8. Supplementary Material

8.1. Voxelization

Initial voxelization was done in world coordinates and used exact voxelization, i.e. any voxel partially intersected by a face was defined as filled. We subsequently filled in hollow models by switching the state of any empty voxel without any free path to the exterior bounding box along any of the 6 rays pointing in the $\pm\hat{i}$, $\pm\hat{j}$ and $\pm\hat{k}$ directions starting at the voxel.

Frustum voxels were calculated based on these filled in world-coordinate voxel grids of the same resolution using nearest neighbour sampling. Near and far planes were based on viewing a sphere of radius 0.5 centred at the origin. Note all reported intersection-over-union values on frustum grids have been reweighted to account for the non-uniform volume of the elements.

8.2. Training Details

All kernels of all networks except the inner-loop CNNs had L2 regularization with weight 5×10^{-4}. No regularization was applied to the inner-loops CNNs.

Networks were trained at 32^3, 64^3, 128^3 and 256^3. All except the highest resolution were trained with a batch size of 16 for 100,000 steps. The image encoder for the 32^3 model was initialized with publicly available weights trained on ImageNet [12]. Other convolutional kernels were initialized with Glorot uniform initialization [16] except the final kernel of each loss network, which was initially a factor of 10^{-3} lower as before. Higher resolution networks were initialized from their lower resolution counterparts.

To allow models to be trained on a desktop GPU (Nvidia GTX 1080-Ti), the highest resolution networks (256^3) used a batch size of 4. We turned off batch normalization for this final model to avoid spurious batch statistics due to the reduced batch size. We observed very little improvement beyond the first few thousand steps, so terminated training after $20,000$ steps.

We used Adam [25] for our outer optimizer. Our 32^3 models used a learning rate of 5×10^{-5}, while higher resolution networks used 2×10^{-5}.

	IGE-MN				IGE-I4			
	32^3	64^3	128^3	256^3	32^3	64^3	128^3	256^3
plane	29.6	44.8	52.9	54.4	30.5	47.8	57.5	57.3
bench	25.5	32.8	35.4	34.6	26.1	37.1	41.2	38.4
cabinet	58.0	66.7	69.0	69.4	59.5	68.6	71.9	70.2
car	57.8	68.8	72.8	73.3	57.9	70.9	74.0	75.2
telephone	54.9	66.3	69.7	68.5	55.0	66.6	73.3	70.0
chair	37.4	46.3	49.2	48.1	38.7	50.1	53.7	51.3
sofa	55.5	63.9	66.5	65.8	56.0	66.7	70.0	68.2
rifle	32.3	44.2	51.0	50.1	32.1	47.4	54.6	51.9
lamp	27.6	35.9	39.5	37.8	29.2	38.7	42.4	39.6
monitor	41.1	48.7	51.2	51.0	41.7	52.2	54.9	51.2
speaker	61.6	69.1	71.5	71.2	63.6	71.4	74.4	73.3
table	33.6	44.0	47.8	48.2	34.8	46.5	52.7	50.5
watercraft	41.5	52.1	56.0	55.4	41.6	54.6	59.2	57.1
$\text{mean}_{\text{cat.}}$	42.8	52.6	56.3	56.0	43.6	55.3	60.0	58.0

Table 6: Mean IoU (in %) evaluated at 256^3 resolution. A single model is trained across all categories. Lower resolution inferences are trilinearly upsampled.

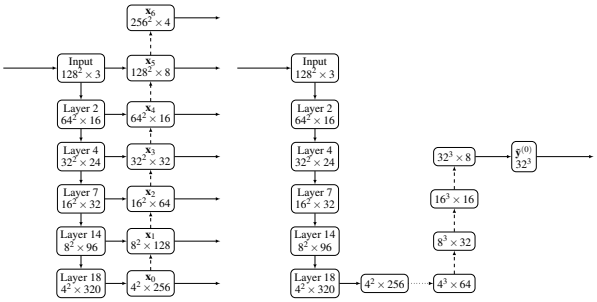

Figure 7: **Left**: Image Feature Network for IGE-MN model. The left column is the standard MobilenetV2 convolutional network [42] and is shared with the initial estimate network. Dashed arrows represent bilinear resizing. $\mathbf{x}_i$ values are the result of a 1×1 convolution on the concatenated inputs followed by batch normalization and rectified linear activation. **Right**: Initial Estimate Network for MN model. The left column is shared with the image feature network. Dashed arrows represent 4^3 deconvolutions with isotropic stride 2 followed by batch normalization and a rectified linear activation. The dotted line is a reshape.

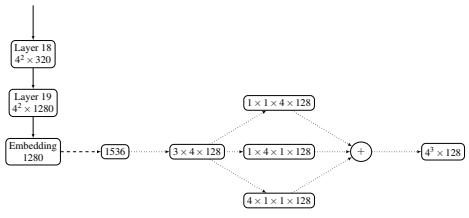

Figure 8: Initial decoding transformation for baseline voxel MN decoder. Operations left-to-right from embedding layer: dense layer, reshape, split/reshape, addition with dimension broadcasting. Note there are no learned parameters due to any dotted arrows