Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation

Damien Robert[1, 2] damien.robert@ign.fr
Bruno Vallet[2] bruno.vallet@ign.fr
Loic Landrieu[2] loic.landrieu@ign.fr

[1]CSAI, ENGIE Lab CRIGEN, Stains, France
[2]Univ Gustave Eiffel, ENSG, IGN, LASTIG, F-77454 Marne-la-Vallee, France

Abstract

Recent works on 3D semantic segmentation propose to exploit the synergy between images and point clouds by processing each modality with a dedicated network and projecting learned 2D features onto 3D points. Merging large-scale point clouds and images raises several challenges, such as constructing a mapping between points and pixels, and aggregating features between multiple views. Current methods require mesh reconstruction or specialized sensors to recover occlusions, and use heuristics to select and aggregate available images. In contrast, we propose an end-to-end trainable multi-view aggregation model leveraging the viewing conditions of 3D points to merge features from images taken at arbitrary positions. Our method can combine standard 2D and 3D networks and outperforms both 3D models operating on colorized point clouds and hybrid 2D/3D networks without requiring colorization, meshing, or true depth maps. We set a new state-of-the-art for large-scale indoor/outdoor semantic segmentation on S3DIS (74.7 mIoU 6-Fold) and on KITTI-360 (58.3 mIoU). Our full pipeline is accessible at https://github.com/drprojects/DeepViewAgg, and only requires raw 3D scans and a set of images and poses.

1. Introduction

The fast-paced development of dedicated neural architectures for 3D data has led to significant improvements in the automated analysis of large 3D scenes [24]. All top-performing methods operate on colorized point clouds, which requires either specialized sensors [62], or running a colorization step which is often closed-source [1–3] and sensor-dependent [33]. However, while colorized point clouds carry some radiometric information, images combined with dedicated 2D architectures are better suited for learning textural and contextual cues. A promising line of work sets out to leverage the complementarity between 3D point clouds and images by projecting onto 3D points the 2D features learned from real [19, 29, 32] or virtual images [15, 36]

Figure 1. **Combining 2D and 3D Information.** We propose to merge the complementary information between point clouds and a set of co-registered images. Using a simple visibility model, we can project 2D features onto the 3D points and use viewing conditions to select features from the most relevant images. We represent images at their position with the symbol and color the 3D points according to the image they are seen in.

Combining point clouds and images with arbitrary poses (*i.e. in the wild*) as represented in Figure 1, involves recovering occlusions and computing a point-pixel mapping, which is typically done using accurate depth maps from specialized sensors [12, 56] or a potentially costly meshing step [8]. Furthermore, when a point is seen in different images simultaneously, the 2D features must be merged in a meaningful way.

In the mesh texturation literature, multi-view aggregation is typically addressed by selecting images for each triangle based on their viewing conditions, *e.g*. distance, viewing angle, or occlusion [4, 39, 59]. Hybrid 2D/3D methods for large-scale point cloud analysis usually rely on heuristics to select a fixed number of images per point and pool their features uniformly without considering viewing conditions. Multi-view aggregation has also been extensively studied for shape recognition [21, 54, 61], albeit in a controlled and synthetic setting not entirely applicable to the analysis of large scenes.

In this paper, we propose to learn to merge features from multiple images with a dedicated attention-based scheme. For each 3D point, the information from relevant images is aggregated based on the point's viewing condition. Thanks to our GPU-based implementation, we can efficiently compute a point-pixel mapping without mesh or true depth maps, and without sacrificing precision. Our model can handle large-scale scenes with an arbitrary number of images per point taken at any position (with camera pose information), which corresponds to a standard industrial operational setting [26, 48, 58]. Using only standard 2D and 3D backbone networks, we set a new state-of-the-art for the S3DIS and KITTI-360 datasets. Our method improves on both standard and hybrid 2D/3D approaches without requiring point cloud colorization, mesh reconstruction, or depth sensors. In this paper, we present a novel and modular multi-view aggregation method for semantizing hybrid 2D/3D data based on the viewing conditions of 3D points in images. Our approach combines the following advantages:

- We set a new state-of-the-art for S3DIS 6-fold (74.7 mIoU), and KITTI-360 Test (58.3 mIoU) without using points' colorization.
- Our point-pixel mapping operates directly on 3D point clouds and images without requiring depth maps, meshing, colorization, or virtual view generation.
- Our efficient GPU-based implementation handles arbitrary numbers of 2D views and large 3D point clouds.

2. Related Work

Attention-Based Modality Fusion. Methods using attention mechanisms to learn multi-modal representation have attracted a lot of attention, in particular for combining textual and visual information [11, 23, 30] as well as videos [27, 42]. Closer to our setting, Lu *et al*. [43] use an attention scheme to select the most relevant parts of an image for visual question answering. Li *et al*. [40] define a two-branch attention-based modality fusion network merging 2D semantic and 3D occupancy for scene completion. Such work confirms the relevance of using attention for learning multi-modal representations.

2D/3D Scene Analysis with Deep Learning. Over the last few years, deep networks specifically designed to handle the 3D modality have reached impressive degrees of performance and maturity, see the review of Guo *et al*. [24]. Recent work [19, 29, 32] propose to use a dedicated 3D network for processing point clouds, while a 2D convolutional network extracts radiometric features which are projected to the point cloud. These methods require the true depth of each pixel to compute the point-pixel mapping, which makes them less applicable in a real-world setting. SnapNet [8], as well as more recent work [15, 36] generate virtual views processed by a 2D network and whose predictions are then projected back to the point cloud. These approaches, while performing well, require a costly mesh reconstruction preprocessing step to generate meaningful images. Some approaches [25, 35] fuse RGB and range images, which requires dedicated sensors and can not handle multiple views with occlusions. Existing hybrid 2D/3D methods rely on a fixed number of images per point chosen with heuristics such as the maximization of unseen points [19, 32, 36]. Then, the different views are merged using pooling operations (max [19, 54] or sum-pool [32]) or based on the 2D features' content [29]. To the best of our knowledge, no method has yet been proposed to leverage the viewing conditions for multi-view aggregation for the semantic segmentation of large scenes.

View Selection. The problem of selecting and merging the best images for a 3D scene has been extensively studied for surface reconstruction and texturing. Images are typically chosen according to the viewing angle with the surface normal [7, 39], proximity and resolution [4, 10], geometric and visibility priors [51], as well as *crispness* [22], and consistency with respect to occlusions [59]. While most of these criteria do not directly apply to point clouds, they illustrate the importance of camera pose information for selecting relevant images.

Related to our setting is the *Next Best View* selection problem [52], which consists in planning the camera position giving the *most information* about an object of interest [17]. This criterion takes different meanings according to the setting, such as the number of unseen voxels [57], diversity [45], information-theoretic measures of uncertainty [31], or can be directly learned end-to-end [44, 63]. Our setting differs in that the images have already been acquired, and the task is to choose which one contains the most relevant information for each point. We draw inspiration from the end-to-end approaches demonstrating that a neural network can assess the quality of information contained in an image from pose information.

The problem of view selection is also addressed in the literature on shape recognition [54]. Features from different images can be merged based on their similarity [60], *discriminativity* [21], or using patch matching schemes [64, 66] or graph-neural networks [61]. Some methods use 3D fea-

tures [65] or camera position [34] to select the best views, but no technique yet makes explicit use of the viewing configuration. Furthermore, these methods operate on synthetic views of artificial shapes, which differs from our goal of analyzing large scenes with images in arbitrary poses.

Closer to our problem, Armeni *et al*. [5] aggregate views using handcrafted heuristics. Bozic *et al*. [9] use a distance-aware attentive view aggregation for 3D reconstruction, but disregard other viewing conditions.

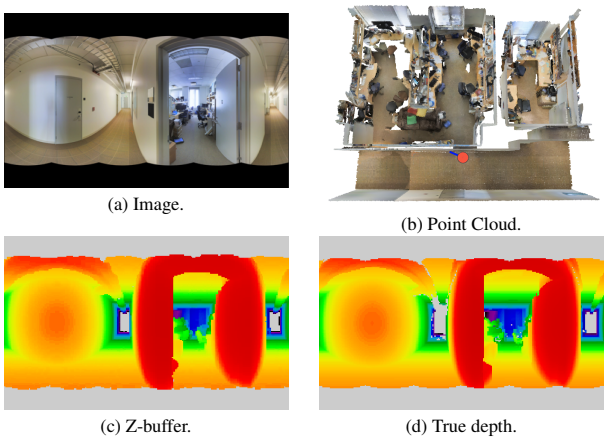
3. Method

Let P be a set of 3D points and I a collection of co-registered images, all acquired from the same scene. We characterize points by their position in space, and images by their pixels' RGB values along with intrinsic and extrinsic camera parameters. Our goal is to exploit the correspondence between points and image pixels to perform 3D point cloud semantic segmentation with features learned from both modalities. Our method starts by computing an occlusion-aware mapping between 3D points and pixels, then uses viewing conditions through an attention scheme to aggregate relevant image features for each 3D point. This approach can be easily integrated into a standard 3D network architecture, allowing us to learn from both point clouds and images simultaneously in an end-to-end fashion.

3.1. Point-Image Mapping

We start by efficiently computing a mapping between the images of I and the points of P. We say that a point-image pair $(p, i) \in P \times I$ is *compatible* if p is visible in i, *i.e*. p is in the frustum of i and not occluded. For such a pair, we define the re-projection $\text{pix}(p, i)$ as the pixel of i in which p is *visible*. Note that as points are zero-dimensional objects (zero-volume), $\text{pix}(p, i)$ is a single pixel. We denote by $v(p)$ the *views* of p, *i.e*. the set of images in which p is visible.

Point-Pixel Mapping Construction. We operate in a general *in the wild* multi-view setting in which the optical axes of the cameras and the 3D sensor are not necessarily aligned. Consequently, computing the point-image mapping requires a *visibility model* to detect occlusions. This can be done by computing a full mesh reconstruction from the point clouds or by using a depth map obtained by a camera-aligned depth sensor or other means. In contrast, we propose an efficient implementation of the straightforward Z-buffering method [53] to compute the mapping directly from images and point clouds. For each image $i \in I$, we replace all 3D points in the frustum of i under a pre-determined distance by a square plane section facing towards i and whose size depends on their distance to the sensor and the resolution of the point cloud. We can compute the projection mask—or *splat*—of each square onto i using the camera parameters of i. We iteratively accumulate all splats in a depth map called Z-buffer by keeping track of the closest point-camera distance for each pixel. Simultaneously, we store corresponding point indices in an index map, along with other relevant point attributes. Once all splats have been accumulated, *visible* points are the ones whose indices appear in the index map. For each visible point p, we set $\text{pix}(p, i)$ as the pixel of i in which p itself is projected. Our GPU-accelerated implementation can process the entire S3DIS dataset [6] subsampled at 5cm (12 million points and 1413 high-resolution equirectangular images) within 65 seconds. See Figure 2 for an illustration, and the Appendix for more details on the mapping computation, alternative visibility models, and our memory-efficient implementation for large-scale mappings.



(a) Image. (b) Point Cloud. (c) Z-buffer. (d) True depth.

Figure 2. **Mapping Computation.** We estimate pixel depth for all (a) images using the co-registered (b) point cloud. We compute (c) Z-buffers with an efficient GPU-accelerated implementation, resulting in depth maps comparable to the (d) true distance given by camera-aligned depth sensors. We use our estimated depth maps to compute point-image mappings. Better seen on a monitor.

Viewing Conditions. To each compatible point-image pair (p, i), we associate a D-dimensional vector $o_{(p,i)}$ describing the conditions under which the point p is seen in i. In practice, we define this vector as a set of $D = 8$ handcrafted features qualifying the observation conditions of (p, i): (i) normalized depth, (ii-iv) local geometric descriptors (linearity, planarity, scattering) (v) viewing angle w.r.t. the estimated normal, (vi) row of the pixel, (vii) local density, (viii) occlusion rate. See the Appendix for more details on the computation and impact of these values.

3.2. Learning Multi-View Aggregation

We denote by $\{f_i^{2D}\}_{i \in I}$ a set of 2D feature maps of width C associated to the images I, typically obtained with a convolutional neural network (CNN). Our goal is to transfer these features to the 3D points by exploiting the correspondence between points and images. However, not all viewing images contain equally relevant information for a given 3D point. We propose an attention-based approach to weigh and

aggregate features from the viewing images for each point p.

View Features. The mapping $\text{pix}(p, i)$ described in Section 3.1 allows us to associate image features to each compatible point-image pair (p, i):

$$\tilde{f}^{\text{2D}}_{(p,i)} = \phi_0 \left(f_i^{\text{2D}} \left[\text{pix}(p, i) \right] \right) , \quad (1)$$

with $\phi_0 : \mathbb{R}^C \mapsto \mathbb{R}^C$ a Multi-Layer Perceptron (MLP). Learned image features can contain information of different natures: contextual, textural, class-specific, and so on. To reflect this consideration, we split the channels of $\tilde{f}^{\text{2D}}_{(p,i)}$ into K contiguous blocks of $\lfloor C/K \rfloor$ channels:

$$\tilde{f}^{\text{2D}}_{(p,i)} = \left[\tilde{f}^{\text{2D}}_{(p,i),1}, \cdots, \tilde{f}^{\text{2D}}_{(p,i),K} \right] . \quad (2)$$

with $[\,\cdot\,]$ the channel-wise concatenation operator. Each block of channels represents a subset of the image information contained in $\tilde{f}^{\text{2D}}$.

View Quality. The conditions under which a point is seen in an image can be more or less conducive to certain types of information, see Figure 3. For example, an image viewing a point from a distance may give important contextual cues, while an image taken close and at a straight angle may give detailed textural information. In contrast, an image in which a point is seen from a slanted angle or under high distortion may not contain relevant information and may need to be discarded. To model these complex dependencies, we propose to predict for each compatible point-image pair (p, i) a set of K *quality scores* $x^k_{(p,i)} \in \mathbb{R}$ from its viewing conditions $o_{(p,i)}$ defined in Section 3.1. The quality $x^k_{(p,i)}$ represents the relevance for point p of the information contained in the feature block k of image i .

For each point p, we consider the set $v(p)$ of images in which it is visible. We propose to learn to predict the view quality $x^k_{(p,i)}$ for each feature block k by considering all images $i \in v(p)$ *simultaneously*. Indeed, the relevance of an image can depend on the context of the other views. For example, while a given image may provide less-than-perfect viewing conditions of a given 3D point, it may be the only available image with global information of the point's context. We use a deep set architecture [67] to map the set of viewing conditions $\{o_{(p,i)}\}_{i \in v(p)}$ to a vector of size K:

$$z_{(p,i)} = \phi_1(o_{(p,i)}) \quad (3)$$

$$x_{(p,i)} = \phi_3 \left(\left[z_{(p,i)}, \phi_2 \left(\max\{z_{(p,i)}\}_{i \in v(p)} \right) \right] \right) , \quad (4)$$

with $\phi_1 : \mathbb{R}^D \mapsto \mathbb{R}^M$, $\phi_2 : \mathbb{R}^M \mapsto \mathbb{R}^M$, and $\phi_3 : \mathbb{R}^{2M} \mapsto \mathbb{R}^K$ three MLPs, M the size of the set embedding, and max the channelwise maximum operator for a set of vectors.

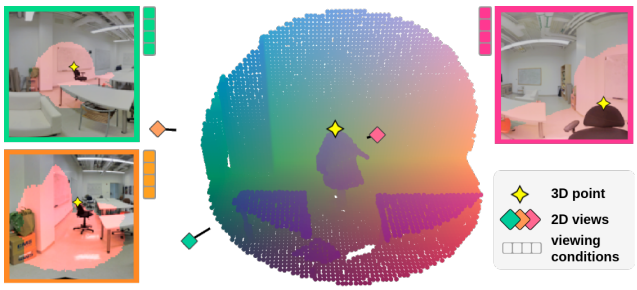


Figure 3. **Multi-View Information.** A 3D point ✦ is seen in several images with different insights. Here, the **green** image contains contextual information, while the **pink** image captures the local texture. The **orange** image sees the point at a slanted angle and may contain no additional relevant information.

View Attention Scores. We can now compute K attention scores $a^k_{(p,i)}$ in $[0, 1]$ corresponding to the relative relevance for point p of the kth feature block of image i. The attentions are obtained by applying a softmax function to the quality scores $x^k_{(p,i)}$ across the images in $v(p)$. To account for the possibly varying number of views per point, we scale the softmax according to the number of images seeing the point p :

$$a^k_{(p,i)} = \text{softmax} \left(\frac{1}{\sqrt{|v(p)|}} \left\{ x^k_{(p,i)} \right\}_{i \in v(p)} \right) . \quad (5)$$

View Gating. A limitation of using a softmax in this context is that the attention scores $\tilde{a}^k_{(p,i)}$ always sum to 1 over $v(p)$ regardless of the overall quality of the image set. Because of occlusion or limited viewpoints, some 3D points may not be seen by any relevant image for a given feature block k (*e.g.* no close or far images). In this case, it may be beneficial to discard an information block from all images altogether and purely rely on geometry. This allows the 2D network to learn image features without accounting for potentially spreading corrupted information to points with dubious viewing conditions. To this end, we introduce a gating parameter g^k_p whose role is to block the transfer of the features block k if the overall quality of the image set $v(p)$ is too low:

$$g^k_p = \text{ReLU} \left(\tanh \left(\alpha_k \max_{i \in v(p)} \left(x^k_{(p,i)} \right) + \beta_k \right) \right) , \quad (6)$$

with $\alpha, \beta \in \mathbb{R}^K$ trainable parameters and ReLU the rectified linear activation [46]. If all quality scores $x^k_{(p,i)}$ are negative for a given point p and block k, the gating parameter g^k_p will be exactly zero and block possibly detrimental information due to sub-par viewing conditions.

Attentive Image Feature Pooling. For each point p seen in one or more images, we merge the feature maps $\tilde{f}^{\text{2D}}_{(p,i)}$

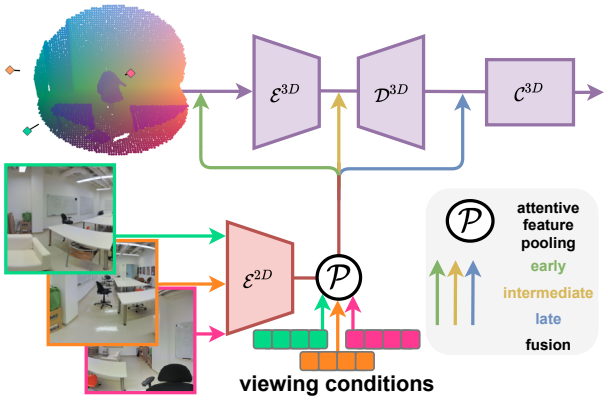


Figure 4. **Bimodal 2D/3D Architecture.** Using our multi-view aggregation module, we combine a 2D convolutional encoder $\mathcal{E}^{2D}$ and a 3D network composed of an encoder $\mathcal{E}^{3D}$, a decoder $\mathcal{D}^{3D}$, and a classifier $\mathcal{C}^{3D}$. We associate relevant 2D features to each 3D point according to their viewing conditions in each compatible image. We propose three different 2D/3D fusion strategies: early (our choice in the experiments), intermediate, and late fusion.

from each view (p, i). For each block k, we compute the sum of the view features $\tilde{f}^{2D}_{(p,i),k}$ weighted by their respective attention scores $a^k_{(p,i)}$ and multiplied by the gating parameter g^k_p. The combined image feature $\mathcal{P}(f^{2D}, p)$ associated to point p is then defined as the channelwise concatenation of the resulting tensors for all blocks:

$$\mathcal{P}(f^{2D}, p) = \left[g^k_p \sum_{i \in v(p)} a^k_{(p,i)} \tilde{f}^{2D}_{(p,i),k} \right]_{k=1}^{K} . \quad (7)$$

3.3. Bimodal Point-Image Network.

We can use the multi-view feature aggregation method described above to perform semantic segmentation of a point cloud and co-registered images by combining a network operating on 3D point clouds and 2D CNN.

Fusion Strategies. We use a 2D fully convolutional network to compute pixel-wise image feature maps f^{2D}. We also consider a 3D deep network following the classic U-Net architecture [50] and composed of three parts: (i) an encoder $\mathcal{E}^{3D}$ mapping the point cloud into a set of 3D feature maps at different resolution (*innermost* map and skip connections); (ii) a decoder $\mathcal{D}^{3D}$ converting these maps into a 3D feature map at the highest resolution (iii) a classifier $\mathcal{C}^{3D}$ associating to each point a vector for class scores of size N, the number of target classes.

As shown in Figure 4, we investigate three classic fusion schemes [25, 32, 35], connecting the image features at different points of the 3D network: (i) directly with the raw 3D features before $\mathcal{E}^{3D}$ (*early fusion*), (ii) in the skip connections (*intermediate fusion*) (iii) between the decoder $\mathcal{D}^{3D}$ and the classifier $\mathcal{C}^{3D}$ (*late fusion:*). See the Appendix for the details and equations for these fusion schemes.

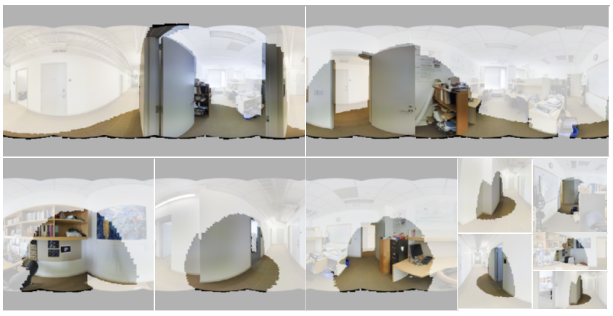


Figure 5. **Dynamic Batching.** We can improve the quantity of information contained in each training batch by cropping images around the sampled point clouds. We represent a set of 10 images with different crop size fitting in a budget of pixels corresponding to 4 full-size images.

Dynamic-Size Image-Batching The number of images $v(p)$ in which a point p is visible can vary significantly. Furthermore, when dealing with large-scale scenes, only a subset P_{sample} of the 3D scene is typically processed at once (*e.g.* spherical sampling). For this reason, the part of an image i for which points of P_{sample} are visible can sometimes be only a small fraction of the entire image. This will typically occur with equirectangular images or when P_{sample} is far away from i. We use the adaptive batching scheme depicted in Figure 5 to stabilize memory usage across batches and avoid needless computations on excessively large images. The first step is to crop each image using the smallest window across a fixed set of sizes (*e.g.* 64×64, 128×64, etc.) such that the crop contains the bounding box of all seen points of P with a given margin. Observing that the memory consumption of a fully convolutional encoder is linear w.r.t. the number of input pixels, we allocate to each point cloud in the batch a *budget* of pixels. Images are then chosen randomly by iteratively selecting images with a probability proportional to their number of pixels and to the number of newly seen points in the cloud, until the pixel budget is spent. Finally, the images are organized into different batches according to their sizes, allowing for their simultaneous processing. Note that at inference time, we can take batches as large as the GPU memory allows.

3.4. Implementation Details

We use sparse encoding for mappings in order to only store compatible point-image pairs. This proves necessary for the large scale, in-the-wild setting with varying number of images seeing each point. Our code is available at `https://github.com/drprojects/DeepViewAgg`, the exact network and training configurations are given in the Appendix, and all the metrics of all runs can be ac-

(a) S3DIS (b) ScanNet (c) KITTI-360.

Figure 6. **Datasets.** Illustration of the sampling procedure for all considered datasets with point clouds alongside some of the available images. The 3D components of batches are constituted of spheres for (a) S3DIS, rooms for (b) ScanNet, and cylinders for (c) KITTI-360.

cessed at https://wandb.ai/damien_robert/DeepViewAgg-benchmark.

4. Experiments

We propose several experiments on public large-scale semantic segmentation benchmarks to demonstrate the benefits of our deep multi-view aggregation module (DeepViewAgg). Our approach yields significantly better results than our 3D backbone directly operating on colorized point clouds. We set a new state-of-the-art for the highly contested S3DIS benchmark using only standard 2D and 3D architectures combined with our proposed module.

4.1. Datasets

S3DIS [6]. This indoor dataset of office buildings contains over 278 million semantically annotated 3D points across 6 building areas—or *folds*. A companion dataset can be downloaded at https://github.com/alexsax/2D-3D-Semantics, and contains 1413 equirectangular images. To represent our large-scale, in-the-wild setting, we merge each fold into a large point cloud and discard all room-related information. We apply minor registration adjustments detailed in the Appendix.

ScanNet [18]. This indoor dataset contains over 1501 scenes obtained from 2.5 million RGB-D images with pose information. To account for the high redundancy between images, we select one in every 50 image. This dataset deviates slightly from our intended setting as 2D and 3D are derived from the same sensors.

KITTI-360 [41]. This large outdoor dataset contains over 100k laser scans and 320k images captured with a multi-sensor mobile platform. We use one image every five from the left perspective camera. We report the classwise performance on the official withheld test set.

Table 1. **Quantitative Evaluation.** Mean Intersection-over-Union of different state-of-the-art methods on S3DIS's Fold 5 and 6-fold, ScanNet Val, and KITTI-360 Test. All methods except the last line are trained on colorized point clouds. **State-of-the-art**, second highest. [1] with 3D supervision only.

Model	S3DIS		ScanNet	KITTI
	Fold 5	6-Fold	Val	360 Test
Methods operating on colorized point clouds				
PointNet++ [49]	-	56.7 [13]	67.6 [14]	35.7 [41]
SPG+SSP [37, 38]	61.7	68.4	-	-
MinkowskiNet [16]	65.4	65.9 [13]	72.4 [47]	-
KPConv [55]	67.1	70.6	69.3 [47]	-
RandLANet [28]	-	70.0	-	-
PointTrans. [20]	**70.4**	73.5	-	-
Our 3D Backbone	64.7	69.5	69.0	53.9
Methods operating on point clouds and images				
MVPNet [32]	62.4	-	68.3	-
VMVF [36]	65.4	-	**76.4**	-
BPNet [29]	-	-	69.7[1]	-
3D Backbone+ DeepViewAgg (ours)	67.2	**74.7**	71.0	**58.3**

General Setting. All datasets provide colorized point clouds obtained with dataset-specific preprocessings. To handle the large size of scans, we define batches using a sampling strategy for S3DIS (2m-radius spheres) and KITTI-360 (6m-radius vertical cylinders), while we process ScanNet room-by-room, see Figure 6. We down-sample the point clouds for processing (S3DIS: 2cm, ScanNet: 3cm, KITTI-360: 5cm) and interpolate our prediction to full resolution for evaluation. To mitigate the memory impact of the 2D encoder, we also down-sample S3DIS images to 1024×512 but keep the full resolution for ScanNet (320×240) and KITTI-360 (1408×376).

4.2. Quantitative Evaluation

In Table 1, we compare the performance of our approach and other learning methods on S3DIS, ScanNet Validation, and KITTI-360 Test using the classwise mean Intersection-over-Union (mIoU) as metric. Our method (DeepViewAgg) uses images in the 2D encoder and *raw uncolored point clouds* in the 3D encoder. All other approaches, including our backbone (3D Backbone), use the colorized point clouds provided by the datasets.

DeepViewAgg sets a new state-of-the-art for S3DIS for all 6 folds and the second-highest performance for the 5th fold. In particular, we outperform the VMVF network [36], showing that our multi-view aggregation model can overtake methods relying on costly virtual view generation using only available images. Furthermore, VMVF uses true depth maps, colorized point clouds, normals, and room-wise normalized information. In contrast, our method only uses raw XYZ data in the 3D encoder and estimates the mappings. Our approach also overtakes the recent PointTransformer [20] (PointTrans.) by 1.2 mIoU points, even though this method outperforms our 3D backbone by 4 points on colorized points. Our model also improves the performance of our 3D backbone on the KITTI-360 test set by 4.4 points, illustrating the importance of images for both indoor and outdoor datasets alike.

While giving reasonable results, our method does not perform as well on the validation set of ScanNet comparatively. We outperform the 2D/3D fusion method of BPNet [29] when restricted to 3D annotations, illustrating the importance of view selection. We argue that the limited variety in the camera points of view of ScanNet RGB-D scans, as well as their small field-of-view and blurriness reduce the quality of the information provided by images. This is reinforced by the impressive performance of VMVF, which synthesizes its own images with controlled points of view and resolution. See Figure 7 for qualitative illustrations.

4.3. Analysis

We conduct further analyses on Fold 5 and Fold 2 of S3DIS (subsampled at 5cm for processing) and the validation set of KITTI-360 in Table 2. We added Fold 2 along the commonly used Fold 5, as it benefited most from our method, and hence is more conducive to evaluating the impact of our design choices.

Modality Combinations. As observed in Table 2, combining a 3D deep network operating on raw 3D features and a 2D network with our method (Best Configuration) improves the performance by over 6 to 15 points compared to the same 3D backbone operating on colorized point clouds alone (XYZRGB). To illustrate that point colorization is not a trivial task, we train our 3D backbone with point clouds colorized by averaging for each point the color of all pixels in which it is visible (XYZ Average-RGB). Compared to the "official" colored point clouds, we observe a drop of 1 point for Fold 5 and 1.3 point for KITTI-360, but a gain of almost 5 points for Fold 2. This shows how different point cloud colorization schemes can yield vastly different results. Not using any radiometric information and purely relying on 3D points without color (XYZ) decreases the score of XYZRGB by a further 3 to 4 points on S3DIS. For KITTI-360, XYZ outperforms XYZ Average-RGB, suggesting that poor colorization can even be detrimental.

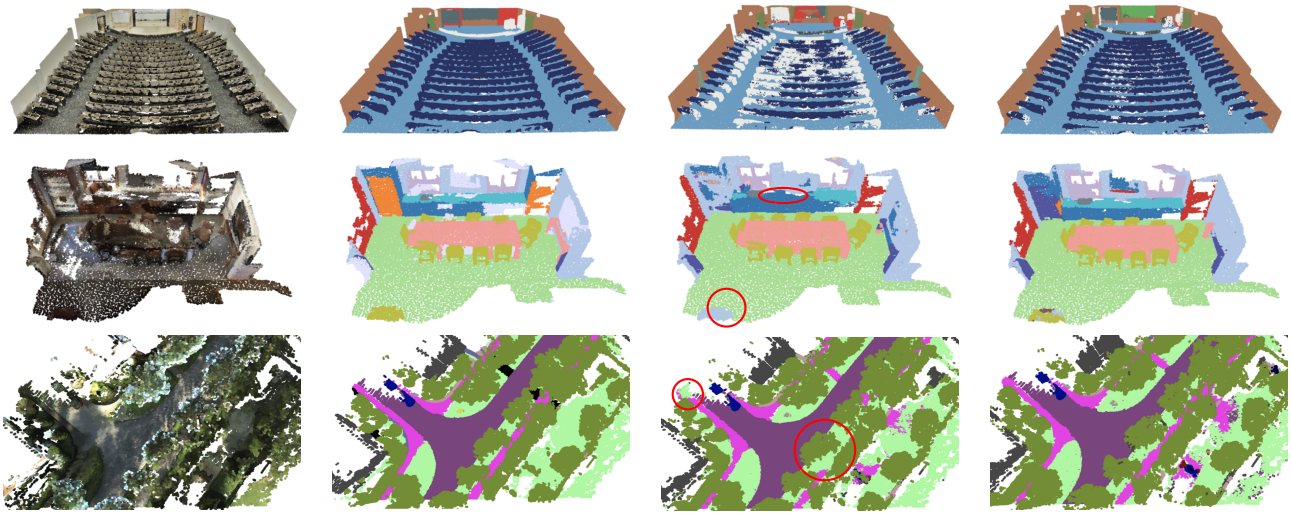
Table 2. **Ablation Study.** Mean IoU comparison of different modalities and design choices on Fold 2 and Fold 5 of S3DIS down-sampled at 5cm for processing and KITTI-360 Val.

Model	S3DIS		KITTI
	Fold 2	Fold 5	360 Val
Best Configuration	63.2	67.5	57.8
Modality Combinations			
XYZRGB	-15.9	-6.0	-3.6
XYZ Average-RGB	-10.8	-7.0	-4.9
XYZ	-19.5	-9.5	-4.1
Pure RGB	-5.3	-5.4	-14.5
Lower Image Resolution	-5.9	-0.8	-0.7
Higher 3D Resolution	-1.0	-0.3	-
Design Choices			
Late Fusion	-9.1	-1.0	-1.3
Only One Group	-4.8	-0.8	-0.4
No Gating	-3.0	-0.4	-1.1
No Dynamic Batch	-6.9	-1.9	-4.5
No Pre-training	-7.2	-6.7	-3.7
MaxPool	-0.8	-1.5	-2.9
Smaller 3D backbone	-0.5	-0.7	+0.7

We also evaluate a scheme in which the 3D network is entirely removed, and 3D points are classified solely based on features coming from a 2D encoder-decoder and our view aggregation module, without any 3D convolution (Pure RGB). This method outperforms even (XYZRGB) for S3DIS, illustrating the relevance of images for point cloud segmentation. On KITTI-360, as many 3D points are not seen by the cameras used, this approach perform worse.

Training our best 2D+3D model with images downsampled by a factor of 2 (Lower Image Resolution) brings a large performance drop. In contrast, using 2cm of 3D resolution instead of 5cm (Higher 3D Resolution) has little impact for S3DIS. We conclude that when the images already contain fine-grained information, the impact of the resolution of the 3D voxel grid decreases.

Design Choices. Using late fusion (Late Fusion) instead of early fusion gives comparable results on Fold 5 and KITTI-360, but significantly worse for Fold 2 for which the gain of using images is more pronounced. Using only one feature group (Only One Group, $K = 1$ in Equation 2) results in a drop of 4.8 points for Fold 2, highlighting that our method



(a) Colorized Point Cloud (b) Ground Truth (c) 3D Backbone Predictions (d) Our Model Predictions.

Figure 7. **Qualitative illustration.** Scenes from our considered datasets (top: S3DIS, middle: ScanNet, bottom: KITTI-360) with (a) colorized point clouds, (b) ground truth point annotations, (c) prediction of the backbone network operating on the colorized point cloud, and (d) our method operating on raw uncolored point clouds and images. Our approach is able to use images to resolve cases in which the geometry is ambiguous or unusual, such as a large amphitheater with tiered rows of seats (top row). Color legend given in the Appendix.

can learn to treat different types of radiometric information specifically. Removing the gating mechanism (No Gating, see Equation 6) decreases the IoU by 3 points for Fold 2, and 1.1 on KITTI-360. Not using dynamic batches forces us to limit ourselves to 4 full-size images per 3D sphere/cylinder, which results in performance drops of 2 to 7 points. Pre-training the 2D network on related open-access datasets (No 2D Pre-Training) accounts for up to 7 mIoU points. Not only do images contain rich radiometric information, but they also allow us to leverage the ubiquitous availability of annotated 2D datasets.

Using featurewise max-pooling to merge the views results in a drop of 1 to 1.5 points for S3DIS and 3 points for KITTI-360. This illustrates that as long as we employ proper mapping, batching, and pre-training strategies, even simple pooling operations can perform very well. However, the addition of our model appears necessary to improve the precision even further and reach state-of-the-art results.

Switching our 3D backbone to a lighter version of MinkowskiNet with decreased widths, we observe no significant impact on the prediction quality. This suggests that we could use our approach successfully with smaller models. See the Appendix for further analysis of our design choices and of the influence of viewing conditions.

Limitations. While our method does not require sensors with aligned optical axes, true depth maps, or a meshing step, we still need camera poses. In some "in the wild" settings, they may not be available or require a pose estimation and registration step which may be costly and error-prone. Our mapping computation also relies on the assumption that the 2D and 3D modalities are acquired simultaneously.

Our multi-view aggregation method operates purely on viewing conditions and does not take the geometric and radiometric features into account in the computation of attention scores. We implemented a self-attention-based approach using such features, which resulted in a significant increase in memory usage without tangible benefits: the viewing conditions appear to be the most critical factor when selecting and aggregating images features.

5. Conclusion

We proposed a deep learning-based multi-view aggregation model for the semantic segmentation of large 3D scenes. Our approach uses the viewing condition of 3D points in images to select and merge the most relevant 2D features with 3D information. Combined with standard image and point cloud encoders, our method improves the state-of-the-art for two different datasets. Our full pipeline can run on a point cloud and a set of co-registered images at arbitrary positions without requiring colorization, meshing, or true depth maps. These promising results illustrate the relevancy of using dedicated architectures for extracting information from images even for 3D scene analysis.

Acknowledgements This work was funded by ENGIE Lab CRIGEN and carried on in the LASTIG research unit of Université Paris-Est.

References

[1] Trimble TX8 3D Laser Scanner. https://geospatial.trimble.com/products-and-solutions/trimble-tx8. Accessed: 2021-11-05. 1

[2] FARO FOCUS LASER SCANNERS. https://www.faro.com/products/construction-bim/farofocus. Accessed: 2021-11-05. 1

[3] Leica RTC360 3D Laser Scanner. https://leica-geosystems.com/products/laser-scanners/scanners/leica-rtc360. Accessed: 2021-11-05. 1

[4] Cédric Allène, Jean-Philippe Pons, and Renaud Keriven. Seamless image-based texture atlases using multi-band blending. In *ICPR*. 2

[5] 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. In *ICCV*, 2019. 3

[6] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 3, 6

[7] Stan Birchfield and Carlo Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999. 2

[8] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 2018. 1, 2

[9] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular RGB scene reconstruction using transformers. *NeurIPS*, 2021. 3

[10] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001. 2

[11] Ozan Caglayan, Loïc Barrault, and Fethi Bougares. Multimodal attention for neural machine translation. *arXiv preprint arXiv:1609.03976*, 2016. 2

[12] Ziyun Cai, Jungong Han, Li Liu, and Ling Shao. Rgb-d datasets using microsoft kinect or similar sensors: a survey. *Multimedia Tools and Applications*, 2017. 1

[13] Thomas Chaton, Nicolas Chaulet, Sofiane Horache, and Loic Landrieu. Torch-points3D: A modular multi-task framework for reproducible deep learning on 3D point clouds. *3DV*, 2020. 6

[14] Dave Z. Chen. Pointnet2.scannet. https://github.com/daveredrum/Pointnet2.ScanNet, 2021. 6

[15] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H Hsu. A unified point-based framework for 3D segmentation. In *3DV*, 2019. 1, 2

[16] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 6

[17] Cl Connolly. The determination of next best views. In *ICRA*, 1985. 2

[18] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 6

[19] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *ECCV*, 2018. 1, 2

[20] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. *ICCV*, 2021. 6, 7

[21] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3D shape recognition. In *CVPR*, 2018. 2

[22] Ran Gal, Yonatan Wexler, Eyal Ofek, Hugues Hoppe, and Daniel Cohen-Or. Seamless montage for texturing models. In *Computer Graphics Forum*, 2010. 2

[23] Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. Multimodal affective analysis using hierarchical attention strategy with word-level alignment. In *Association for Computational Linguistics*, 2018. 2

[24] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2

[25] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *ACCV*. Springer, 2016. 2, 5

[26] David Hodgetts. Laser scanning and digital outcrop geology in the petroleum industry: A review. *Marine and Petroleum Geology*, 2013. 2

[27] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017. 2

[28] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020. 6

[29] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *CVPR*, 2021. 1, 2, 6, 7

[30] Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based multimodal neural machine translation. In *Conference on Machine Translation*, 2016. 2

[31] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3D reconstruction. In *ICRA*. IEEE, 2016. 2

[32] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3D scene understanding. In *CVPR Workshops*, 2019. 1, 2, 5, 6

[33] Arttu Julin, Matti Kurkela, Toni Rantanen, Juho-Pekka Virtanen, Mikko Maksimainen, Antero Kukko, Harri Kaartinen, Matti T Vaaja, Juha Hyyppä, and Hannu Hyyppä. Evaluating the quality of tls point cloud colorization. *Remote Sensing*, 2020. 1

[34] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. RotationNet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *CVPR*, 2018. 3

[35] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Fuseseg: LiDAR point cloud segmentation fusing multi-modal data. In *WACV*, 2020. 2, 5

[36] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3D semantic segmentation. In *ECCV*, 2020. 1, 2, 6, 7

[37] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *CVPR*, 2019. 6

[38] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 6

[39] Victor Lempitsky and Denis Ivanov. Seamless mosaicing of image-based texture maps. In *CVPR*, 2007. 2

[40] Siqi Li, Changqing Zou, Yipeng Li, Xibin Zhao, and Yue Gao. Attention-based multi-modal fusion network for semantic scene completion. In *AAAI*, 2020. 2

[41] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *ICCV*, 2021. 6

[42] Xiang Long, Chuang Gan, Gerard Melo, Xiao Liu, Yandong Li, Fu Li, and Shilei Wen. Multimodal keyless attention fusion for video classification. In *AAAI*, 2018. 2

[43] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. *NeurIPS*, 2016. 2

[44] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3D object reconstruction. *Pattern Recognition Letters*, 2020. 2

[45] Farzin Mokhtarian and Sadegh Abbasi. Automatic selection of optimal views in multi-view object recognition. In *BMVC*, 2000. 2

[46] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 4

[47] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-context data augmentation for 3D scenes. *3DV*, 2021. 6

[48] Massimiliano Pepe, Sebastiano Ackermann, Luigi Fregonese, and Cristiana Achille. 3D point cloud model color adjustment by combining terrestrial laser scanner and close range photogrammetry datasets. In *International Conference on Digital Heritage*, 2016. 2

[49] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 6

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MIICAI*. Springer, 2015. 5

[51] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2

[52] William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *Computing Surveys*, 2003. 2

[53] Wolfgang Strasser. Fast curve and surface generation for interactive shape design. *Computers in Industry*, 1982. 3

[54] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *CVPR*, 2015. 2

[55] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 6

[56] Robert J Valkenburg and Alan M McIvor. Accurate 3D measurement using a structured light system. *Image and Vision Computing*, 1998. 1

[57] J Irving Vasquez-Gomez, L Enrique Sucar, and Rafael Murrieta-Cid. View/state planning for three-dimensional object reconstruction under uncertainty. *Autonomous Robots*, 2017. 2

[58] Juho-Pekka Virtanen, Sylvie Daniel, Tuomas Turppa, Lingli Zhu, Arttu Julin, Hannu Hyyppä, and Juha Hyyppä. Interactive dense point clouds in a game engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2020. 2

[59] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3D reconstructions. In *ECCV*, 2014. 2

[60] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3D object recognition. *BMVC*, 2019. 2

[61] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3D shape analysis. In *CVPR*, 2020. 2

[62] Iain H Woodhouse, Caroline Nichol, Peter Sinclair, Jim Jack, Felix Morsdorf, Tim J Malthus, and Genevieve Patenaude. A multispectral canopy LiDAR demonstrator project. *Geoscience and Remote Sensing Letters*, 2011. 1

[63] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[64] Ze Yang and Liwei Wang. Learning relationships for multi-view 3D object recognition. In *ICCV*, 2019. 2

[65] Haoxuan You, Yifan Feng, Xibin Zhao, Changqing Zou, Rongrong Ji, and Yue Gao. PVRNet: Point-view relation neural network for 3D shape recognition. In *AAAI*, 2019. 3

[66] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3D object recognition. In *CVPR*, 2018. 2

[67] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *NeurIPS*, 2017. 4