches [20, 11, 5, 21] have explored synthesizing virtual views from real data to alleviate these problems, sampling views from better viewpoints and even rendering additional data channels. While this approach can improve segmentation in 2D, they still use simple multi-view fusion strategies and do not leverage the geometric features, resulting in difficulties resolving ambiguous predictions in 2D.

Closer to our setting, 3DSceneGraph [1] and 2D3DNet [10] proposed to aggregate semantic labels following a distance-based weighting scheme. Similarly, 2D3DNet further processes the features with 3D convolutions, but do not exploit other viewing conditions for fusion. Additionally, they focus on learning 3D segmentation using only pseudo-labels from 2D, while our task assumes labels in 3D to further improve performance for both the 2D and 3D domain.

Combined 2D/3D segmentation. Some existing methods have leveraged the complementary features between 2D and 3D. They typically extract features using 2D convolutions and project them back to 3D [8, 19]. Some approaches [20, 11] generate virtual views to further improve performance in 2D. Another method [15] jointly learns 2D and 3D segmentation, allowing information to flow from both domains. While these approaches are similar to ours, they do not leverage view-dependent and geometrical features to learn multi-view fusion but instead use simple aggregation methods. These multi-modal networks are generally constrained in the number of input images, effectively limiting the performance on large scenes. More recently, Robert *et*

al. [29] propose to learn multi-view feature fusion of 2D extracted features for 3D segmentation. While they are able to improve the computational efficiency of hybrid 2D/3D frameworks, they are still limited to simple 2D backbones during training. Our approach for multi-view semantic fusion can be applied to any 2D segmentation network.

Video semantic segmentation. Existing work on video semantic segmentation focuses on increasing temporal consistency in the predictions. Most methods are based on optical flow [16, 22], either using it to propagate semantic predictions or features to following frames, or as regularization for network training [23]. Although optical flow is widely used, it is still computationally expensive and is susceptible to errors from sudden viewpoint changes and occlusions, which is especially prevalent in indoor scenes. Differently, some works focus on integrating temporal relations using attention mechanisms [14, 34], but still assume conformity of features between frames.

3. Method

Given a semantic video sequence with segmentation maps M, a 3D point cloud reconstruction of the scene P and the aligned camera poses of each view, our goal is to refine M by leveraging multi-view information inherent in correspondences between 3D points and 2D pixels. The overall structure of the pipeline is presented in Figure 2. First, the correspondences between points and masks are calculated using a visibility model. For each point-image pair, viewing descriptors are computed using camera pose and local point geometric information. Then, for each point, the viewing descriptors and segmentation labels from all corresponding views are processed simultaneously by an attention-based fusion module to aggregate relevant view-dependent, geometric and semantic features. These multi-view fused features are concatenated to the XYZ-features of the points and further processed by a 3D network to to spatially refine predictions. Finally, the refined predictions are projected to 2D, acquiring multi-view refined segmentation masks.

3.1. Preprocessing

Prior to training the multi-view refinement network, a series of steps were taken to prepare the input data. This section explains the construction of the point-image mapping and its viewing descriptors.

3.1.1 Point-image mapping

Pixels in 2D space cannot be easily back-projected to 3D space because of occlusions. Ambiguity arises when two objects share the same line of sight from the camera view. To resolve this pixel to object assignment, a visibility model is needed. Z-buffering [29] method is used to compute the point-image mapping between the point cloud and images. Compared to the traditional methods that either use true depth maps from a depth-sensor or an expensive mesh reconstruction step, this method is computationally more efficient while removing the need for true depth maps.

The valid point-image pairs $(p, i) \in P \times I$ are calculated for each point p in the point cloud and each image i in the video sequence. A pair is valid if p is seen in i without it being occluded. The projected pixel location of p within i is denoted as $pix(p, i)$.

The point-image mapping is then constructed as followed. For each image $i \in I$, the points in the frustum of i are first placed on a plane orthogonal to i at a set distance. Each point is assigned a cube of varying size that decreases based on its distance to i, ensuring that cubes closer to the image hide cubes behind them. The maximum distance was set to 8m and points further away were dropped. Then, the projection mask or *splat* of each cube is calculated using the camera parameters of i. Repeatedly, each splat is collected in a depth map, called Z-buffer, which stores the closest distance of each point-pixel projection. The indices of the closest points are stored in a separate index buffer. After this process, the Z-buffer contains only the non-occluded points and the projected pixel location for each point p within each image i is saved in $pix(p, i)$, which forms the final point-image pairs (p, i).

3.1.2 Viewing descriptors

The correctness of segmentation masks produced by a 2D network can depend on the viewpoint from which the image was taken. To describe the properties of a point-image pair, several view-dependent features and local geometric features are calculated, following Robert *et al.* [29]. A vector of 7 computed features is assigned to each (p, i) pair, forming the viewing descriptors $o_{(p,i)}$.

The view-dependent features are:

- **Projected depth.** The distance between the camera and seen object can affect the quality of cues received by the 2D network. For example, an object too distant from the camera may be perceived with less detail. The depth is computed by taking the distance between a point and its corresponding viewpoint, and is further normalized by the maximum distance of 8m.
- **Viewing angle.** When a surface is viewed at a right angle, it could be captured more completely than when it is viewed at a slanted angle. This in turn affects the segmentation. The viewing angle is computed by taking the absolute cosine of the angle between the estimated normal vector and the camera viewing direction.
- **Occlusion rate.** Objects and surfaces in the background are often partially occluded. This reduces the relevant context, thus views with less occlusion tend to

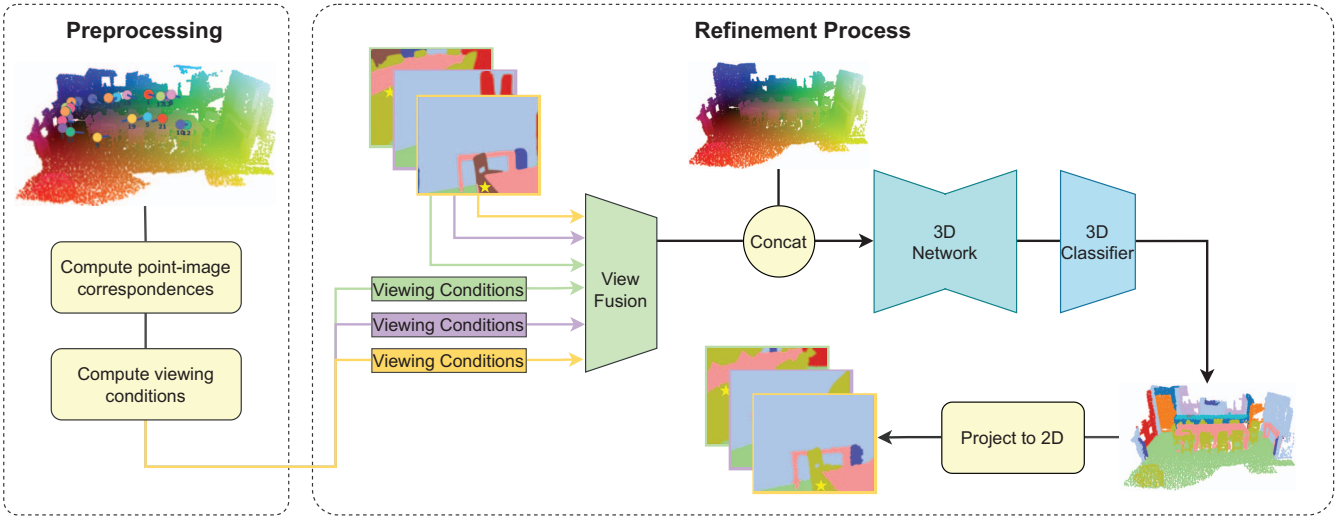

Figure 2: Our multi-view fusion and spatial refinement pipeline. First, the data is preprocessed for a scene. Then, multiple semantic views are fused into one consistent semantic feature based on their viewing conditions. Lastly, the semantic and geometric features are jointly processed by a 3D network to refine predictions, and the per-point semantics are predicted. Semantic segmentation results in 2D are acquired by projection.

produce a better segmentation. The occlusion rate of a point from a certain view is calculated as the ratio of the 50 nearest points that are non-occluded in that same view.

And the local geometric features are:

- **Local density.** The density around a point can affect the significance of other neighborhood-based descriptors. For example, occlusion rates provide more accurate representations when computed on a locally dense area, since the neighboring points are less spread out. The local density is calculated as the area of the smallest disk containing all 50 nearest neighbors, divided by the square of the voxel grid resolution.
- **Linearity, planarity and scattering.** These geometric descriptors provide information on the shape of a local area. They signal the spread of points in one, two and three dimensions, respectively. Surfaces that are planar can be best captured in 2D, while highly linear and thin surfaces can be more difficult for a 2D network to segment. Moreover, highly irregular surfaces can occlude parts of itself when captured in 2D. Hence, these features can help discern the quality of the camera views. They are calculated using the eigenvalues from the covariance matrix of a point and its 50 nearest points, following Demantké *et al.* [9].

3.2. Learning multi-view semantic fusion via Transformers

In indoor scenes, a 3D point can be seen from multiple views. Since viewpoint changes can affect the predicted segmentation mask, its corresponding 2D labels can become inconsistent with each other. To produce multi-view consistent predictions for each point, simple fusion methods, such as taking the average, can be used. However, these do not take the quality of each view into account when fusing the predictions, thus producing a wrong label when the correct label was not the majority vote. To tackle this challenge, this section proposes a multi-view feature fusion method based on Transformer self-attention that learns to aggregate relevant features from multiple views based on their viewing conditions.

3.2.1 View fusion

The viewing conditions of a point-image pair is described by a set of view-dependent and local geometric features. However, the factors that define a high quality view are context dependent. For example, a wall is best recognizable from a distance with a frontal view, while a sink is best viewed from up close with a downwards view. To capture these complex dependencies between viewing conditions and semantic object classes, the predicted class labels are concatenated to the viewing conditions to form the Transformer input.

The steps to produce a multi-view fused feature vector for one 3D point p is given as follows. First, a maximum of K views that contain p are randomly selected. The semantic label of the selected point-image pairs (p, i) is then extracted:

$$s_{(p,i)} = M_i[pix(p, i)], \tag{1}$$

where M_i is the segmentation map from the 2D model for

image i. They are then concatenated to the viewing descriptors to form feature vectors:

$$x_{(p,i)} = [o_{(p,i)}, \tilde{s}_{(p,i)}], \tag{2}$$

with $\tilde{s}_{(p,i)}$ being the one-hot encoded semantic label. These features are then projected to embedding vectors $\theta_{(p,i)} \in \mathbb{R}^D$ using a single linear layer:

$$\theta_{(p,i)} = \mathbf{Linear}(x_{(p,i)}) \tag{3}$$

To satisfy the sequential input format of Transformers, each selected view i and its embedding vector $\theta_{p,i}$ is treated as a single sequence element, and the number of vectors in θ_p is zero padded to length K. The Transformer then takes the embedding vectors and predicts a multi-view fused feature $\phi \in \mathbb{R}^D$:

$$\phi_p = \mathbf{Transformer}([\theta_{(p,1)}, ..., \theta_{(p,K)}]) \tag{4}$$

These final features are further used in the 3D network.

3.3. Spatial refinement

Although the view fusion module produces view-consistent features, these features are still processed independently for each 3D point, resulting in incoherent segmentation between object boundaries. We exploit the geometric context to refine the initial segmentation by applying a sparse convolutional network to the fusion output. For each point $p \in P$, we first concatenate its XYZ-coordinates to the fused features $\phi_p \in \mathbb{R}^D$, forming the 3D input features $\tilde{\phi}_p \in \mathbb{R}^{D+3}$. These are then concurrently processed at various scales by down-sampling and up-sampling, allowing the network to attend to neighboring features on both local and global spatial contexts.

3.3.1 Network architecture

The work of Xiong *et al.* [35] was followed for the architectural design. The view fusion module is visualized in Fig. 3. The Transformer encoder consists of four stacked layers. Each layer has a Multi-Head Attention block with two attention heads and a Feed-Forward Network, as well residual connections. The Feed-Forward Network consists of two linear layers and two dropout layers, with a GELU activation layer in between. For the network's embedding and hidden dimensions, we use $D = 64$ and $H = 256$. Layer normalization [2] is used throughout the architecture.

In order to aggregate features from multiple views into one multi-view fused feature, a learnable embedding ([CLS] token) is appended to the input sequence. In the Transformer encoder, this token interacts with other elements from the input sequence through an attention scheme. After processing it with the Transformer encoder, we extract

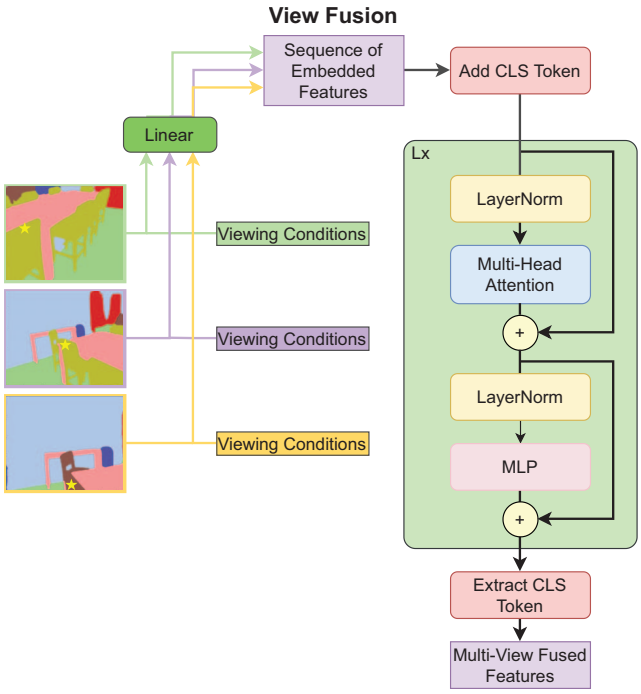

Figure 3: Transformer encoder

this [CLS] token, which finally forms the multi-view fused feature $\phi \in \mathbb{R}^D$.

Res16UNet34 is used as our 3D network, which is based on MinkowskiNet [6] and is similar in structure to the UNet. This architecture consists of 5 encoding and 5 decoding layers, with skip connections on each layer depth. The encoding layers are composed of a strided convolution with kernel size=[3, 2, 2, 2, 2] and stride=[1, 2, 2, 2, 2], and is followed by N=[0, 2, 3, 4, 6] ResNet blocks [12] with channel size=[64, 32, 64, 128, 256]. The decoding layers each begin with a strided transposed convolution with kernel size=[2, 2, 2, 2, 3] and stride=[2, 2, 2, 2, 1], followed by N=1 ResNet blocks of channel size=[128, 128, 96, 96, 96]. Each convolutional layer is followed by a BatchNorm [18] and a ReLU layer. Finally, the output passes through a 3D classification head with a single linear layer and a softmax activation to get per-class predictions.

4. Experiments

4.1. Dataset

ScanNet [7] is a large-scale RGB-D video dataset with over 2.5 million frames and their corresponding camera poses, captured with an IPad Air2 and an attached depth sensor. It contains 3D reconstructions of various scenes, such as offices, living rooms and bathrooms. The dataset is split into a training set with 1201 scans and a validation set with 312, all annotated with 20 semantic label cat-

Fusion method	2D mIoU	CC	TC
-	72.4	70.4	78.2
Random selection	69.6	68.3	80.9
Majority voting	73.7	72.2	87.4
Weighted averaging [1]	73.9	73.7	87.4
Ours	77.4	80.3	92.6

(a) Mask2Former

Fusion method	2D mIoU	CC	TC
-	74.4	72.1	81.9
Random selection	71.8	69.9	82.5
Majority voting	75.4	73.1	88.1
Weighted average [1]	75.6	74.7	88.1
Ours	79.2	81.0	92.8

(b) ViT-Adapter

Table 1: Comparison of the proposed multi-view semantic fusion approach against statistical fusion methods, as well as state-of-art single image semantic segmentation baselines, Maks2Former and ViT-Adapter. The performance is measured in 2D mIoU, cross-view consistency (CC) and temporal consistency (TC). The proposed method significantly improves the performance of our baselines in all metrics.

egories. Additionally, there is a withheld test set containing 100 scenes for the online benchmark.

We extract point clouds from the reconstructed surfaces and use the XYZ-coordinates as 3D input. For each scene, we subsample the image sequence by selecting key frames, following [31]. A key frame is selected if its relative translation is larger than $t_{max} = 0.3$ and its relative rotational angle is greater than $t_{rot} = 15$. This leaves an average of 150 views per scene. We resize the 2D images to 640×480 pixels.

4.2. Implementation Details

The maximum number of selected views per point is set to $K = 6$ for the view fusion Transformer.

Our network is trained using the cross entropy loss and stochastic gradient descent solver with momentum and weight decay set to 0.85 and 10^{-4}, respectively. The OneCycleLR scheduler from the Pytorch [28] library is employed with a minimum and maximum learning rate between 0.012 and 0.3. A batch size of 6 is used and the network is trained for 60 epochs. Data augmentation is applied to the input point clouds including random scaling, rotation around all three axes, symmetry around X and Y axes, and a Gaussian jitter is added to the input viewing conditions. Additionally out-of-context data augmentation using Mix3D [27] is applied, which combines two point clouds of different scenes. During training, the number of sampled views per scene is limited to 100.

4.3. Evaluation Metrics

The proposed method is evaluated considering two aspects: accuracy and consistency of predictions. The accuracy is measured using the mean of class-wise Intersection over Union (mIoU) on the 2D and 3D predictions. For temporal consistency (TC) [32], the segmentation from every two neighboring frames is warped and the mIoU difference is calculated, taking the first frame as source and second as target. FlowNet2 is used [17] to calculate the optical flow.

A downside of the TC metric is that it does not account for longer dependencies within the video, which occurs frequently in indoor video data. Ideally, predictions of objects that re-appear in view should be consistent with respect to associated predictions in the past. To capture this aspect, the cross-view consistency metric is formulated based on Shannon Entropy [30]:

$$CC(X) = -\sum_{x \in X} p(x) \log p(x) \cdot \frac{1}{\log |X|}, \qquad (5)$$

where X denotes the set of distinct class labels from the predictions, $p(x)$ the likelihood of x and $|X|$ the size of the label set. $p(x)$ is estimated as the proportion of labels belonging to class x. The cross-view consistency measures the coherence between predictions from multiple views. It is further normalized by the length of the set to handle label sets with varying number of classes.

4.4. Results

Comparison to statistical multi-view semantic fusion methods. In this experiment, the effect of the proposed multi-view fusion pipeline on indoor video semantic segmentation is compared against statistical multi-view semantic fusion methods. The performance is evaluated using 2D mIoU, cross-view consistency (CC) and temporal consistency (TC). These metrics are used to highlight the improvement on individual frame semantic segmentation as well as temporal coherency.

The experiments are performed using two off-the-shelf state-of-art image semantic segmentation algorithms, Mask2Former [4] and ViT-Adapter [3], and these are considered as the baselines. These networks are first trained on labeled images to create dense semantic predictions for images, which are then used by the proposed multi-view fusion pipeline and other multi-view semantic fusion approaches. Unless otherwise specified, evaluations are carried out on the ScanNet validation split, using all sampled images in a scene. The final 2D segmentation masks are

Method	Avg.	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	refrigerator	shower curtain	toilet	sink	bathtub	otherfurniture
Mask2Former	72.4	85.7	91.2	66.9	82.4	77.1	72.7	78.4	65.6	62.0	**73.9**	**47.1**	60.0	63.2	73.0	71.4	74.5	92.5	**71.8**	83.5	55.2
Majority voting	73.7	85.5	92.3	70.4	84.9	80.0	77.6	79.1	69.4	63.1	72.0	43.9	59.6	66.6	75.2	73.4	74.9	91.3	67.8	83.5	62.9
Weighted averaging [1]	73.9	86.0	92.5	70.2	85.0	80.7	77.7	79.6	69.9	63.9	72.4	43.8	59.4	66.6	75.7	73.7	75.6	91.9	67.9	83.8	62.5
Mask2Former (Ours)	**77.4**	**88.7**	**95.7**	**72.9**	**86.7**	**88.6**	**83.2**	**83.5**	**73.4**	**66.2**	71.7	46.3	**63.8**	**68.8**	**81.8**	**78.1**	**81.1**	**93.8**	69.7	**87.6**	**67.2**
ViT-Adapter	74.4	86.9	91.8	68.2	84.6	80.0	76.2	79.8	73.9	69.6	**74.2**	**36.1**	58.3	68.4	73.5	78.4	76.4	93.5	**73.3**	85.4	60.0
Majority voting	75.4	86.0	92.6	71.6	**88.0**	82.7	81.3	80.7	76.4	70.0	71.3	30.7	59.0	71.6	76.4	80.2	77.4	91.8	68.9	84.5	67.0
Weighted averaging [1]	75.6	86.6	92.7	71.8	87.7	82.8	80.1	81.0	76.1	70.5	71.7	31.5	59.8	71.5	76.8	80.0	77.6	92.4	69.2	84.1	67.1
ViT-Adapter (Ours)	**79.2**	**88.7**	**95.8**	**75.0**	87.9	**91.2**	**87.4**	**84.8**	**79.8**	**72.6**	73.5	31.8	**65.6**	**76.6**	**81.6**	**84.7**	**85.1**	**94.2**	69.7	**88.1**	**69.0**

Table 2: Class-wise 2D IoU scores.

acquired by re-projecting the aggregated 3D semantic prediction. The voxel resolution is set to 0.03 and each point cloud is downsampled accordingly using grid-sampling.

The performance of our approach is compared against several statistical multi-view label fusion methods:

- Select random: selects a view at random and assigns the associated semantic label to the 3D point.
- Majority voting: counts the semantic label from each individual view and assigns the most frequent class to the 3D point. This method is the closest to Bayesian averaging [26, 13], which does not apply to our problem since we focus on hard labels.
- Weighted averaging [1]: gives labels from each view a weight based on their distance to the 3D point, following the heuristic that objects closer to the camera are better visible. It then aggregates the weights and selects the label with the largest weight. This can be seen as a weighted majority voting scheme.

The results on Mask2Former and ViT-Adapter are shown in Table 1a and 1b, respectively. The proposed refinement pipeline shows significant improvement with respect to the baselines and outperforms all label fusion methods. It achieves an mIoU score of 77.4% on Mask2Former, and 79.2% on ViT-Adapter, which is 3.5% and 3.6% higher than the best-performing fusion method. Additionally, a large increase in both temporal and cross-view consistency is seen for the proposed method compared to majority voting and weighted averaging. Specifically, the proposed method is the only fusion method that is able to achieve a cross-view consistency score above 80%, highlighting the benefits of learning multi-view fusion directly from viewing conditions and leveraging geometric context. Fig. 5 provides an example visualization.

To gain a deeper understanding of model performance, the 2D class-wise IoU scores are compared in Table 2. Overall, the proposed method outperforms other statistical fusion methods substantially on all classes or achieves a similar performance. Moreover, it shows the largest improvement gains on classes that are geometrically distinct but share similar appearances with other classes in 2D, such as *sofa*, *chair*, *curtain*, *shower curtain*. Comparing to the statistical fusion methods, they are not able to achieve similar improvements on such classes. This demonstrates the added benefit of spatial reasoning, which helps to recover ambiguous predictions.

Figure 4: Visualization of 2D refined semantic segmentation using ViT-Adapter as input. From left to right: rgb image, weighted averaging segmentation, our segmentation, ground-truth segmentation. Weighted averaging produces ragged segmentation, while the proposed method shows clearer boundaries and smoother surfaces.

Figure 5: Visualization of an ambiguous class in 2D. From left to right: rgb image, weighted averaging segmentation, our segmentation, ground-truth segmentation. It it obvious that there are multiple views predicting chair as sofa. As a result, weighted averaging fails to make a robust prediction. In contrast, the proposed method predicts correct class labels for chair.

On the other hand, the refinement performs worse on the *picture* and *sink* classes. This can be explained by the limited geometrical context surrounding a picture, which is often attached to walls, while sinks are generally more occluded and often do not have too many visible viewpoints available.

Comparison to 2D/3D segmentation methods. In the second experiment, the proposed pipeline is compared to

state-of-the-art segmentation methods that also exploit the complementary information between 2D and 3D. To ensure a fair comparison, we subsample every 100 frames on ScanNet validation set and upscale the final 2D segmentation to 1296×968. The 3D segmentation is interpolated to 0.01 voxel size and includes all points of the scene, consistent with other methods. The proposed approach is evaluated using ViT-Adapter's image segmentation result.

Table 3 reports the mIoU results. The proposed method outperforms on 2D mIoU, overtaking the previous best method [20] by 4.6 points. Similarly, the proposed method also shows strong performance in 3D, performing significantly better than methods that jointly learn 2D feature extraction [15, 29, 19], which arguably provide more flexible contextual information for 3D segmentation compared to pure semantic features from 2D. This highlights the benefit of processing a large number of images concurrently in a multi-view fusion scheme, which is expensive for general 2D/3D methods. On the other hand, the images are still limited in their viewpoints and, thus, important parts of the scene can be left unseen.

	2D	3D
BPNet [15]	71.9	73.9
VMFusion [20]	74.9	76.4
DeepViewAgg [29]	-	71.0
MVPNet [19]	-	68.3
ViT-Adapter	73.2	-
ViT-Adapter (Ours)	79.5	76.4

Table 3: Semantic segmentation results on the official ScanNet validation split of different state-of-the-art methods that operate on point cloud and images.

Cross-model adaptation. To study the generalizability of the method, the network is applied to an unseen 2D segmentation network without re-training and its performance is measured. Table 4 presents the results in 2D and 3D mIoU. Training with Mask2Former and evaluating on ViT-Adapter slightly decreases the performance in both domains when compared to the default refinement setup for ViT-Adapter. On the other hand, by first training on ViT-Adapter and then refining Mask2Former segmentation, a small improvement is noticed in 2D and 3D mIoU. This suggests that training the refinement network on a stronger 2D segmentation model can help generalization on smaller models.

Ablation study The final experiment aims to study the effect of individual components in the network. Only keeping the multi-view fusion module results in a 2D mIoU of 76.3, while the standalone 3D network achieves a score of 78.9. In comparison, the complete network scores 79.1 mIoU. This shows that spatial refinement is more effective in the full network, but the multi-view fusion module is necessary to further increase performance.

Training	Refinement	2D	3D
Mask2Former	Mask2Former	77.4	77.7
	ViT-Adapter	78.8	79.5
ViT-Adapter	ViT-Adapter	79.2	79.8
	Mask2Former	77.6	77.8

Table 4: Cross-model adaptation of the 2D semantic segmentation network. We report performance in mIoU.

Moreover, the influence of each viewing condition on the multi-view fusion module is measured by removing one viewing condition at a time. Table 5 reports the results in 2D mIoU. The most impactful features are the viewing angle, the projected depth and the occlusion. Intuitively, these features are what determine the visibility of an object the most. The drop-out of every feature results in a drop in performance, which signals the usefulness of each proposed feature.

Feature	2D mIoU
projected depth	75.7
linearity	76.1
planarity	76.0
scattering	76.0
viewing angle	75.4
density	76.1
occlusion	75.8
baseline	76.3

Table 5: Influence of the viewing conditions. We measure mIoU performance after replacing each feature with its statistical mean.

5. Conclusion

In conclusion, we have presented a novel approach for temporally consistent semantic segmentation for indoor RGB-D videos that combines multi-view semantic fusion with spatial reasoning to produce view-invariant semantic features. By exploiting the rich information provided by multiple views, our proposed approach can produce more accurate and robust semantic features, even in cases where individual views may lose global context information or where there is significant appearance change or occlusion between frames. The proposed approach has been validated on ScanNet validation set, where it achieved state-of-the-art results in terms of cross-view consistency, temporal consistency, and 2D mIoU performance.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera, 2019. 2, 6, 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 5

[3] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions, 2022. 6

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation, 2021. 1, 6

[5] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H. Hsu. A unified point-based framework for 3d segmentation, 2019. 2

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks, 2019. 5

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017. 5

[8] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. *CoRR*, abs/1803.10409, 2018. 2

[9] J. Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based scale selection in 3d lidar point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3812:97–102, 2012. 4

[10] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3d semantic segmentation with only 2d image supervision, 2021. 2

[11] Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurélien Plyer, and David Filliat. Snapnet-r: Consistent 3d multi-view semantic labeling for robotics. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 669–678, 2017. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[13] Alexander Hermans, Georgios Floros, and Bastian Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631–2638, 2014. 2, 7

[14] Ping Hu, Fabian Caba Heilbron, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation, 2020. 1, 3

[15] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. *CoRR*, abs/2103.14326, 2021. 2, 8

[16] Junhwa Hur and Stefan Roth. Joint optical flow and temporally consistent semantic segmentation, 2016. 1, 3

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. 6

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 5

[19] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding, 2019. 2, 8

[20] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation, 2020. 2, 8

[21] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Deep projective 3d semantic segmentation, 2017. 2

[22] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation, 2018. 1, 3

[23] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference, 2020. 3

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 1

[25] Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Diffuser: Multi-view 2d-to-3d label diffusion for semantic scene segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13589 – 13595, Piscataway, NJ, 2021. IEEE. 2021 IEEE International Conference on Robotics and Automation (ICRA 2021); Conference Location: Xi'an, China; Conference Date: May 30 – June 5, 2021. 2

[26] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks, 2016. 2, 7

[27] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *CoRR*, abs/2110.02210, 2021. 6

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[29] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5575–5584, 2022. 3, 8

[30] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. 6

[31] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video, 2021. 6

[32] Serin Varghese, Yasin Bayzidi, Andreas Bär, Nikhil Kapoor, Sounak Lahiri, Jan David Schneider, Nico Schmidt, Peter Schlicht, Fabian Hüger, and Tim Fingscheidt. Unsupervised temporal consistency metric for video segmentation in highly-automated driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1369–1378, 2020. 6

[33] Vibhav Vineet, Ondrej Miksik, Morten Lidegaard, Matthias Nießner, Stuart Golodetz, Victor A. Prisacariu, Olaf Kähler, David W. Murray, Shahram Izadi, Patrick Pérez, and Philip H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 75–82, 2015. 2

[34] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation, 2021. 1, 3

[35] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. *CoRR*, abs/2002.04745, 2020. 5

[36] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016. 1