MESHMVS: MULTI-VIEW STEREO GUIDED MESH RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning based 3D shape generation methods generally utilize latent features extracted from color images to encode the objects' semantics and guide the shape generation process. These color image semantics only implicitly encode 3D information, potentially limiting the accuracy of the generated shapes. In this paper we propose a multi-view mesh generation method which incorporates geometry information in the color images explicitly by using the features from intermediate 2.5D depth representations of the input images and regularizing the 3D shapes against these depth images. Our system first predicts a coarse 3D volume from the color images by probabilistically merging voxel occupancy grids from individual views. Depth images corresponding to the multi-view color images are predicted which along with the rendered depth images of the coarse shape are used as a contrastive input whose features guide the refinement of the coarse shape through a series of graph convolution networks. Attention-based multi-view feature pooling is proposed to fuse the contrastive depth features from different viewpoints which are fed to the graph convolution networks.

We validate the proposed multi-view mesh generation method on ShapeNet, where we obtain a significant improvement with 34% decrease in chamfer distance to ground truth and 14% increase in the F1-score compared with the state-of-the-art multi-view shape generation method.

1 INTRODUCTION

3D shape generation is a long-standing research problem in computer vision and computer graphics with applications in autonomous driving, augmented reality, etc. Conventional approaches mainly leverage multi-view geometry based on stereo correspondences between images but are restricted by the coverage provided by the input views. With the availability of large-scale 3D shape datasets and the success of deep learning in several computer vision tasks, 3D representations such as voxel grid Choy et al. (2016); Tulsiani et al. (2017); Yan et al. (2016) and point cloud Yang et al. (2018); Fan et al. (2017) have been explored for single-view 3D reconstruction. Among them, triangle mesh representation has received the most attention as it has various desirable properties for a wide range of applications and is capable of modeling detailed geometry without high memory requirement.
Single-view 3D reconstruction methods Wang et al. (2018); Huang et al. (2015); Kar et al. (2015); Su et al. (2014) generate the 3D shape from merely a single color image but suffer from occlusion and limited visibility which leads to low quality reconstructions in the unseen areas. Multi-view methods Wen et al. (2019); Choy et al. (2016); Kar et al. (2017); Gwak et al. (2017) extend the input to images from different viewpoints which provides more visual information and improves the accuracy of the generated shapes. Recent work in multi-view mesh reconstruction Wen et al. (2019) introduces a multi-view deformation network using perceptual feature from each color image for refining the meshes generated by Pixel2Mesh Wang et al. (2018). Although promising results were obtained, this method relies on perceptual features from color images which do not explicitly encode the objects' geometry and could restrict the accuracy of the 3D models.

In this work, we present a novel multi-view mesh generation method where we start by predicting coarse volumetric occupancy grid representations for the color images of each input viewpoint independently using a shared fully convolutional network which are merged into a single voxel grid in a probabilistic fashion followed by `cubify` Gkioxari et al. (2019) operation to convert it to a triangle

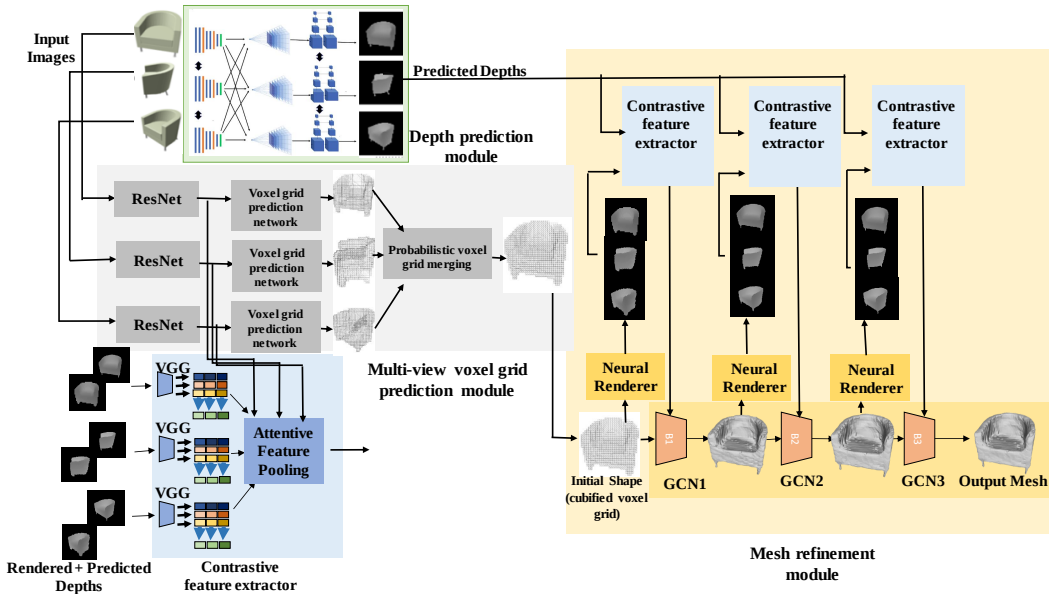

Figure 1: **Architecture of the proposed method**. The *voxel grid prediction module* predicts coarse voxel grid representation which is further refined by a series of *GCN*s. The GCNs use *contrastive depth features* from rendered depths of the current shape and the predicted depths from *MVSNet*. Multi-view features are pooled using a multi-head attention mechanism.

mesh. We then use Graph Convolutional Network (GCN) Scarselli et al. (2008); Wang et al. (2018) to fine-tune the cubified voxel grid in a coarse-to-fine manner. The GCN refines the coarse mesh by using the feature vector of each graph node (mesh vertices) obtained by projecting the vertices on the 2D contrastive depth features. The contrastive depth features are extracted from the rendered depth maps of the current mesh and predicted depth maps from a multi-view stereo network. We also propose an attention-based method to fuse feature from multiple views that can learn the importance of different views for each of the mesh vertices. Constrains between the intermediate refined mesh from GCN with predicted depth maps of different viewpoints further improve the final mesh quality. By employing multi-view voxel grid generation and refining it using geometry information from both the current mesh (through the rendered depth maps) and predicted depth maps, we are able to generate high-quality meshes. We validate our method on the ShapeNet Chang et al. (2015) benchmark and our method achieves the best performance among all previous multi-view and single-view mesh generation methods.

2 Related Work

2.1 Traditional Shape Generation Methods

3D model generation has traditionally been tackled using multi-view geometry principles. Among them, structure-from-motion (SfM) Schonberger & Frahm (2016) and simultaneous localization and mapping (SLAM) Cadena et al. (2016) are popular techniques that perform 3D reconstruction and camera pose estimation at the same time. Closer to our problem setup, multi-view stereo methods infer 3D geometry from images with known camera parameters. Volumetric methods Kar et al. (2017); Kutulakos & Seitz (2000); Seitz & Dyer (1999) predict voxel grid representation of objects by estimating the relationship between each voxel and object surfaces. Point cloud based methods Furukawa & Ponce (2009); Lhuillier & Quan (2005) start with a sparse point cloud and gradually increase the density of points to obtain a final dense point cloud of the object. Durou et al. (2008); Zhang et al. (1999); Favaro & Soatto (2005) reason about shading, texture and defocus to reason about visible parts of the object and infer its 3D geometry. While the results of these works are impressive in terms of quality and completeness of reconstruction, they still struggle with poorly textured and non-reflective surfaces and require carefully selected input views.

2.2 DEEP SHAPE GENERATION METHODS

Deep learning based approaches can learn to infer 3D structure from training data and can be robust against poorly textured and reflective surfaces as well as limited and arbitrarily selected input views. These methods can be categorized into single view and multi-view methods. Huang et al. (2015); Su et al. (2014) use shape component retrieval and deformation from a large dataset for single-view 3D shape generation. Kurenkov et al. (2018) extend this idea by introducing free-form deformation networks on retrieved object templates from a database. Some work learn shape deformation from ground truth foreground masks of 2D images Kar et al. (2015); Yan et al. (2016); Tulsiani et al. (2017). Recurrent Neural Networks (RNN) based methods Choy et al. (2016); Kar et al. (2017); Gwak et al. (2017) are another popular solution to solve this problem. Gwak et al. (2017); Lin et al. (2019) introduce image silhouettes along with adversarial multi-view constraints and optimize object mesh models using multi-view photometric constraints. Predicting mesh directly from color images was proposed in Wang et al. (2018); Wickramasinghe et al. (2019); Pan et al. (2019); Wen et al. (2019); Gkioxari et al. (2019); Tang et al. (2019). DR-KFS Jin et al. (2019) introduces a differentiable visual similarity metric while SeqXY2SeqZ Han et al. (2020) represents 3D shapes using a set of 2D voxel tubes for shape reconstruction. Front2Back Yao et al. (2020) generates 3D shapes by fusing predicted depth and normal images and DV-Net Jia et al. (2020) predicts dense object point clouds using dual-view RGB images with a gated control network to fuse point clouds from the two views.

2.3 DEPTH ESTIMATION

Compared to 3D shape generation, depth prediction is an easier problem formulation since it simplifies the task to per-view depth map estimation. Deep learning based multi-view stereo depth estimation was first introduced in Hartmann et al. (2017) where a learned cost metric is used to estimate patch similarities. DeepMVS Huang et al. (2018) warps multi-view images to 3D space and then applies deep networks for regularization and aggregation to estimate depth images. Learned 3D cost volume based depth prediction was proposed in MVSNet Yao et al. (2018) where a 3 dimensional cost volume is built using homographically warped 2D features from multi-view images and 3D CNNs are used for cost regularization and depth regression. This idea was further extended by Chen et al. (2019); Luo et al. (2019); Gu et al. (2019); Yao et al. (2019).

3 METHODOLOGY

Figure 1 shows the architecture of the proposed system which takes as input multi-view color images of an object with known poses and outputs a triangle mesh representing the surface of the object.

3.1 MULTI-VIEW VOXEL GRID PREDICTION

Single-view Voxel Grid Prediction The single-view voxel branch consists of a ResNet feature extractor and a fully convolutional voxel grid prediction network. It generates the coarse initial shape of an object from one viewpoint as voxel occupancy grid using a color image. Here, we set the resolution of the generated voxel occupancy grid as $32 \times 32 \times 32$. The voxel prediction networks for all viewpoints share the same weights.

Probabilistic Occupancy Grid Merging Voxel occupancy grid predicted from a single viewpoint suffers from occlusion and limited visibility. In order to fuse voxel grids from different viewpoints, we propose a probabilistic occupancy grid merging method which merges the voxel grids from each input viewpoint probabilistically to obtain the final voxel grid output. This allows occluded regions in one view to be estimated from other views where those regions are visible as well as increase the confidence of prediction in overlapping regions. Occupancy probability of each voxel is represented by $p(x)$ which is converted to log-odds (logit):

$$l(x) = log\frac{p(x)}{1 - p(x)} \tag{1}$$

Bayesian update on the probabilities reduce to simple summation of log likelihoods Konolige (1997). Hence, the multi-view log-odds of a voxel is given by:

$$l(x) = l_1(x) + l_2(x) + ... + l_n(x) \tag{2}$$

where l_i is the voxel's log-odds in view i and n is the number of input views. The final voxel probability x is obtained by applying the inverse function of Equation (1) which is a sigmoid function.

3.2 MESH REFINEMENT

The `cubified` mesh from the voxel branch only provides a coarse reconstruction of the object's surface. We apply graph convolutional networks which represent each mesh vertex as one graph node and deforms them to more accurate positions.

GCN-based Mesh Deformation The features pooled from multi-view images along with 3D coordinates of the vertices in world frame are used as features of the graph nodes. Series of Graph-based Convolutional Network (GCN) blocks are applied to deform a mesh at the current stage to the next stage, starting with the `cubified` voxel grids. A graph convolution deforms mesh vertices by propagating features from neighboring vertices by applying $f_i^{'} = ReLU(W_0 f_i + \sum_{j \in \mathcal{N}(i)} W_1 f_j)$ where $\mathcal{N}(i)$ is the set of neighboring vertices of the *i*-th vertex in the mesh, $f_{\{\}}$ represents the feature vector of a vertex, and W_0 and W_1 are learnable parameters of the model. Each GCN block utilizes several graph convolutions to transform the vertex features along with a final *vertex refinement* operation where the features along with vertex coordinates are further transformed as $v_i^{'} = v_i + tanh(W_{vert}[f_i; v_i])$ where the matrix W_{vert} is another learnable parameter to obtain the deformed mesh.

Contrastive Depth Feature Extraction Yao et al. (2020) demonstrate that using intermediate, image-centric 2.5D representations instead of directly generating 3D shapes in global frame from raw 2D images can improve 3D reconstruction quality. We therefore propose to formulate the features for graph nodes using 2.5D depth maps as input additional inputs alongside the RGB features. Specifically, we render the meshes at different GCN stages to depth image at all the input views using Kato et al. (2018) and use them along with predicted depths for depth feature extraction. We call this form of depth input `contrastive depth` as it contrasts the rendered depths of the current mesh against the predicted depths and allows the network to reason about the deformation better than when using predicted depth or color images alone. Given the 2D features, corresponding feature vectors of individual vertices can be found by projecting the 3D vertex coordinates to the feature planes using known camera parameters. We use VGG-16 Simonyan & Zisserman (2014) as our contrastive depth feature extraction network.

Multi-View Depth Estimation We extend MVSNet Yao et al. (2018) and predict the depth maps of all views since the original implementation predicts depth of only one reference view. This is achieved by transforming the feature volumes to each view's coordinate frame using homography warping and applying identical cost volume regularization and depth regression on each view. Detailed network architecture diagram of this module is provided in the appendix.

Attention-based Multi-View Feature Pooling In order to fuse multi-view contrastive depth features, we formulate an attention module by adapting multi-head attention mechanism originally designed for sequence to sequence machine translation using transformer (encoder-decoder) architecture Vaswani et al. (2017). In a transformer architecture the encoder hidden state is mapped to lower dimension key-value pairs (**K**, **V**) while the decoder hidden state is mapped to a query vector **Q** using independent fully connected layers. The encoder hidden state in our case is the multi-view features while the decoder hidden state is the *mean* of the multi-view features. The attention weights are computed using scaled-dot product:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{N}})\mathbf{V} \tag{3}$$

where N is the number of input views.

Multiple attention *heads* are used which are concatenated and transformed to obtain the final output

$$head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{4}$$

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1; ...; head_h]\mathbf{W}^0 \tag{5}$$

where multiple $\mathbf{W}$ are parameters to be learned, h is the number of attention heads and $i \in [1, h]$.

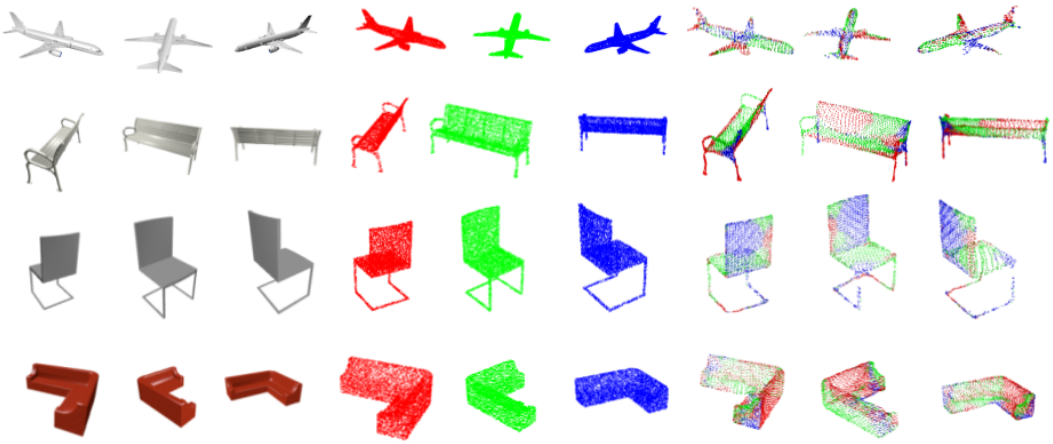

Figure 2: **Attention weights visualization.** From left to right: input images from 3 viewpoints, corresponding ground truth point clouds color-coded by their view order and the predicted mesh vertices color-coded by the attention weights of the views. Only the view with maximum attention weight is visualized for each predicted points for clarity.

We choose multi-head attention as our feature pooling method since it allows the model to attend information from different representation subspaces of the features by training multiple attentions in parallel. This method is also invariant to the order and number of input views. We visualize the learned attention weights (average of each attention heads) in Figure 2 where we can observe that the attention weights roughly takes into account the visibility/occlusion information from each view.

3.3 Loss functions

Mesh losses The losses which are derived from Wang et al. (2018) to constrain the mesh predicted by each GCN block (P) to resemble the ground truth (Q) include Chamfer distance $\mathcal{L}_{\text{chamfer}}(\text{P}, \text{Q}) = |\text{P}|^{-1} \sum_{(p,q)\in\Lambda_{\text{P,Q}}} ||p-q||^2 + |\text{Q}|^{-1} \sum_{(q,p)\in\Lambda_{\text{Q,P}}} ||q-p||^2$ and surface normal loss $\mathcal{L}_{\text{normal}}(\text{P}, \text{Q}) = -|\text{P}|^{-1} \sum_{(p,q)\in\Lambda_{\text{P,Q}}} |u_p \cdot u_q| - |\text{Q}|^{-1} \sum_{(q,p)\in\Lambda_{\text{Q,P}}} |u_q \cdot u_p|$ with additional regularization in the form of edge length loss $\mathcal{L}_{\text{edge}}(\text{V}, \text{E}) = \frac{1}{|E|} \sum_{(v,v')\in E} ||v-v'||^2$ for visually appealing results.

Depth loss Our depth prediction network is supervised using adaptive reversed Huber loss (also known as BerHu criterion) Lambert-Lacroix & Zwald (2016). $\mathcal{L}_{depth} = |x|, \text{if } |x| \leq c$, otherwise $\frac{x^2+c^2}{2c}$

Contrastive depth loss BerHu loss is also applied between the rendered depth images at different GCN stages and the predicted depth images. $\mathcal{L}_{contrastive} = |x|, \text{if } |x| \leq c$, otherwise $\frac{x^2+c^2}{2c}$

Voxel loss Binary cross-entropy loss between the predicted voxel occupancy probabilities and the ground truth occupancies is used as voxel loss to supervise the voxel predictions $\mathcal{L}_{\text{voxel}} = -\Big(p(x)log\big(p(x)\big) + \big(1-p(x)\big)log\big(1-p(x)\big)\Big)$

Final loss We use the weighted sum of the individual losses discussed above as the final loss to train our model in an end-to-end fashion. $\mathcal{L} = \lambda_{\text{chamfer}}\mathcal{L}_{\text{chamfer}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{contrastive}}\mathcal{L}_{\text{contrastive}} + \lambda_{\text{voxel}}\mathcal{L}_{\text{voxel}}$, where $\mathcal{L}$ is the final loss term.

4 Experiments

4.1 Experimental Setup

Comparisons We evaluate the proposed method against various multi-view shape generation methods. The state-of-the-art method is Pixel2Mesh++ Wen et al. (2019) (referred as *P2M++*). Wen et al. (2019) also provide a baseline by directly extending Pixel2Mesh Wang et al. (2018) to operate on multi-view images (referred as *MVP2M*) using their statistical feature pooling method to aggregate features from multiple color images. Results from additional multi-view shape generation baselines 3D-R2N2 Choy et al. (2016) and LSM Kar et al. (2017) are also reported.

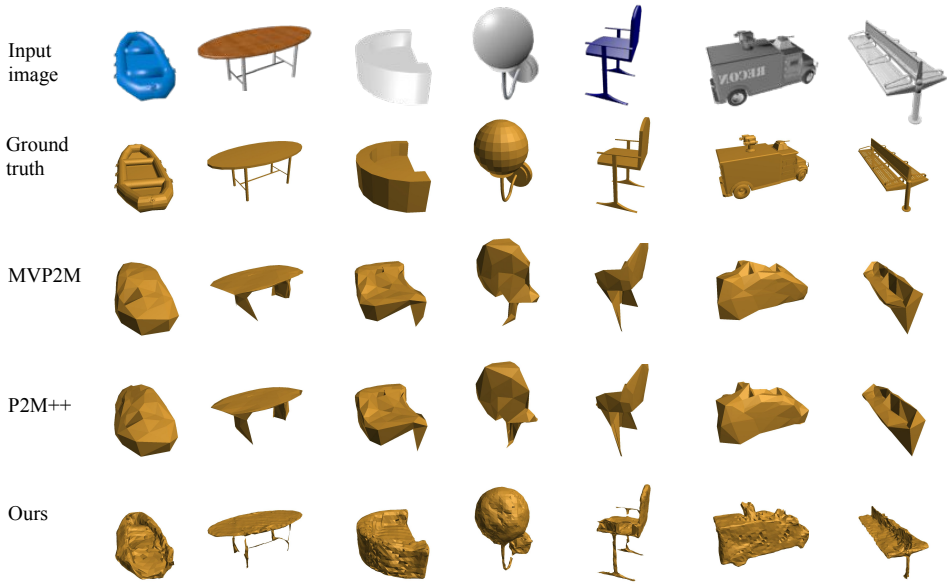

Figure 3: **Qualitative evaluation** on ShapeNet dataset. **From top to bottom**: one of the input images, ground truth mesh, multi-view extended Pixel2Mesh, Pixel2Mesh++, and ours. Our predictions are closer to the actual shape, especially for the objects with more complex topologies.

Dataset We evaluate our method against the state-of-the-art methods on the dataset from Choy et al. (2016) which is a subset of ShapeNet Chang et al. (2015) and has been widely used by recent 3D shape generation methods. It contains 50K 3D CAD models from 13 categories. Each model is rendered with a transparent background from 24 randomly chosen camera viewpoints to obtain color images. The corresponding camera intrinsics and extrinsics are provided in the dataset. Since the dataset does not contain depth images, we render them using a custom depth renderer at the same viewpoints as the color images and with the same camera intrinsics. We follow the training/testing/validation split of Gkioxari et al. (2019).

Implementation For the depth prediction module, we follow the original MVSNet Yao et al. (2018) implementation. The output depth dimensions reduces by a factor of 4 to 56×56 from the 224×224 input image. The number of depth hypotheses is chosen as 48 which offers a balance between accuracy and running/training time efficiency. These depth hypotheses represent values from 0.1 m to 1.3 m at an interval of 25 mm. These values were chosen based on the range of depths present in the dataset.

The hierarchical features obtained from "Contrastive Depth Features Extractor" are of total 4800 dimensions for each view. The aggregated multi-view features are compressed to 480 dimensional after applying attentive feature pooling. 5 attention heads are used for merging multi-view features. The loss function weights are set as $\lambda_{\text{chamfer}} = 1$, $\lambda_{\text{normal}} = 1.6 \times 10^{-4}$, $\lambda_{\text{depth}} = 0.1$, $\lambda_{\text{contrastive}} = 0.001$ and $\lambda_{\text{voxel}} = 1$. Two settings of λ_{edge} were used, $\lambda_{\text{edge}} = 0$ (referred as *Best*) which gives better quantitative results and $\lambda_{\text{edge}} = 0.2$ (referred as *Pretty*) which gives better qualitative results.

Training and Runtime The network is optimized using Adam optimizer with a learning rate of 10^{-4}. The training is done on 5 Nvidia RTX-2080 GPUs with effective batch size 5. The depth prediction network (MVSNet) is trained independently for 30 epochs. Then the whole system is trained for another 40 epochs with the weights of the MVSNet frozen. Our system is implemented in PyTorch deep learning framework and it takes around 60 hours for training.

Evaluation Metric Following Wang et al. (2018); Wen et al. (2019), we use F1-score as our evaluation metric. The F1-score is the harmonic mean of precision and recall where the precision/recall are calculated by finding the percentage of points in the predicted/ground truth that can find a nearest neighbor from the other within a threshold τ. Two values of τ are used: 10^{-4} and 2×10^{-4} m^2.

4.2 Comparison with previous Multi-view Shape Generation Methods

We quantitatively compare our method against previous works for multi-view shape generation in Table 1 and show the effectiveness of our methods in improving the shape quality. Our method

outperforms the state-of-the-art method Pixel2Mesh++ Wen et al. (2019) with a decrease in chamfer distance to ground truth by 34% and 15% increase in F1-score at threshold τ. Note that in Table 1 the same model is trained for all the categories but accuracy on individual categories as well as average over the categories are evaluated. We provide the chamfer distances in the appendix.

Category	F-score (τ) ↑						F-score (2τ) ↑					
	3D-R2N2	LSM	MVP2M	P2M++	Ours (pretty)	**Ours (best)**	3D-R2N2	LSM	MVP2M	P2M++	Ours (pretty)	**Ours (best)**
Couch	45.47	43.02	53.17	57.56	71.63	**73.63**	59.97	55.49	73.24	75.33	85.28	**88.24**
Cabinet	54.08	50.80	56.85	65.72	75.91	**76.39**	64.42	60.72	76.58	81.57	87.61	**88.84**
Bench	44.56	49.33	60.37	66.24	81.11	**83.76**	62.47	65.92	75.69	79.67	90.56	**92.57**
Chair	37.62	48.55	54.19	62.05	77.63	**78.69**	54.26	64.95	72.36	77.68	88.24	**90.02**
Monitor	36.33	43.65	53.41	60.00	74.14	**76.64**	48.65	56.33	70.63	75.42	86.04	**88.89**
Firearm	55.72	56.14	79.67	80.74	92.92	**94.32**	76.79	73.89	89.08	89.29	96.81	**97.67**
Speaker	41.48	45.21	48.90	54.88	66.02	**67.83**	52.29	56.65	68.29	71.46	79.76	**82.34**
Lamp	32.25	45.58	50.82	62.56	72.47	**75.93**	49.38	64.76	65.72	74.00	82.00	**85.33**
Cellphone	58.09	60.11	66.07	74.36	85.57	**86.45**	69.66	71.39	82.31	86.16	93.40	**94.28**
Plane	47.81	55.60	75.16	76.79	89.23	**92.13**	70.49	76.39	86.38	86.62	94.65	**96.57**
Table	48.78	48.61	65.95	71.89	82.37	**83.68**	62.67	62.22	79.96	84.19	90.24	**91.97**
Car	59.86	51.91	67.27	68.45	77.01	**80.43**	78.31	68.20	84.64	85.19	88.99	**92.33**
Watercraft	40.72	47.96	61.85	62.99	75.52	**80.48**	63.59	66.95	77.49	77.32	86.77	**90.35**
Mean	46.37	49.73	61.05	66.48	78.58	**80.80**	62.53	64.91	77.10	80.30	88.49	**90.72**

Table 1: **Qualitative comparison** against state-of-the-art multi-view shape generation methods. We report F-score on each semantic category along with the mean over all categories using two thresholds τ and 2τ for nearest neighbor match where τ=10^{-4} m^2.

We also provide visual results for qualitative assessment of the generated shapes by our *Pretty* model in Figure 3 which shows that it is able to more accurately predict topologically diverse shapes.

4.3 Ablation Studies

Contrastive Depth Feature Extraction We evaluate several methods for contrastive feature extraction (Sub-section 3.2). These methods are 1) *Input Concatenation*: using the concatenated rendered and predicted depth maps as input to the VGG feature extractor, 2) *Input Difference*: using the difference of the two depth maps as input to VGG, 3) *Feature Concatenation*: concatenating features from rendered and predicted depths extracted by shared VGG, 4) *Feature Difference*: using difference of the features from the two depth maps extracted by shared VGG, and 5) *None*: using the VGG features from the predicted depths only. The quantitative results are summarized in Table 2 and shows that *Input Concatenation* method produces better results than other formulations.

	F1-τ	F1-2τ
(1) Input Concatenation	**80.80**	**90.72**
(2) Input Difference	80.41	90.54
(3) Feature Concatenation	80.45	90.54
(4) Feature Difference	80.30	90.40
(5) None	79.40	89.95

Table 2: **Comparisons of different contrastive depth formulations**. In 1st and 2nd rows, concatenation and difference of the rendered and predicted depths are fed to VGG feature extractor while in 3rd and 4th rows, concatenation and difference of the VGG features from the depths is used for mesh refinement. 5 (None) uses VGG features from predicted depth only.

Attention Module In the 5th row and 6th row of Table 3, we present the performance of the proposed attention method against statistical feature pooling Wen et al. (2019) and a simpler attention mechanism Hu et al. (2020); Yang et al. (2020) where the pooled features are simply the weighted sum of the multi-view features. We find that the three method perform similarly on our final architecture but multi-head attention method performs better on more light-weight architectures.

Contrastive Depth Losses We also evaluate the effect of using additional regularization from contrastive depth losses: rendered depth vs. predicted depth and rendered depth vs. ground truth depth in the 3rd, 4th and 5th rows of Table 3 which show that introducing the additional loss terms to constrain the refined meshes improves the accuracy of the generated shapes.

Ground truth depth as input In row 7 we use ground truth instead of predicted depths which gives the upper bound on our mesh prediction accuracy in relation to the depth prediction accuracy.

Sphere initialization Row 8 uses a sphere as the coarse shape instead of cubified voxel grid.

Naive multi-view Mesh R-CNN In row 9 of Table 3 we extend Mesh R-CNN Gkioxari et al. (2019) to multi-view using statistical feature pooling method proposed in Wen et al. (2019) for mesh refinement while in row 10 we further extend their single-view voxel grid prediction method to our probabilistic multi-view voxel grid prediction.

	F1-τ	F1-2τ
(1) Baseline framework	79.82	90.18
(2) **Baseline + rendered vs predicted depth loss (final model)**	**80.80**	**90.72**
(3) Baseline + rendered vs GT depth loss	80.35	90.55
(4) Baseline + rendered vs predicted depth loss + rendered vs GT depth loss	80.45	90.56
(5) Baseline with stats pooling	79.63	90.10
(6) Baseline with simple attention	80.03	90.21
(7) Baseline with GT depth	**84.58**	**92.86**
(8) Sphere initialization	73.78	85.49
(9) Naive multi-view Mesh R-CNN (single-view voxel prediction)	72.74	84.99
(10) Naive multi-view Mesh R-CNN (multi-view voxel prediction)	76.97	88.24

Table 3: **Comparison of shape generation accuracy with different settings** of additional contrastive depth losses, multi-view feature pooling. The Baseline framework uses multi-head attention mechanism without any contrastive depth losses.

Number of View We test the performance of our framework with respect to the number of views. Table 4 shows that the accuracy of our method increases as we increase the number of input views for training. These experiments also validate that the attention-based feature pooling can efficiently encode features from different views to take advantage of larger number of views.

Table 5 shows the results when using different number of views during testing on our model trained with 3 views which indicates that increasing the number of views during testing does not improve the accuracy while decreasing the number of views can cause a significant drop in accuracy.

Metric	2	3	4	5	6
F1-τ	73.60	80.80	82.61	83.76	84.25
F1-2τ	85.80	90.72	91.78	92.73	93.14

Table 4: **Accuracy w.r.t the number of views during training**. The evaluation was performed on the same number of views as training.

Metric	2	3	4	5	6
F1-τ	72.46	80.80	80.98	80.94	80.85
F1-2τ	84.49	90.72	91.03	91.16	91.20

Table 5: **Accuracy w.r.t the number of views during testing**. The same model trained with 3 views was used in all of the cases.

5 Conclusion

We propose a neural network based solution to predict 3D triangle mesh models of objects from images taken from multiple views. First, we propose a multi-view voxel grid prediction module which probabilistically merges voxel grids predicted from individual input views. We then cubify the merged voxel grid to triangle mesh and apply graph convolutional networks for further refining the mesh. The features for the mesh vertices are extracted from contrastive depth input consisting of rendered depths at each refinement stage along with the predicted depths. The proposed mesh reconstruction method outperforms existing methods with a large margin and is capable of reconstructing objects with more complex topologies.

References

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1538–1547, 2019.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pp. 628–644. Springer, 2016.

Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1): 22–43, 2008.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, 2005.

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9785–9795, 2019.

Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. *arXiv preprint arXiv:1912.06378*, 2019.

JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *2017 International Conference on 3D Vision (3DV)*, pp. 263–272. IEEE, 2017.

Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. Seqxy2seqz: Structure learning for 3d shapes by sequentially predicting 1d occupancy segments from 2d coordinates. *arXiv preprint arXiv:2003.05559*, 2020.

Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1586–1594, 2017.

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. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2821–2830, 2018.

Qixing Huang, Hai Wang, and Vladlen Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

Xin Jia, Shourui Yang, Yuxin Peng, Junchao Zhang, and Shengyong Chen. Dv-net: Dual-view network for 3d reconstruction by fusing multiple sets of gated control point clouds. *Pattern Recognition Letters*, 131:376–382, 2020.

Jiongchao Jin, Akshay Gadi Patil, Zhang Xiong, and Hao Zhang. Dr-kfs: A differentiable visual similarity metric for 3d shape reconstruction, 2019.

Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1966–1974, 2015.

Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pp. 365–376, 2017.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Kurt Konolige. Improved occupancy grids for map building. *Autonomous Robots*, 4(4):351–367, 1997.

Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 858–866. IEEE, 2018.

Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

Sophie Lambert-Lacroix and Laurent Zwald. The adaptive berhu penalty in robust regression. *Journal of Nonparametric Statistics*, 28(3):487–514, 2016.

Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005.

Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 969–978, 2019.

Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 10452–10461, 2019.

Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9964–9973, 2019.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Hao Su, Qixing Huang, Niloy J Mitra, Yangyan Li, and Leonidas Guibas. Estimating image depth using shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4541–4550, 2019.

Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 52–67, 2018.

Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1042–1051, 2019.

Udaranga Wickramasinghe, Edoardo Remelli, Graham Knott, and Pascal Fua. Voxel2mesh: 3d mesh model generation from volumetric data, 2019.

Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in neural information processing systems*, pp. 1696–1704, 2016.

Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction. *International Journal of Computer Vision*, 128(1): 53–73, 2020.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–215, 2018.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 767–783, 2018.

Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5525–5534, 2019.

Yuan Yao, Nico Schertler, Enrique Rosales, Helge Rhodin, Leonid Sigal, and Alla Sheffer. Front2back: Single view 3d shape reconstruction via front to back prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 531–540, 2020.

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

A APPENDIX

NETWORK ARCHITECTURE

MVSNET ARCHITECTURE

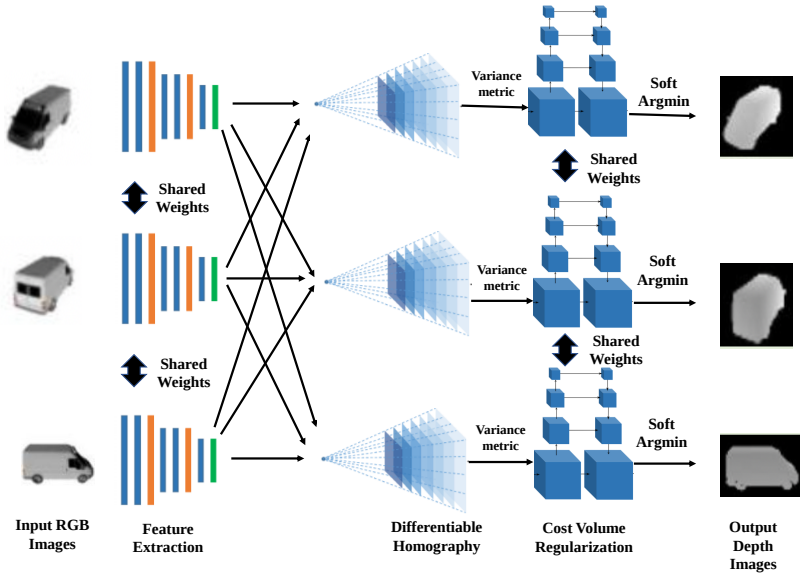

Figure 4: Depth prediction network (MVSNet) architecture

Our depth prediction module is based on MVSNet Yao et al. (2018) which constructs a regularized 3D cost volumes to estimate the depth map of the reference view. Here, we extent MVSNet to predict

the depth maps of all views instead of only the reference view. This is achieved by transforming the feature volumes to each view's coordinate frame using homography warping and applying identical cost volume regularization and depth regression on each view. This allows the reuse of pre-regularization feature volumes for efficient multi-view depth prediction invariant to the order of input images. Figure 4 shows the architecture of the our depth estimation module.

PROBABILISTIC OCCUPANCY GRID MERGING

We use single-view voxel prediction network from Gkioxari et al. (2019) to predict predicts voxel grids for each of the input images in their respective local coordinate frames. The occupancy grids are transformed to global frame (which is set to the coordinate frame of the first image) by finding the equivalent global grid values in the local grids after applying bilinear interpolation on the closest matches. The voxel grids in global coordinates are then probabilistically merged according to Sub-section 3.1 of the main submission.

EXPERIMENTS

We quantitatively compare our method against previous works for multi-view shape generation in Table 6 and show effectiveness of our proposed shape generation methods in improving shape quality. Our method outperforms the state-of-the-art method Pixel2Mesh++ Wen et al. (2019) with decrease in chamfer distance to ground truth by 34%, which shows the effectiveness of our proposed method. Note that in Table 6 same model is trained for all the categories but accuracy on individual categories as well as average over all the categories are evaluated.

Category	Chamfer Distance (CD) ↓				
	3D-R2N2	LSM	MVP2M	P2M++	**Ours**
Couch	0.806	0.730	0.534	0.439	**0.220**
Cabinet	0.613	0.634	0.488	0.337	**0.230**
Bench	1.362	0.572	0.591	0.549	**0.159**
Chair	1.534	0.495	0.583	0.461	**0.201**
Monitor	1.465	0.592	0.658	0.566	**0.217**
Firearm	0.432	0.385	0.305	0.305	**0.123**
Speaker	1.443	0.767	0.745	0.635	**0.402**
Lamp	6.780	1.768	0.980	1.135	**0.755**
Cellphone	1.161	0.362	0.445	0.325	**0.138**
Plane	0.854	0.496	0.403	0.422	**0.084**
Table	1.243	0.994	0.511	0.388	**0.181**
Car	0.358	0.326	0.321	0.249	**0.165**
Watercraft	0.869	0.509	0.463	0.508	**0.175**
Mean	1.455	0.664	0.541	0.486	**0.211**

Table 6: **Qualitative comparison** against state-of-the-art multi-view shape generation methods. Following Wen et al. (2019), we report Chamfer Distance in $m^2 \times 1000$ from ground truth for different methods. Note that same model is trained for all the categories but accuracy on individual categories as well as average over all the categories are evaluated.

ABLATION STUDIES

Coarse Shape Generation We conduct comparisons on voxel grid predicted from our proposed probabilistically merged voxel grids against single view method Gkioxari et al. (2019). As is shown in Table 7, the accuracy of the initial shape generated from probabilistically merged voxel grid is higher than that from individual views.

Accuracy at different GCN stages We analyze the accuracy of meshes at different GCN stages in Table 8 to validate that our method produces the meshes in a coarse-to-fine manner.

Resolution of Depth Prediction We conduct experiments using different numbers of depth hypotheses in our depth prediction network (Sub-section A), producing depth values at different

Metric	Single-view	Multi-view
F1-τ	25.19	31.27
F1-2τ	36.75	44.46

Table 7: **Accuracy of predicted voxel grids** from single-view prediction compared against the proposed probabilistically merged multi-view voxel grids. The voxel branch was trained separately without the mesh refinement and evaluation was performed on the cubified voxel grids. We use three views for probabilistic grid merging.

Metric	Cubified	Stage-1	Stage-2	Stage-3
F1-τ	31.48	76.78	79.88	80.80
F1-2τ	44.40	88.32	90.19	90.72

Table 8: **Accuracy of the refined meshes at different GCN stages**. 1, 2 and 3 indicate the performance at the corresponding graph convolution blocks while *Cubified* is for the cubified voxel grids used as input for the first GCN block. All the stages, including the voxel prediction, were trained jointly and hence the accuracy of voxel predictions varies from that in Table 7.

resolutions. A higher number of depth hypothesis means finer resolution of the predicted depths. The quantitative results with different hypothesis numbers are summarized in Table 9. We set depth hypothesis as 48 for our final architecture which is equivalent to the resolution of 25 mm.

Metric	24	48	72	96
F1-τ	80.29	80.80	80.69	80.34
F1-2τ	90.43	90.72	90.74	90.47

Table 9: **Accuracy w.r.t the number of depth hypothesis**. A higher number of depth hypothesis increases the resolution of predicted depth values at the expense of higher memory requirement. The range of depths for all the models are same and based on the minimum/maximum depth in the ShapeNet Chang et al. (2015) dataset.

Generalization Capability We conduct experiments to evaluate the generalization capability of our system across the semantic categories. We train our model with only 12 out of the 13 categories and test on the category that was left out. Table 10 shows that the accuracy generally does not decrease significantly when compared with the model that was trained on all 13 categories.

Category	F-score (τ) ↑		F-score (2τ) ↑	
	Excluding	Including	Excluding	Including
Couch	63.29	73.63	80.79	88.24
Cabinet	68.26	76.39	83.10	88.84
Bench	76.08	83.76	87.42	92.57
Chair	60.60	78.69	75.93	90.02
Monitor	67.26	76.64	81.57	88.89
Firearm	78.59	94.32	86.28	97.67
Speaker	62.39	67.83	77.77	82.34
Lamp	63.50	75.93	74.66	85.33
Cellphone	67.24	86.45	80.54	94.28
Plane	57.48	92.13	67.27	96.57
Table	76.41	83.68	86.86	91.97
Car	59.08	80.43	75.58	92.33
Watercraft	64.97	80.48	78.95	90.35

Table 10: **Accuracy when a category is excluded** during training and evaluation is performed on the category to verify how well training on other categories generalizes to the excluded category.