Adversarial Generation of Continuous Implicit Shape Representations

Marian Kleineberg, Matthias Fey and Frank Weichert

TU Dortmund University, Germany

Figure 1: *Our GAN architecture generates continuous signed distance functions of shapes. Above objects are obtained via Marching Cubes.*

Abstract

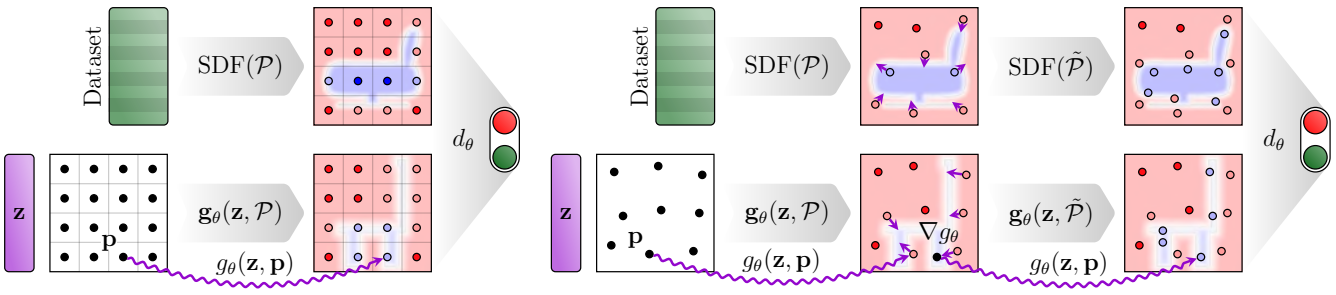
This work presents a generative adversarial architecture for generating three-dimensional shapes based on signed distance representations. While the deep generation of shapes has been mostly tackled by voxel and surface point cloud approaches, our generator learns to approximate the signed distance for any point in space given prior latent information. Although structurally similar to generative point cloud approaches, this formulation can be evaluated with arbitrary point density during inference, leading to fine-grained details in generated outputs. Furthermore, we study the effects of using either progressively growing voxel- or point-processing networks as discriminators, and propose a refinement scheme to strengthen the generator's capabilities in modeling the zero iso-surface decision boundary of shapes. We train our approach on the SHAPENET *benchmark dataset and validate, both quantitatively and qualitatively, its performance in generating realistic 3D shapes. Our source code is available under* https://github.com/marian42/shapegan.

1. Introduction

Shape synthesis tackles the task of generating new, diverse, and realistic shapes and is a central problem in many applications requiring high-quality 3D assets. Inspired by the success of generative adversarial networks (GANs), there has been a tremendous effort in applying those methods in the 3D domain [GFK*18, WZX*16, ADMG17]. In contrast to images, however, 3D shapes can be expressed using various different representations [ASS*18], each with their own trade-offs across fidelity and efficiency capabilities.

Recently, the implicit shape representation in the form of signed distance functions (SDFs) emerged to a powerful tool for reconstructing shapes from prior information [PFS*19]. Instead of discretizing space, this approach encodes a fully continuous distance field while being both efficient and expressive at the same time. In practice, this formulation has unlimited resolution, models arbitrary shape topography, ensures watertight surfaces and can be easily converted to voxels, meshes or point clouds. However, neural networks for continuous shape representations have been exclusively trained to learn latent embeddings via likelihood-based autoencoder or autodecoder schemes [PFS*19, CZ19, MON*19, LW19].

In this work, we want to explore how we can effectively produce continuous signed distance fields via generative adversarial modeling. Especially, we study how to combine generated continuous signed distance fields with either voxel- or point-processing discriminators, and show that both schemes can be easily enhanced by the usage of progressively growing GANs [KALL18]. Furthermore, we propose a refinement strategy to let our point-based discriminator better focus on the zero iso-surface of shapes, which also strengthens the generator's representational power in return.



(a) Voxel-based approach (b) Point-based approach

Figure 2: *Two-dimensional visualization of our generative pipeline: (a) The voxel-based approach uses a fixed number of stationary points as input to a 3D CNN discriminator. In contrast, (b) the point-based approach uniformly samples points in space, samples additional points near the surface based on $\nabla_{\mathbf{p}} g_\theta(\mathbf{z}, \mathcal{P})$, and inputs the refined point set $\tilde{\mathcal{P}}$ with their generated SDF values into a point-processing discriminator.*

2. Related Work

3D shape analysis and generation has a long history in computer vision and computer graphics. Here, we focus on the most directly related works using deep neural networks:

While neural networks can be used to classify *triangle meshes* [HHF*19, FLWM18], they are generally not well-suited to generate them directly due to inherent dicretization problems. Therefore, the problem of end-to-end mesh generation is typically tackled by learning deformations of primitives [SUHR17, WZL*18]. However, these approaches are often limited in generating meshes of the same genus as the base mesh. Further works [GFK*18, GYW*19] circumvent this problem by assembling multiple deformed primitives, but resulting meshes will not be closed, can have holes and primitives may overlap.

Successful image generation techniques based on convolutional neural networks (CNNs) [RMC15] have been also applied to rasterized 3D representations known as *voxel volumes*, where each voxel stores binary occupancy information [WZX*16] or an SDF value of the voxel center [DQN17]. However, voxel-based approaches are memory intensive, and are hence limited to low resolutions that can only model coarse-grained structures. Although data adaptive representations such as octrees [TDB17] can reduce the required memory footprint, it leads to complicated implementations while still being limited to rather small voxel grid sizes [MON*19].

Inspired by unstructured methods operating on unordered point sets [QSMG17], various methods have been proposed to encode or generate *surface point clouds* [ADMG17, YHCOZ18, VFM19]. However, those methods rely on reconstructing meshes in a potentially error-prone post-processing step [KH13], and are generally limited to generating point clouds of fixed sizes.

Recently, the encoding of shapes has been tackled by learning *continuous implicit functions* in the form of signed distances [PFS*19] or binary occupancies [CZ19, MON*19, LW19]. These representations have been shown to be both computationally and memory efficient and yet allow for obtaining or visualizing high-resolution geometry via Marching Cubes [LC87] or direct rendering using Sphere Tracing [Har96], respectively. However, neither of these approaches apply generative adversarial networks for implicit shape modeling and instead rely on likelihood-based (encoder-)decoder architectures. Differentiable variants of Sphere Tracing [LZP*19] and Marching Cubes [LDG18] have been proposed to train implicit shape and voxel representations.

3. Adversarial Generation of Implicit Shape Representations

Our generative adversarial network for synthesizing shapes consists of a generator and discriminator. Given a random latent code, our generator produces a continuous three-dimensional signed distance field, while the discriminator's job is to provide useful feedback to improve the generator. Our complete pipeline is shown in Figure 2, which we will now explain in more detail:

Signed Distance Functions. Given a spatial point $\mathbf{p} \in \mathbb{R}^3$, the *signed distance function* $\mathrm{SDF}(\mathbf{p}) \in \mathbb{R}$ encodes the point's distance to its closest surface point, where the sign indicates whether $\mathbf{p}$ lies inside $(-)$ or outside $(+)$ the object. In contrast to binary occupancy information, signed distance functions yield additional useful properties. For example, it allows us to sample a surface point cloud $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ of arbitrary cardinality N by translating uniformly sampled points $\mathbf{p}_i \in \mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ to their closest surface point:

$$\mathbf{s}_i = \mathbf{p}_i - \mathrm{SDF}(\mathbf{p}_i) \cdot \nabla_{\mathbf{p}_i} \mathrm{SDF}(\mathbf{p}_i) \tag{1}$$

Generator. We model our generator in close analogy to the DeepSDF [PFS*19] autodecoder architecture. Given a compact, low-dimensional encoding $\mathbf{z} \in \mathbb{R}^d$ of a complete shape, the DeepSDF decoder g_θ learns the mapping

$$g_\theta(\mathbf{z}, \mathbf{p}) \approx \mathrm{SDF}(\mathbf{p}), \tag{2}$$

where g_θ is parametrized via trainable parameters θ and is implemented as an MLP without the usage of any convolutional layers. Since g_θ is conditioned on a latent vector $\mathbf{z}$, the same neural network can be used to model the SDFs of multiple objects. Note that g_θ processes only a single point, but can be nonetheless effectively trained against the ground-truth SDF value. Overall, this formulation is advantageous over signed distance voxel grids [DQN17] since it allows us to model the SDF as a continuous function. However, it cannot be directly applied to a GAN setup since we argue

that a discriminator is not able to distinguish between real and fake solely based on the signed distance of a single sample point. Instead, the discriminator needs to validate the quality of the signed distance field as a whole. Therefore, we provide context to the discriminator by evaluating the generator for a batch of points $\mathcal{P}$, and inject the latent information $\mathbf{z}$ to each point $\mathbf{p}_i \in \mathcal{P}$ individually:

$$\mathbf{g}_\theta(\mathbf{z}, \mathcal{P})_i = g_\theta(\mathbf{z}, \mathbf{p}_i) \tag{3}$$

This scheme shares similar advantages to the DEEPSDF decoder. Although it requires fixed point sizes during training (in analogy to related point cloud approaches), we can still obtain high-resolution meshes during inference by varying the sampling density.

We now proceed to present two discriminator variants, a *voxel-based* and a *point-based* approach, to discriminate between real and generated signed distance fields:

Voxel-based Discriminator. In our first variant, we explore the possibility of using a 3D CNN as our discriminator d_θ similar to the one proposed by Wu *et al*. [WZX*16]. Here, both $\mathbf{g}_\theta(\mathbf{z}, \mathcal{P})$ and $d_\theta(\mathbf{g}_\theta(\mathbf{z}, \mathcal{P}), \mathcal{P})$ expect input points $\mathcal{P}$ to describe a regular grid of fixed resolution. Hence, $\mathbf{g}_\theta$ and d_θ can be easily combined by rearranging the generated SDF values from $\mathbf{g}_\theta$ into a voxel volume. Although being straightforward to implement, this approach shares *some* of the disadvantages of related voxel-based approaches: It only makes use of a fixed number of stationary points to discriminate between real and fake examples, and is furthermore quite memory-inefficient to train. However, since g_θ is modeled to be continuous (in contrast to related approaches [WZX*16, DQN17]), we can even expect reasonable outputs for points never seen during training, *cf.* Section 4.

Point-based Discriminator. An alternative to modeling the discriminator d_θ in a voxel-based fashion builts upon recent advances in permutation-invariant point-processing networks. Instead of only providing the discriminator with stationary point samples, we may input *any* point by making use of the POINTNET architecture [QSMG17] as our discriminator d_θ [ADMG17, VFM19]. Here, uniformly distributed points are first transformed *independently* into a high-dimensional space before a joint representation is obtained using the permutation-invariant max-pooling operation

$$d_\theta(\mathbf{g}_\theta(\mathbf{z}, \mathcal{P}), \mathcal{P}) = \gamma_\theta\left(\max_{\mathbf{p}\in\mathcal{P}} h_\theta(g_\theta(\mathbf{z}, \mathbf{p}), \mathbf{p})\right) \in \mathbb{R} \tag{4}$$

with $\mathbf{p} \sim \mathcal{U}(-1, 1)^3$, and γ_θ and h_θ denoting trainable MLPs [QSMG17]. In contrast to the original POINTNET proposal, we also inject the respective signed distance to each raw point so that g_θ takes both positional and signed distance information into account. Compared to the voxel-based approach, this formulation allows g_θ to know about *any* point in space, instead of solely being required to generate reasonable outputs for a fixed number of stationary points.

Note that d_θ is not limited to the POINTNET architecture, but may employ any network architecture that is able to process irregularly structured data in a permutation-invariant fashion. For example, the recent works in the field of *geometric deep learning* provides a large number of operators to choose from [ASS*18], with potential capabililties to also take relational information into account. Extensions like POINTNET++ [QYSG17] may even improve upon the results presented in this paper since those methods should be able to capture more fine-grained details. We leave the usage of those discriminators for future work.

Zero Iso-surface Decision Boundary. For reconstructing meshes from SDFs, we are mostly interested in the precise modeling of the zero iso-surface decision boundary and care less about maintaining a metric SDF for larger distances. While DEEPSDF overcomes this problem via a more aggressively sampling near the surface of an object, this solution cannot be applied directly since we do not know about the object's surface in advance. Nonetheless, $\mathbf{g}_\theta(\mathbf{z}, \mathcal{P})$ already provides great capabililties to accurately predict the coarse-grained surface of an object. Following up on Equation (1), we strengthen the generator's representational power by sampling additional points near the surface of an object based on the *generated* SDF values of uniformly distributed points

$$\tilde{\mathbf{g}}_\theta(\mathbf{z}, \mathcal{P}) = \mathbf{g}_\theta(\mathbf{z}, \mathcal{P}) \bigcup_{\substack{\mathbf{p}\in\mathcal{P} \\ |g_\theta(\mathbf{z},\mathbf{p})|<\delta}} \left\{ g_\theta(\mathbf{z}, \mathbf{p} - g_\theta(\mathbf{z}, \mathbf{p}) \cdot \nabla_\mathbf{p} g_\theta(\mathbf{z}, \mathbf{p}) + \varepsilon) \right\} \tag{5}$$

with $\varepsilon \sim \mathcal{N}(0, \sigma^2)^3$. Here, for each point $\mathbf{p} \in \mathcal{P}$ that is sufficiently close to a predicted surface ($|g_\theta(\mathbf{z}, \mathbf{p})| < \delta$), we project it onto the surface using the gradients of g_θ w.r.t. $\mathbf{p}$, and sample additional points following a gaussian distribution. The discriminator then takes the *refined point set*

$$\tilde{\mathcal{P}} = \mathcal{P} \cup \{\mathbf{p} - g_\theta(\mathbf{z}, \mathbf{p}) \cdot \nabla_\mathbf{p} g_\theta(\mathbf{z}, \mathbf{p}) + \varepsilon\} \tag{6}$$

and their respective SDF values as input, and can hence more specifically draw its attention to the modeling of the zero iso-surface decision boundary. Note that $\tilde{\mathbf{g}}_\theta$ is still fully differentiable and can hence be trained in an end-to-end fashion using SGD.

Training. For training our generative adversarial networks, we make use of pre-processed ground-truth distance fields SDF($\mathcal{P}$) stemming from human designed meshes. We input both generated and real samples $d_\theta(\mathbf{g}_\theta(\mathbf{z}, \mathcal{P}), \mathcal{P})$ and $d_\theta(\text{SDF}(\mathcal{P}), \mathcal{P})$ to the discriminator d_θ, respectively, while optimizing for the typical GAN minimax objective [GPAM*14]. Furthermore, we noticed that making use of progressively growing GANs [KALL18] improves the training stability and quality of generated objects. That is, we progressively increase voxel resolution and network depth in our voxel-based discriminator, and simply increase the number of point samples in our point-based architecture as training proceeds. Our generative network $\mathbf{g}_\theta$ does not need to be increased in both cases, since it is already invariant towards specific resolutions by design.

4. Experiments

We trained our proposed generative architectures on the chair and airplane categories taken from the SHAPENET repository [CFG*]. We randomly divide shapes into train/validation/test sets using an 85%-5%-10% split ratio. The data preparation pipeline needs to handle non-watertight meshes and meshes with inconsistently oriented normals, as both appear in the dataset, *cf.* Figure 4. Our approach to data preparation is based on the method employed by Park *et al*. [PFS*19]. We render each mesh from 50 equidistant camera angles and project the depth buffers back into object space,

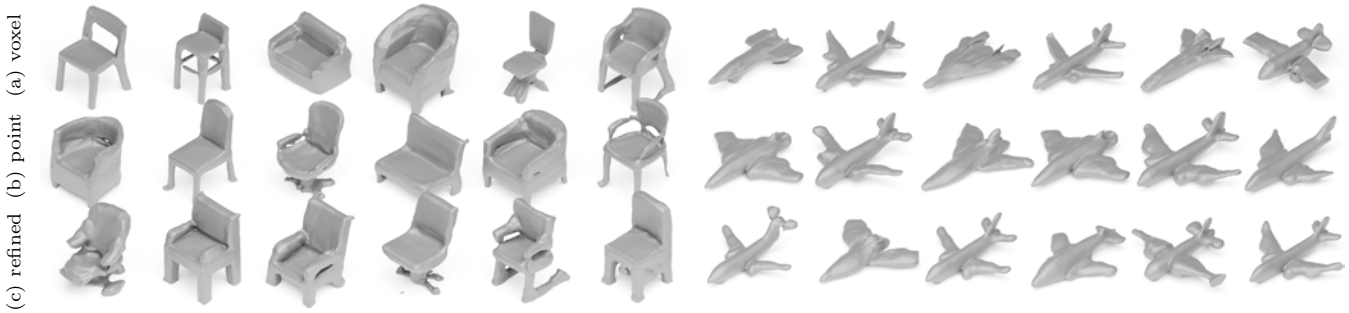


Figure 3: *Qualitative examples of generated shapes for (a) the voxel-based discriminator (top), (b) the point-based discriminator (middle) and (c) the refined point-based discriminator (bottom).*

Figure 4: *Calculating SDFs: (a) Small triangles with deviating normals may produce artifacts in the SDFs [PFS*19] and (b) are avoided by using depth information to determine signs. (c) Non-watertight meshes lead to discontinuous SDFs and are discarded.*

Figure 5: *Latent space interpolation. The first and last latent codes were obtained by overfitting shapes of the dataset and the remaining latent codes were obtained by linear interpolation. The images are rendered with Sphere Tracing [Har96] from signed distance fields given by our generator network.*

resulting in a surface point cloud. For each query point, we then calculate the distance to its closest surface point. To determine the sign, we transform the sample point into the viewport coordinates of each render. We use the depth buffers to check if the point is closer to the camera than the surface of the shape at the corresponding pixel. A point is considered outside the mesh if it is seen by any of the virtual cameras. We discard objects with discontinuous SDFs and those with less than 1% of uniformly sampled points inside the shape, resulting in 4 189 and 2 156 examples for the chair and airplane categories, respectively. Our pre-processing pipeline is publicly available on GitHub.[†]

[†] https://github.com/marian42/mesh_to_sdf

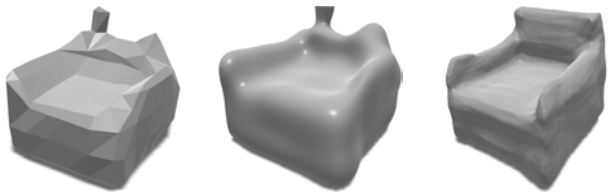
Table 1: *Quantitative evaluation across* ShapeNet *categories.*

	Method		JSD	MMD-CD	MMD-EMD	COV-CD	COV-EMD
Chair	[ADMG17]		0.238	**0.0029**	0.136	**33**	13
	[VFM19]		0.100	**0.0029**	0.097	30	26
	Ours	voxel	**0.076**	0.0037	0.087	**33**	**36**
		point	0.082	0.0036	0.088	28	29
		refined	0.078	0.0037	**0.086**	30	28
Airplane	[ADMG17]		0.182	0.0009	0.094	31	9
	[VFM19]		0.083	**0.0008**	0.071	31	14
	Ours	voxel	0.093	0.0019	**0.066**	31	**34**
		point	0.114	0.0026	0.078	28	31
		refined	**0.072**	0.0019	0.070	**36**	31

Architecture and Parameters. Our generator g_θ follows the design of DeepSDF [PFS*19] and is composed of 8 fully connected layers with hidden dimensionalities of 256, layer normalization and ReLU activiation. The input is once again reinjected via concatenation after the fourth layer. We use 128-dimensional latent vectors $\mathbf{z}$ sampled from a normal distribution. The voxel-based discriminator uses 3D convolutions with a stride of 2 and a kernel size of 4 to achieve powers of 2 for all intermediate voxel resolutions, followed by two dense layers (128, 1) [WZX*16]. We use four steps of progressive growing [KALL18] with samples of increasing resolution (8^3, 16^3, 32^3, 64^3). The PointNet discriminator [QSMG17] uses 4 layers with weights shared across the point dimension (64, 128, 256 and 512), followed by global max-pooling and 3 dense layers (256, 128, 1). We uniformly sample 64^3 points for ground-truth SDF values per object, while only using a small but progressively growing subset of points as input. For training with refined points $\tilde{\mathcal{P}}$, we sample additional ground-truth values according to Equation (6).

For optimization, we make use of the WGAN-GP [GAA*17] objective and employ RMSprop with a fixed learning rate of 10^{-4}. All models have been trained for a maximum of 2 000 epochs, while we select the final model based on validation results.

Results. Qualitative results of our models are shown in Figures 3 and 1. From a visual standpoint, all models generate a diverse set of convincing shapes, and are able to produce, *e.g.*, thin chair legs and a precise modeling of the airplane tail. In Figure 5, we demonstrate



(a) 8^3 (b) 8^3 interpolated to 128^3 (c) 128^3

Figure 6: *Generalization capabililties of our voxel-based approach when trained on* 8^3 *voxels: (a) A random latent code evaluated at* 8^3 *raster points, (b)* 8^3 *raster points linearly upscaled to* 128^3*, (c) the same latent code evaluated with* 128^3 *raster points.*

latent space continuity of our generator by linearly interpolating between latent codes of dataset shapes.

For a quantitative evaluation, we follow the experimental protocol of recent generative point cloud approaches [ADMG17, VFM19]. We report the Jensen-Shannon divergence (JSD) between marginal distributions in the 3D space, and the coverage (COV) and minimum matching distance (MMD) based on the Chamfer distance (CD) and the earth mover's distance (EMD) between point sets, *cf.* Table 1. Point clouds are obtained by sampling 2 048 points on meshes generated via Marching Cubes. Given that the Chamfer distance is known to be unreliable [ADMG17, VFM19], our results achieve better or equal values for the metrics under consideration. In general, the refined point-based approach improves the results of using uniformly sampled points, while there is not yet a clear winner between the voxel- and point-based approaches.

Furthermore, the continuous formulation of our generator provides great generalization capabililties even for points never seen during training. This is shown by the high-resolution examples from the voxel-based approach in Figure 3 and a case study in Figure 6.

5. Conclusion

We presented two methods to produce continuous signed distance fields via generative adversarial modeling. Further works include the study of more expressive point-processing discriminators, and the addition of regularization schemes to penalize invalid SDFs. In addition, discriminator architectures based on differentiable Marching Cubes [LDG18] or differentiable Sphere Tracing [LZP*19] could be examined.

Acknowledgments

This work has been supported by the *German Research Association (DFG)* within the Collaborative Research Center SFB 876, *Providing Information by Resource-Constrained Analysis*, project A6.

References

[ADMG17] ACHLIOPTAS P., DIAMANTI O., MITLIAGKAS I., GUIBAS L.: Learning representations and generative models for 3D point clouds. *CoRR* (2017). arXiv:1707.02392. 1, 2, 3, 4, 5

[ASS*18] AHMED E., SAINT A., SHABAYEK A., CHERENKOVA K., DAS R., GUSEV G., AOUADA D.: A survey on deep learning advances on different 3D data representations. *CoRR* (2018). arXiv:1808.01462. 1, 3

[CFG*] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: ShapeNet: An information-rich 3D model repository. *CoRR*. arXiv:1512.03012. 3

[CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *CVPR* (2019). 1, 2

[DQN17] DAI A., QI C. R., NIESSNER M.: Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *CVPR* (2017). 2, 3

[FLWM18] FEY M., LENSSEN J. E., WEICHERT F., MÜLLER H.: SplineCNN: Fast geometric deep learning with continuous B-Spline kernels. In *CVPR* (2018). 2

[GAA*17] GULRAJANI I., AHMED F., ARJOVSKY M., DUMOULIN V., COURVILLE A. C.: Improved training of Wasserstein GANs. In *NIPS* (2017). 4

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR* (2018). 1, 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *NIPS* (2014). 3

[GYW*19] GAO L., YANG J., WU T., YUAN Y.-J., FU H., LAI Y.-K., ZHANG H.: SDM-NET: Deep generative network for structured deformable mesh. *CoRR* (2019). arXiv:1908.04520. 2

[Har96] HART J. C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer 12*, 10 (1996), 527–545. 2, 4

[HHF*19] HANOCKA R., HERTZ A., FISH N., GIRYES R., FLEISHMAN S., COHEN-OR D.: MeshCNN: A network with an edge. In *SIGGRAPH* (2019). 2

[KALL18] KARRAS T., AILA T., LAINE S., LEHTINEN J.: Progressive growing of GANs for improved quality, stability, and variation. In *ICLR* (2018). 1, 3, 4

[KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Transactions on Graphics 32*, 3 (2013), 29:1–29:13. 2

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH* (1987). 2

[LDG18] LIAO Y., DONNE S., GEIGER A.: Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2916–2925. 2, 5

[LW19] LITTWIN G., WOLF L.: Deep meta functionals for shape representation. In *ICCV* (2019). 1, 2

[LZP*19] LIU S., ZHANG Y., PENG S., SHI B., POLLEFEYS M., CUI Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. *arXiv preprint* (2019). arXiv:1911.13225. 2, 5

[MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3D reconstruction in function space. In *CVPR* (2019). 1, 2

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR* (2019). 1, 2, 3, 4

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR* (2017). 2, 3, 4

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS* (2017). 3

[RMC15] RADFORD A., METZ L., CHINTALA S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR* (2015). arXiv:1511.06434. 2

[SUHR17] SINHA A., UNMESH A., HUANG Q., RAMANI K.: SurfNet: Generating 3D shape surfaces using deep residual networks. In *CVPR* (2017). 2

[TDB17] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *ICCV* (2017). 2

[VFM19] VALSESIA D., FRACASTORO G., MAGLI E.: Learning localized generative models for 3d point clouds via graph convolution. In *ICLR* (2019). 2, 3, 4, 5

[WZL*18] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.-G.: Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV* (2018). 2

[WZX*16] WU J., ZHANG C., XUE T., FREEMAN B., TENENBAUM J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NIPS* (2016). 1, 2, 3, 4

[YHCOZ18] YIN K., HUANG H., COHEN-OR D., ZHANG H.: P2P-NET: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics 37*, 4 (2018), 152. 2