Inverse Graphics GAN: Supplemental Material

A Related Work

Geometry Based Approaches (or 3D Reconstruction): Reconstructing the underlying 3D scene from only 2D images has been one of the long-standing goals of computer vision. Classical work in this area has focused on geometry based approaches in the single instanced setting where the goal was only to reconstruct a single 3D object or scene depicted in one or more 2D images [3, 9, 4, 14, 25, 42, 47, 49, 48]. This early work was not learning based, however, and so was unable to reconstruct any surfaces which do not appear in the image(s).

Learning to Generate from 3D Supervision: Learning-based 3D reconstruction techniques use a training set of samples to learn a distribution over object shapes. Much past work has focused on the simplest learning setting in which we have access to full 3D supervision. This includes work on generating voxels [5, 8, 45, 55, 56], generating point-clouds [1, 12, 23, 58, 1, 26], generating meshes [17, 40, 53] and generating implicit representations [2, 7, 16, 21, 33, 34, 41, 46, 57]. Creating 3D training data is much more expensive, however, because it requires either skilled artists or a specialized capture setup. So in contrast to all of this work we focus on learning only from unstructured 2D image data which is more readily available and cheaper to obtain.

Learning to Generate from 2D Supervision: Past work on learning to generate 3D shapes by training on only 2D images has mostly focused on differentiable renderers. We can categorize this work based on the representation used. Mesh techniques [24, 6, 15, 18] are based on deforming a single template mesh or a small number of pieces [18], while Loper and Black [30] and Palazzi et al. [39] use only a low-dimensional pose representation, so neither is amenable to generating arbitrary topologies. Concurrent work on implicit models [38, 29] can directly learn an implicit model from 2D images without ever expanding to another representation, but these methods rely on having camera intrinsics for each image, which is usually unavailable with 2D image data. Our work instead focuses on working with unannotated 2D image data.

The closet work to ours uses voxel representations [13, 19]. Voxels can represent arbitrary topologies and can easily be converted to a mesh using the marching cubes algorithm. Furthermore, although it is not a focus of this paper, past work has shown that the voxel representation can be scaled to relatively high resolutions through the use of sparse representations [45].Gadelha et al. [13] employs a visual hull based differential renderer that only considers a smoothed version of the object silhouette, while Henzler et al. [19] relies on a very simple emission-absorption based lighting model. As we show in the results, both of these models struggle to take advantage of lighting and shading information which reveals surface differences, and so they struggle to correctly represent concavities like bathtubs and sofas.

In contrast to all previous work, our goal is to be able to take advantage of fully-featured industrial renderers which included many advanced shading, lighting and texturing features. However these renderers are typically not built to be differentiable, which is challenging to work with in machine learning pipelines. The only work we are aware of which uses an off-the-shelf render for 3D generation with 2D supervision is Rezende et al. [44]. In order to differentiate through the rendering step they use the REINFORCE gradients [54]. However, REINFORCE scales very poorly with number of input dimensions,allowing them to show results on simple meshes only. In contrast, our method scales *much* better since dense gradient information can flow through the proxy neural renderer.

Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS 2020.

Neural Rendering With the success of 2D generative models, it has recently become popular to skip the generation of an explicit 3D representation *Neural Rendering* techniques focus only on simulating 3D by using a neural network to generate 2D images directly from a latent space with control over the camera angle [11, 36, 50] and properties of objects in the scene [28]. In contrast, our goal is to generate the 3D shape itself, not merely controllable 2D renders of it. This is important in circumstances like gaming where the underingly rendering framework is may be fixed, or where we need direct access to the underlying 3D shape itself, such as in CAD/CAM applications. We do however build directly on RenderNet Nguyen-Phuoc et al. [37] which is a neural network that can be trained to generate 2D images from 3D shapes by matching the output of an off-the-shelf renderer.

Differentiating Through Discrete Decisions Recent work has looked at the problem of differentiating through discrete decisions. Maddison et al. [31] and Jang et al. [22] consider smoothing over the discrete decision and Tucker et al. [52] extends this with sampling to debias the gradient estimates. In section the main paper we have discuss why these methods cannot be applied in our setting. Liao et al. [27] discusses why we cannot simply differentiate through the Marching Cubes algorithm, and also suggests using continuous voxel values to generate a probability distribution over 3D shapes. However in their setting they have ground truth 3D data so they directly use these probabilities to compute a loss and do not have to differentiate through the voxel sampling process as we do when training from only 2D data.

B A note on the challenges of applying REINFORCE in our setting

The main challange of the paper lies in the fact that the generation process

$$\boldsymbol{x}_c \sim p_G(\boldsymbol{x}_c) \Leftrightarrow \boldsymbol{z} \sim p(\boldsymbol{z}), \boldsymbol{x}_c = G_\theta(\boldsymbol{z}),$$
$$\boldsymbol{y} \sim p_G(\boldsymbol{y}) \Leftrightarrow \boldsymbol{x}_c \sim p_G(\boldsymbol{x}_c), \boldsymbol{x}_d \sim p(\boldsymbol{x}_d|\boldsymbol{x}_c), \boldsymbol{y} = R_d(\boldsymbol{x}_d).$$

involves sampling a discrete variable $\boldsymbol{x}_d$ thus making the generator's loss non-differentiable w.r.t. θ. An initial solution would be to use the REINFORCE gradient estimator [54]:

$$\nabla_\theta \mathcal{L}_{\text{gen}}(\theta) = \mathbb{E}_{p_G(\boldsymbol{x}_c)}\left[\mathbb{E}_{p(\boldsymbol{x}_d|\boldsymbol{x}_c)}\left[\log D_\phi(R_d(\boldsymbol{x}_d))\right] S_G(\boldsymbol{x}_c)\right], \quad S_G(\boldsymbol{x}_c) = \nabla_\theta \log p_G(\boldsymbol{x}_c).$$

The expectation term $\mathbb{E}_{p(\boldsymbol{x}_d|\boldsymbol{x}_c)}\left[\log D_\phi(R_d(\boldsymbol{x}_d))\right]$ is called the "reward" of generating $\boldsymbol{x}_c \sim p_G(\boldsymbol{x}_c)$. Intuitively, the gradient ascent update using the above formula would encourage the generator to generate $\boldsymbol{x}_c$ with high reward, thus fooling the discriminator. However, REINFORCE is not directly applicable as the score function $S_G(\boldsymbol{x}_c)$ is intractable (the distribution $p_G(\boldsymbol{x}_c)$ is implicitly defined). A second attempt would replace the discrete sampling step $\boldsymbol{x}_d \sim p(\boldsymbol{x}_d|\boldsymbol{x}_c)$ with continuous value approximations [22, 31], which shall enable differentiation. However, the off-the-shelf renderer is treated as black-box so we do not assume the back-propagation mechanism is implemented there. Even worse, the discrete variable sampling step is necessary as the off-the-shelf renderer can only work with discrete voxel maps. Therefore the instance-level gradient is not defined on the discrete voxel grid, and gradient approximation methods based on continuous relaxations cannot be applied neither.

C Additional Computational Results

C.1 Implementation Details

Our rendering engine is based on the Pyrender [32] which is built on top of OpenGL. We employ a 3D convolutional GAN architecture for the generator [55] with a 64^3 voxel resolution. To incorporate the viewpoint, the rigid body transformation embeds the 64^3 grid into a 128^3 resolution. We render the volumes with a RenderNet [37] architecture, with the modification of using 2 residual blocks in 3D and 4 residual blocks in 2D only. The discriminator architecture follows the design in DCGAN [43], taking images of 128^2 resolution. Additionally, we add spectral normalization to the discriminator [35] to stablize training.

We employ a 1:1:1 updating scheme for generator, discriminator and the neural renderer, using learning rates of 2e-5 for the neural renderer and 2e-4 for both generator and discriminator. The Discriminator Output Matching loss is weighted by $\lambda = 100$ over the $\mathcal{L}_2$ loss. We found that training was stable against changes in λ and extensive tuning was not necessary. The binerization distribution $p(\boldsymbol{x}_d|\boldsymbol{x}_c)$ was chosen as a global thresholding, with the threshold being distributed uniformly in $[0, 1]$.

C.2 Experimental Details

To generate the training data, we illuminate ShapeNet objects fo the three categories Chair, Couch and Bathtub from two beam light sources. We uniformly sample from all 360° of rotation, with an elevation between 15 and 60 degrees for bathtubs and couches and -30 and 60 degrees for chairs. The limitation on elevation angle is chosen to generate a data set that is as realistic as possible, given the object category.

We chose to evaluate the quality of the generated 3D models by rendering them to 2D images and computing Fréchet Inception Distances (FIDs) [20]. This focuses the evaluation on the observed visual quality, preventing if from considering what the model generates in the unobserved insides of the objects. All FID scores reported in the main paper use an Inception network [51] retrained to classify Images generated with our renderer, classifying rendered images of the 21 largest object classes in ShapeNet with 95.3% accuracy. In This supplemental material we also show FID scores using the Inception network trained on ImageNet [10].

C.3 Extended Figures and FID scores

We include a Figure of samples generated by our proposed approach in the unlimited setting, demonstrating that our approach is capable of generating samples of high visual quality 1. We additionally report FID scores on a small data set that contains only 500 samples, for each of the three categories Chairs, Couches and Bathtubs. We also consider a limited viewpoint (LVP) setting for the chairs dataset where the azimuth rotation is limited to 60° in each direction, and elevation is limited to be between 15 ad 60 degrees, to simulate the viewpoint bias observed in natural photographs. Results for this set can be seen in Figure 2. We note that the difference between our proposed method and the baseline in detecting flat surfaces and concaveties is particularly clear in this setting for the chair data. The FID scores alongside the scores reported in the main paper can be found in Table 1.

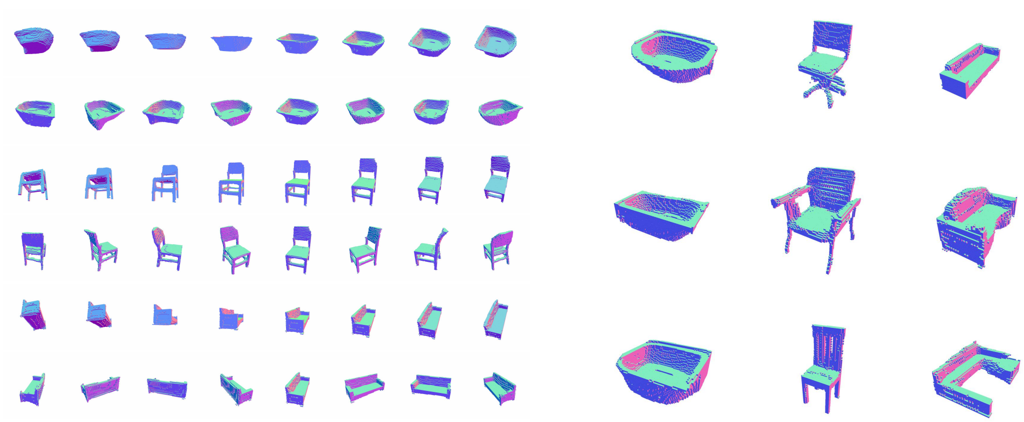
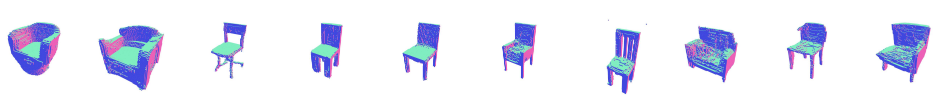


Figure 1: Normal Maps of objects generated by IG-GAN on the 'Unlimited' datasets. The left panel shows a single sample rendered in different view points, and the right panel shows multiple samples rendered from a canonical viewpoint.



(a) IG-GAN (Ours) (b) Absorption Only

Figure 2: Results on the chairs LVP dataset. Unlike our method, the baseline can not extract sufficient information from the data to create chair samples with flat surfaces.

Table 1: FID scores computed on ShapeNet objects (bathtubs, couches and chairs), using Inception weights retrained on ShapeNet.

# of Images	500			**One Per Model** ($\approx$ 3000)			**Unlimited**			
Dataset	**Tubs**	**Couches**	**Chairs**	**Tubs**	**Couches**	**Chairs**	**Tubs**	**Couches**	**Chairs**	**LVP**
2D-DCGAN	737.7	540.5	672.8	461.8	354.3	362.3	226.7	210.9	133.2	237.9[1]
Visual Hull	305.8	279.3	183.4	184.6	106.2	37.1	90.1	35.1	15.7	34.5
Absorbtion Only	336.9	282.9	218.2	275.8	78.0	32.8	104.5	25.5	23.8	38.6
IG-GAN (Ours)	**187.8**	**114.1**	**119.9**	**67.5**	**35.8**	**20.7**	**44.0**	**17.8**	**13.6**	**20.6**

C.4 A real world data set: Chanterelles

We demonstrate that the proposed method is able to produce realistic samples when trained on a dataset of natural images. Figure 3 shows samples from a model trained on the Chanterelle mushrooms dataset from Henzler et al. [19]. We prepare the *Chanterelle* data by cropping and resizing the images to 128^2 resolution and by unifying the mean intensity in an additive way. Note that the background of the natural images has been masked out in the original open-sourced dataset.

Figure 3: Chanterelle mushroom dataset samples and generated shapes from our model trained on this dataset.

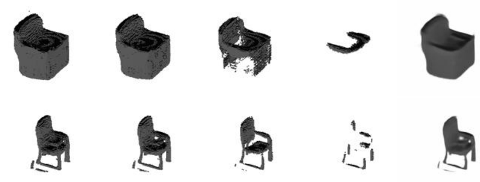
C.5 Ablation Study

Discriminator output matching We study the effect of the proposed discriminator output matching (DOM) loss in various scenarios. In Table 2, we report the FID scores on the models trained without the DOM loss, from this comparison we see that the DOM loss plays a crucial role in learning to generate high-quality 3D objects. In Figure 4 we can see that that the non-binary volumes sampled from the generator can be rendered to a variety of different images by OpenGL, depending on the random choice of threshold. Without the DOM loss, the trained neural renderer simply averages over these potential outcomes, considerably smoothing the result in the process and losing information about fine structures in the volume. This leads to weak gradients being passed to the generator, considerably deteriorating sample quality.

Table 2: Ablation results without discriminator output matching (DOM) when training on chairs/couches "one per model" datasets. We either fix the pre-trained neural renderer ("Fixed"), or continuing to train it during GAN training ("Retrained"). The generator samples fed to the discriminator are rendered using either OpenGL or the neural renderer. For reference, our model is equivalent to the Retrained OpenGL setup with the addition of the DOM loss and achieves FID scores 20.7/35.8.

	OpenGL	RenderNet
Retrained	86.4/180.1	74.7/144.8
Fixed	113.7/323.5	103.9/124.6

Another setting from Table 2 shows the discriminator trained using generated samples rendered by the neural renderer instead of OpenGL. This inherently prevents the mode collapse observed in the above setting. However, it leads to the generator being forced by the discriminator to produce binary voxel maps early on in training. This seems to lead to the generator getting stuck in a local optima, hence deteriorating sample quality.



(a) 0.1 (b) 0.2 (c) 0.5 (d) 0.8 (e) neural renderer

Figure 4: Samples from a generator trained without DOM, rendered using the neural renderer on the continuous sample and OpenGL on various thresholds.

Table 3: Comparisons of neural renderer pre-trainings on different 3D shapes. FIDs are reported for the 'One per model' chairs.

	Chairs	Tables	Random
Ours	22.6	20.7	20.4
Fixed	37.8	105.8	141.9

Pre-training We investigate the effect of various pre-trainings of the neural renderer. All other experiments were conducted with the neural renderer pre-trained on the *Tables* data from ShapeNet (see Table 1). As a comparison, we run the proposed algorithm on the chair data using a neural renderer pre-trained on either the Chair data itself or a simple data set consisting of randomly sampled cubes. As shown in Table 3, the quality of the results produced by our method is robust to changes in the pre-training of the neural renderer. In contrast, if we use a fixed pre-trained renderer it produces reasonable results if pre-trained directly on the domain of interest, but deteriorates significantly if trained only on a related domain. Note that we assume no access to 3D data in the domain of interest so in practice we cannot pre-train in this way.

C.6 FID Scores Calculated with Inception Network Trained on ImageNet Classification

In the main paper as well as in the paragraphs above, all FID scores were calculated using an Inception network which was trained to classify gray-scale ShapeNet renders that look similar to the training data used for all of our models. Below we report, for the same set of experiments, FID scores calculated using the traditional ImageNet trained Inception network in tables 5, 4 and 6.

D A note on Emission-Absorption

We chose to compare to the Absorption-only (AO) model from Henzler et al. [19] and not the Emission-Absorption (EA) model. The EA model was designed to incorporate color information into the differentiable rendering engine. In addition to the occupancy/absorbtion value generated at each voxel, this model also generates one or more emission values at each voxel that can represent either 3-channel color, or a single grey-scale value. The focus of our paper was only on shape,

[1] A fair comparison to 2D-DCGAN is impossible, as the generator is trained on LVP (Limited View Point) data, but to facilitate easy comparison all FID evaluations are computed with same test data (which includes views from all 360°).

Table 4: FID scores computed on ShapeNet objects (bathtubs, couches and chairs), using original Inception weights without retraining.

# of Images	500			**One Per Model** (≈ 3000)			**Unlimited**			
Dataset	**Tubs**	**Couches**	**Chairs**	**Tubs**	**Couches**	**Chairs**	**Tubs**	**Couches**	**Chairs**	**LVP**
2D-DCGAN	356.9	324.6	291.9	211.0	156.5	196.8	210.1	117.8	78.8	101.5[1]
Visual Hull	117.3	130.0	153.2	66.5	78.7	47.3	29.7	41.4	**22.0**	36.5
Absorbtion Only	117.6	115.9	**110.7**	74.3	70.3	46.6	37.6	36.6	31.4	41.8
IG-GAN (Ours)	**95.1**	**91.4**	131.4	**41.1**	**36.5**	**32.1**	**23.4**	**18.6**	22.6	**29.2**

Table 5: Ablation results without discriminator output matching (DOM) when training on chairs/couches "one per model" datasets. We either fix the pre-trained neural renderer ("Fixed"), or continuing to train it during GAN training ("Retrained"). The generator samples fed to the discriminator are rendered using either OpenGL or the neural renderer. For reference, our model is equivalent to the Retrained OpenGL setup with the addition of the DOM loss and achieves FID scores 32.1/36.5. FID scores calculated using an Inception network trained on ImageNet.

	OpenGL	RenderNet
Retrained	93.6/146.6	58.1/88.2
Fixed	149.3/238.6	61.6/100.0

Table 6: Comparisons of neural renderer pre-trainings on different 3D shapes. FIDs are reported for the 'One per model' chairs. FID scores calculated using an Inception network trained on ImageNet.

	Chairs	Random	Tables
Ours	32.7	31.4	32.1
Fixed	43.0	84.4	69.5

however, leaving color generation for future work. Thus the underlying ShapeNet voxel data used in our experiments does not have any color channel information, and consists of only a single 0-1 occupancy value for each voxel. Therefore, including the additional emission value would only result in providing the EA model additional freedom that it should not use when modeling the data. In the following we show that if we assume that the model generates only a single occupancy channel and the emitted color is fixed globally, then the EA model naturally reduces to the AO model.

In Henzler et al. [19] the expression

$$\rho_{EA}(\mathbf{v}) = \frac{\sum_{i=1}^{n_z} v_{a,i} v_{e,i} \prod_{j=1}^{i}(1-v_{a,j})}{\sum_{i=1}^{n_z} v_{a,i} \prod_{j=1}^{i}(1-v_{a,j}) + \epsilon} \left[1 - \prod_{j=1}^{n_z}(1-v_{a,j})\right]$$

is used for the Emission-Absorption model, where $v_{e,j}$ denotes the emission coefficient, $v_{e,j}$ the absorption, n_z the number of voxels along the chosen dimension and the index j refers to the j-th occupancy of the 3D model along a straight line through the volume. The regularization parameter ϵ is chosen small to numerically stabilize the quotient. In the case of data generated from shape information alone, we can consider that all objects are perfectly white, which would equate to $v_{e,j} = 1$. In this case, the quotient

$$\frac{\sum_{i=1}^{n_z} v_{a,i} v_{e,i} \prod_{j=1}^{i}(1-v_{a,j})}{\sum_{i=1}^{n_z} v_{a,i} \prod_{j=1}^{i}(1-v_{a,j})}$$

naturally reduces to one, leaving the expression

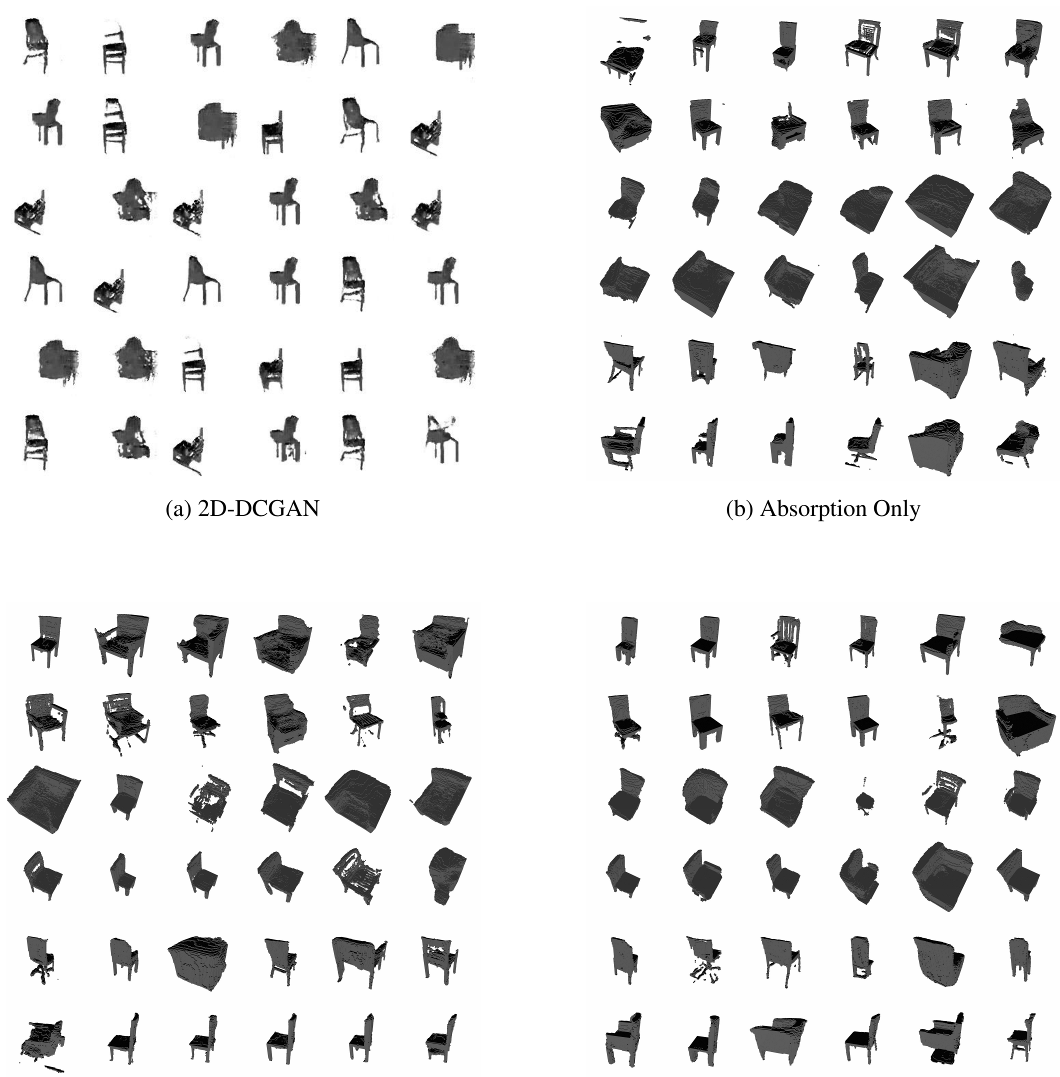
$$\rho_{EA}(\mathbf{v}) = 1 - \prod_{j=1}^{n_z}(1-v_{a,j}),$$

for the EA model, which is identical to the AO model. Hence, the two imaging models are the same in our setting of where images are obtained from 3D shapes with a single globally emitted color.

E Random Samples from Each Model Trained on Each of the Datasets [1]

The rest of the supplemental contains a set of tables where each table contains random samples from one of the models trained on one of the datasets. For each set of random samples we show black and white renders as well as normal map renders. In each figure the view angle is held fixed across all samples in a single row, but each image represents a completely independently sampled underlying 3D model. Note that we cannot show normal map renderers for 2D-DCGAN generations since the 2D-DCGAN only generates images, not 3D models. At the end we also show samples from the dataset rendered in the same way for comparison.

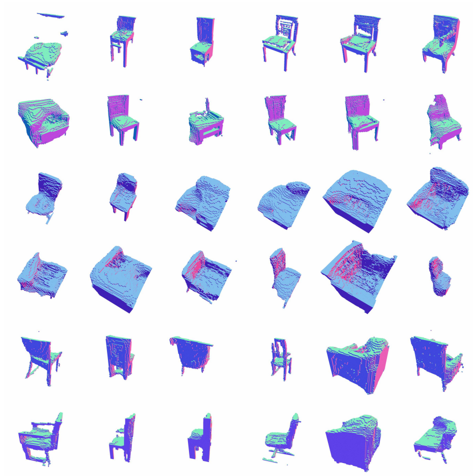
[1] This document does not contain the full supplemental material due to size restrictions. Full material available at `https://figshare.com/s/56084c4d7df8f57f15d9`



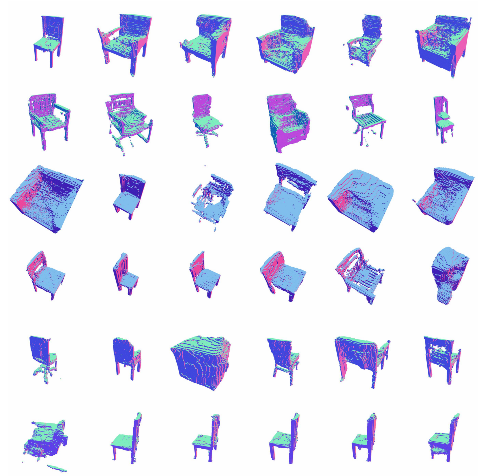
(a) 2D-DCGAN (b) Absorption Only

(c) Visual Hull (d) IG-GAN (Ours)

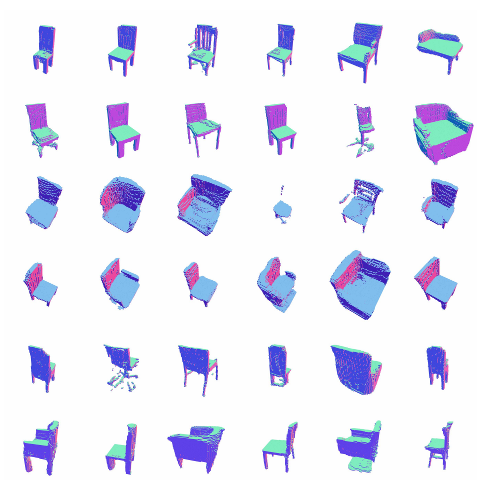
Figure 5: Samples from models trained on the Chairs data in the ’one sample per object’ setting (6667 training images).



(a) Absorption Only



(b) Visual Hull



(c) IG-GAN (Ours)

Figure 6: Samples from models trained on the Chairs data in the 'one sample per object' setting (6667 training images), r,endered as normal maps.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *International Conference on Machine Learning*, 2018.

[2] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In *Advances in Neural Information Processing Systems*, pages 2032–2041, 2019.

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, January 2011.

[4] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 388–393. IEEE, 2001.

[5] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.

[6] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2019.

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[8] 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*, pages 628–644. Springer, 2016.

[9] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[11] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[12] 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*, pages 605–613, 2017.

[13] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411. IEEE, 2017.

[14] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[15] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018.

[16] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7154–7164, 2019.

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[18] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, pages 1–20, 2019.

[19] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[21] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 336–354, 2018.

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[23] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 802–816, 2018.

[24] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

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

[26] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

[27] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018.

[28] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. *arXiv preprint arXiv:1912.05237*, 2019.

[29] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 8293–8304, 2019.

[30] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[31] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[32] Matthew Matl. Pyrender. URL `https://github.com/mmatl/pyrender`.

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[34] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4743–4752, 2019.

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[36] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. *arXiv preprint arXiv:1904.01326*, 2019.

[37] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *Advances in Neural Information Processing Systems 31*, pages 7891–7901, 2018.

[38] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. *arXiv preprint arXiv:1912.07372*, 2019.

[39] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. End-to-end 6-dof object pose estimation through differentiable rasterization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[40] 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*, pages 9964–9973, 2019.

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[42] Andrew Prock and Chuck Dyer. Towards real-time voxel coloring. In *Proceedings of the DARPA Image Understanding Workshop*, volume 1, page 2. Citeseer, 1998.

[43] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*, 2016.

[44] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in neural information processing systems*, pages 4996–5004, 2016.

[45] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[46] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019.

[47] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

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

[49] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[50] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1119–1130, 2019.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[52] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

[53] 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)*, pages 52–67, 2018.

[54] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[55] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

[56] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2690–2698, 2019.

[57] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 490–500, 2019.

[58] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019.