Multi-Object Occupancy Networks

Gianluca Galletti*, Nina Kirakosyan*
{g.galletti, nina.kirakosyan}@tum.de

Abstract

Reliably reconstructing full 3D scenes from sparse representations, such as point clouds, has been a very active topic of research in the recent years. Learnable 3D implicit functions are a popular class of models that have been shown to be very successful when applied to this task. Despite the promising results, most of the current SotA models still consider the entire scene as one object, and discard any semantic information coming from the objects placed in the scene. We propose Multi-Object Occupancy Networks (MOONets in short), which are a novel set of implicit functions that explicitly make use of object segmentation, by encoding, decoding and reconstructing each object in the scene separately. This has the double advantage of leveraging semantic information from instance segmentation, and inducing perfect robustness towards object transformations. Additionally, we contribute with our custom synthetic room dataset, augmented with ground truth object instance segmentation. We achieve results comparable to the state of the art on the synthetic rooms dataset. While not improving on previous works, our method enables reliable on-the-fly reconstruction of scenes with any object-wise transformation, with a single forward pass and without any loss of performance. Source code for both the model architecture and the custom dataset generation scripts is available at github.com/gerkone/multiobject-onet.

1. Motivation

Recently, learnable implicit functions have been shown to be highly capable at reconstructing 3D meshes from sparse point clouds. Despite the great success, when applied to very complex shapes, such as room scans, most state of the art methods sometimes partially, or even completely, fail to capture some objects in detail. Furthermore, any non-equivariant SotA method fails to robustly capture relative object transformations; namely, reconstructing the same room with an object moved or rotated can lead to completely different outputs.

With Multi-Object Occupancy Networks we try to address these two shortcomings by incorporating semantic information coming from an instance segmentation network directly into an ONet-like architecture, in order to predict end-to-end one occupancy grid per object. With this object-wise occupancy it is then possible to naturally produce a segmented mesh for the room, where each segment can potentially be independently transformed to produce out-of-distribution scenes (e.g. with objects upside down) which other models would fail to reconstruct.

2. Related work

The multi-object scene reconstruction pipeline encompasses several separate task definitions, including object reconstruction from point cloud data, multi-object scene understanding, and 3D instance segmentation.

2.1. Working with point clouds

Special architectures have been proposed for directly working with point cloud data. Most notably, PointNet [1, 2] architecture addresses the issue of working directly with unordered point cloud sets. It is widely used in various point cloud applications, such as semantic and part segmentation, and classification. Alternatively, Hierarchical UNet GNNs have also been used in similar settings [3, 4].

More recent works, Vector-Neurons [5] and E-GraphONet [6] extend the 1D neurons to 3D vector space to map geometry into the latent space. With introduced equivariant layers such approaches improve model generalization over a set of geometric transformations.

2.2. Occupancy Networks

Occupancy Network. Current widely used approaches for 3D representation often rely on implicit representations by Occupancy Networks (ONet) [7]. This approach learns a continuous 3D mapping of the surface geometry. The idea is to reason about the 3D representation not as a discrete space, like voxels or meshes, but as a continuous function, defined at each possible value in the relevant 3D space.

The occupancy function is parametrized by a neural network, which takes the location $p \in \mathbb{R}^3$ and observation $x \in X$ as input, and outputs a real value y. The output represents the occupancy probability for the sampled location.

$$f_\theta : \mathbb{R}^3 \times X \to [0, 1]$$

The inferred volumetric space can be then discretized by calculating the occupancy for each point in a sample grid of certain resolution, and setting points above a chosen threshold to be occupied.

Convolutional Occupancy Networks. Some limitations of implicit representations like occupancy networks have been addressed by ConvONet [8]. It replaces the less expressive fully connected architecture with convolutions, resulting in translation equivariant representation. Incorporation of convolutions also allows for scalability to more complex, multi-object scenes.

2.3. Multi-object reconstruction

The task of multi-object scene reconstruction and understanding is an active research area of interest. Total 3D Understanding [9] introduced an end-to-end pipeline for reconstructing

[*] Equal contribution

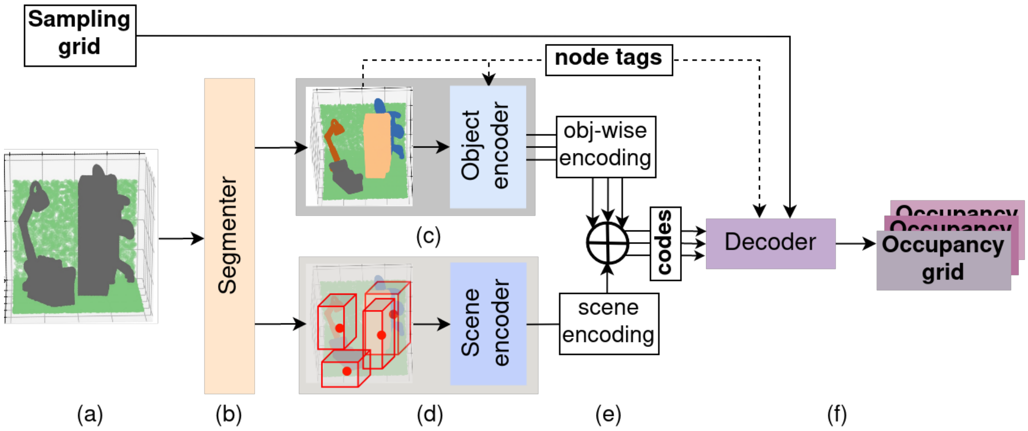

Figure 1. MOONet architecture. Solid lines represent features, dashed lines represent indices. (a) Inputs: point cloud and sampling points. (b) Instance segmentation model, returning segmented point cloud and keypoints. (c) Segmented point cloud (normalized object positions) and *object-wise encoder* E^{obj}. (d) Keypoints (bounding boxes and centroids) and *scene-level encoder*E^{scn}. (e) *latent codes* c as concatenation of object c_{obj} and scene c_{scn} embeddings. (f) Multi-object decoder D and final *per-object occupancy grid* $O_{[o,j]}$.

joint room layout, object pose and mesh reconstruction for indoor scenes from a single image. Another example of scene understanding from image data is Holistic 3D Scene Understanding pipeline [10], which, again, uses implicit representations to predict object shape, object pose, and scene layout.

Attempting to extend scene understanding tasks directly to point cloud data is a particularly challenging problem. VoteNet [11] proposed a way of adapting deep Hough voting for point cloud setting. The model is able to detect object centers and combine the votes to output bounding boxes and semantic labels. Semantic scene understanding from point data was addressed by RfD-Net [12], which performs semantic instance reconstruction. Here object semantics and shapes are learnt using reconstruction form prediction principle, disentangling object localization from shape prediction.

2.4. Instance segmentation

Instance segmentation is a combination of the object detection and semantic segmentation tasks. Most methods approach the problem via breaking down this two-tier problem setting.

One way to achieve this is by trying to retrieve semantic labels, followed by clustering/grouping of the segmented objects into distinct object instances. SoftGroup [13, 14] is an example of such a method, where each point is allowed to be bottom-up grouped into multiple object classes, and then top-down refined into a separate instance. Alternatively, other approaches first detect object instances, followed by segmentation, as done in 3D-BoNet [15] by regressing objects' bounding boxes, then segmenting foreground masks for each box. More recent approaches, like the transformer-based Mask3D [16], try to directly predict instance masks and their semantic labels using the attention mechanism.

3. Multi-Object Occupancy Networks

3.1. Model architecture

Figure 1 shows the full model architecture. Follows a more detailed explanation of each component.

(a) **Inputs**. The inputs are a 3D point cloud X_i, made up of point positions (no colors) and the *"query points"* p_j, a 3D sampling grid representing the points where the network is expected to return the occupancy probability O.

(b) **Instance segmentation**. The segmenter takes care of segmenting the input point cloud into single object instances. This is done by assigning an object ID, which we call **node tag** T_i, to every point in the point cloud. Points in the same object will have the same tag. Additionally, the segmenter extracts *keypoints*, per-object bounding boxes bb_o and centroids c_o:

$$bb_o = \left\{min(X_{obj(o)}), max(X_{obj(o)})\right\},$$
$$c_o = \frac{1}{N}\sum_{i \in obj(o)} X_i,$$
$$\text{where } obj(o) = \{i \in (1, N) \mid T_i = o\}$$

(c) **Object encoding**. The object encoder $c_{obj} := E^{obj}(X, T)$ embeds object-wise information into a point-wise code (shape $(cdim_{obj}, N)$). Since the information propagation is restricted by the node tags, the object encoder is focused only on single objects, as information can only be exchanged intra-object. On top of this, point positions X^{obj} are normalized with regards to the scene centroid to avoid introducing unwanted scene information in the network $X_i^{obj} = X_i - \frac{1}{N}\sum_i^N X_i$.

(d) **Scene encoding**. The scene encoder $c_{scn} := E^{scn}(bb, c)$ embeds the keypoints extracted by the segmenter into a global object-wise code (shape $(cdim_{scn}, n_{obj})$). Scene-level information (inter-object relative positions) only enters MOONet through the E^{scn}.

(e) **Codes**. The latent codes are build from object and scene codes ($c = c_{obj} \oplus c_{scn}$). They represent a point embedding in scene-level latent space, which are then used to condition the deconding process.

(f) **Decoding**. The decoder $O := D(\{c, X\}, p, T)$ reconstructs one occupancy grid per object. It takes the 3D query points as input and is conditioned on the latent codes. In particular, the returned occupancy grid $O_{[o,j]}$ is indexed by the node tag o and the query point index j.

3.2. Encoders, decoders

Because encoder and decoder models proposed in previous works cannot be directly used in a multi-object setting, we had to come up with our versions of a few of the more common architectures. In particular, we introduce two new Dynamic Graph Convolution [17] models, one encoder and one decoder, and an adapted PointNet++ [2] encoder.

In general, components of MOONet rely on a special object indexing (**node tags**). Our models are designed to make use of this indexing, for example to limit feature propagation only between intra-object points (i.e. points in the same objects, with the same node tag) and drop intra-object connections (i.e. edges that cross objects, between points with a different node tag).

MO-DGCNN. Our multi-object Dynamic Graph Convolution (MO-DGCNN) comes in two flavours: one encoder which takes the point cloud as input and returns point embeddings, and one decoder which takes both point cloud and sample grid points as input and return per-object occupancies. Figure 2 shows the *grid-MO-DGCNN* decoder.

The core differences between regular DGCNN [17] and our multi-object version is that the k-NN search used for the hierarchical features is restricted to intra-object edges, meaning that there is no feature propagation across objects. On top of this, the *grid-MO-DGCNN* decoder uses the pooling indices from the nearest point to reconstruct the object occupancies.

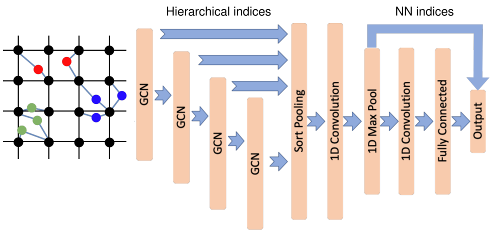

Figure 2. MO-DGCNN model applied to sample grids as well as point clouds. This particular architecture is used as decoder. Node tag is represented as point color.

MO-PointNet++. The multi-object PointNet++ models extend the standard PointNet++ architecture [2] with *hierarchical indices* on top of points and features. When coarse-graining, our model only pools within the same object, and keeps track of which index is assigned to which fine point. This way it can later reconstruct and assign the correct points during interpolation. Figure 3 shows the MO-PointNet++ model.

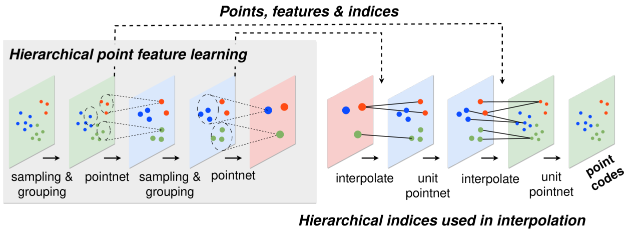

Figure 3. MO-PointNet++ model. Hierarchical coarse graining, keeps track of points, features and indices while pooling (left); Interpolation of coarse points reconstructs the object using the hierarchical indices (right). Node tag is represented as point color.

3.3. Instance segmentation

For our pipeline completeness, we also investigated possible approaches for point cloud instance segmentation. In particular, we experimented with the SoftGroup [13, 14] model for our pipeline. We managed to integrate our custom Synthetic Indoor Scene Dataset into the SoftGroup flow.

During our investigation, we had to deal with the challenge of mismatch between the data format and properties between the available instance segmentation approaches and our setting. These methods rely both on positional pointwise information, as well as additional point features like color properties, which are not supported by our dataset. Regenerating ShapeNet objects, extracting that additional data was, unfortunately, unattainable with our resources in the scope of this project. Consequently, we cannot present suitable results for the segmentation task on our dataset at this stage, and we leave this challenge to be addressed in future work.

4. Dataset

For object instance-level scene understanding, our task required a dataset of multi-object scenes with instance labels for each point in the input point cloud, as well as the ground truth sample grid.

Our baseline model ConvONet uses a custom Synthetic Indoor Scene Dataset for scene-level mesh reconstruction. This is a dataset of 5000 synthetic scenes, containing multiple ShapeNet [18] objects. It in turn relies on the subset of preprocessed ShapeNet objects from the original ONet [7] pipeline. The full preprocessed dataset is made publicly available by the authors, however it does not contain instance-level pointwise labels, instead providing pointwise semantic labels. As the scenes contain (possibly colliding) objects from repeating classes, the available point-wise information was not enough for extracting ground truth instance labels necessary for our framework.

We regenerated our custom version of the Synthetic Indoor Scene Dataset, with separate semantic, as well as instance-level pointwise information. As in ConvONet, we used 5 object classes from ShapeNet data: chair, table, sofa, lamp and cabinet. Each scene contains 4-8 randomly sampled objects of the given classes. For each number of objects (4-8) we gen-

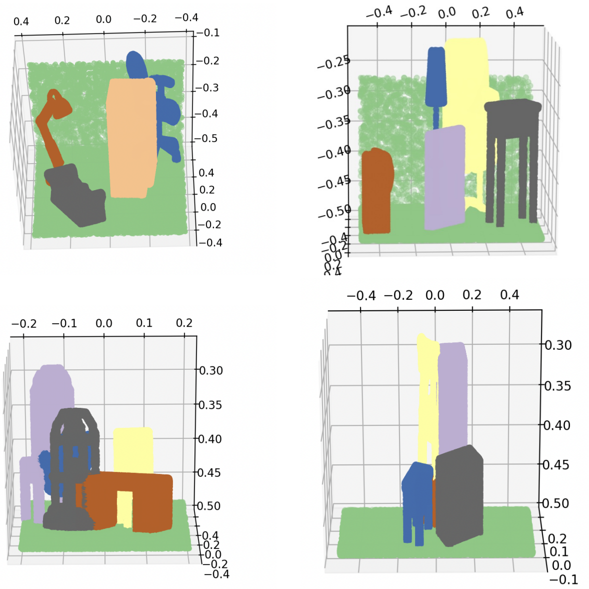

Figure 4. Point clouds from our Synthetic Indoor Scene Dataset with semantic and instance information. Color represent the node tag of every point. Green points represent walls and floor planes.

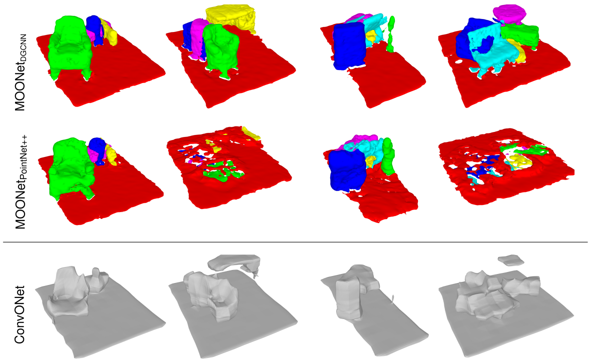

Figure 5. 3D reconstructions of different scenes from point clouds. Objects are marked with different colors (when possible).

erated 1000 unique scenes, with different object placements and scales. Figure 4 shows four samples of generated scene point clouds, with segmented ground-truth object instances. For each scene we sampled 10 distinct samples of the scene point cloud and the sample occupancy grid, resulting in a total of 50 thousand unique input samples. The dataset was divided into train-validation-test sets with 75-5-20 percent ratio.

5. Results

5.1. IoU against baseline

In Table 1, MOONet shows to significantly outperform the first occupancy network model [7] on all IoU thresholds. While not surpassing ConvONet [8] on lower threshold, our MOONet$_{DGCNN}$ is shown to be considerably better on thresholds $> 70\%$.

Table 1. MOONet and baseline Intersection-over-Union results at different thresholds on the *ssr* dataset. All MOONet models use *grid-MO-DGCNN* decoders. The subtext *DGCNN* and *PN++* refers to MO-DGCNN and MO-PointNet++ encoder respectively. *"MOONet (w/o seg)"* refers to a MOONet model without segmentation, where the whole scene is treated as a single object.

Model	Iou@50	IoU@90	IoU@95
Onet	0.36	0.30	0.27
ConvONet	**0.61**	0.44	0.40
MOONet$_{DGCNN}$	0.35	**0.49**	**0.54**
MOONet$_{PN++}$	0.33	0.38	0.38
MOONet (w/o seg)	0.32	0.44	0.48

Figure 5 shows different rooms reconstructed by MOONet and ConvONet. Interestingly, MOONet with MO-PointNet++ as encoder exhibits a few outlier reconstructions where the occupancy is "squashed" on the ground plane. This could derive from a weak scene understanding which originates from our adaptation of the farthest-point sampling algorithm [2]. We found that that sometimes ConvONet (or any model without access to semantic information) misses objects, while MOONet is less prone to do so, possibly due to how object and scene understanding are split and then combined.

5.2. Object-wise transformations

One of the advantages of MOONet is that it allows object transformations on-the-fly, meaning that once a scene has been generated, objects can be transformed without additional forward calls; in particular, to do the same with ConvONet one would need to call the model for every transformation. Figure 6 shows this capability when moving the green chair; MOONet models with segmentation are faster and more accurate when reconstructing the transformed scenes.

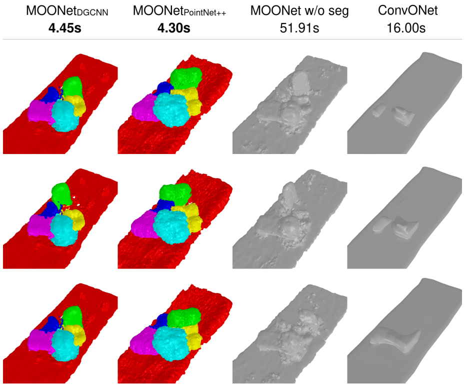

Figure 6. Scene reconstructions with object translations. The times are reported for 50 reconstructions with different translation vetors.

5.3. Training details

All models were trained until convergence for a maximum of 48 hours on the full extent ($\sim$ 40k training samples) of the *semantic synthetic room* dataset. ONet and ConvONet have 1.7M and 1.5M parameters respectively; ONet uses a ResNet encoder and decoder, while ConvONet uses triplane convolution [8]. As for MOONet, *MOONet$_{DGCNN}$* and *MOONet (w/o seg)* use a MO-DGCNN encoder with 128 hidden size, 4 blocks and $k = 30$ neighbours for the k-NN search, resulting is 750K params. *MOONet$_{PN++}$* uses a MO-PointNet++ encoder with 256 hidden size and 3 hierarchical blocks, with 1.5M params. The decoder is the same for all MOONet models: *grid-MO-DGCNN* with 128 hidden size and 5 blocks. The scene encoder is the same as well, and a custom fully connected EGNN [19] model is used.

6. Discussion

We presented MOONet, a novel set of architectures for direct multi-object 3D reconstruction from point clouds, which naturally produce a segmented object-wise mesh. Abeit not outperforming the previous state of the art, we show two advantages of our architecture: the core benefit of a direct multi-object mesh is the possibility to apply geometric transformations to the object cheaply: only one initial forward pass per scene is needed, after that objects can be transformed independently. Additionally, we observe that introducing semantic information, such as the instance segmentation, can improve performance with the relatively small added cost of segmentation. MOONets also have two significant drawbacks: first, runtime is one order of magnitude worse than ONet or ConvONet. We found that the k-NN search with a large k present in both encoders and decoder is the main source of the bad runtime. Second, the occupancy is almost a 0-1 classifier, as highlighted by higher thresholds in Table 1.

Finally, we suggest that future works in the direction of Multi-Object Occupancy Networks could explore a) dedicated multi-object encoders and decoders, to improve performance and speed up inference; b) fully integrated segmentation networks; c) even faster multi-mesh generation, with a custom multi-object MISE [7] algorithm.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” 2017. 1

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” 2017. 1, 3, 4

[3] M. Meraz, M. A. Ansari, M. Javed, and P. Chakraborty, “Dc-gnn: drop channel graph neural network for object classification and part segmentation in the point cloud,” *International Journal of Multimedia Information Retrieval*, vol. 11, pp. 123–133, Jun 2022. 1

[4] W. Shi, Ragunathan, and Rajkumar, “Point-gnn: Graph neural network for 3d object detection in a point cloud,” 2020. 1

[5] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas, “Vector neurons: A general framework for so(3)-equivariant networks,” *CoRR*, vol. abs/2104.12229, 2021. 1

[6] Y. Chen, B. Fernando, H. Bilen, M. Nießner, and E. Gavves, “3d equivariant graph implicit functions,” 2022. 1

[7] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” 2019. 1, 3, 4

[8] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *European Conference on Computer Vision (ECCV)*, 2020. 1, 4

[9] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. Zhang, “Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 52–61, 2020. 1

[10] C. Zhang, Z. Cui, Y. Zhang, B. Zeng, M. Pollefeys, and S. Liu, “Holistic 3d scene understanding from a single image with implicit representation,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8829–8838, 2021. 2

[11] C. R. Qi, O. Litany, K. He, and L. J. Guibas, “Deep hough voting for 3d object detection in point clouds,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[12] Y. Nie, J. Hou, X. Han, and M. Niessner, “Rfd-net: Point scene understanding by semantic instance reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4608–4618, June 2021. 2

[13] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo, “Softgroup for 3d instance segmentation on 3d point clouds,” in *CVPR*, 2022. 2, 3

[14] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C.-D. Yoo, “Softgroup for 3d instance segmentation on point clouds,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2698–2707, 2022. 2, 3

[15] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, “Learning object bounding boxes for 3d instance segmentation on point clouds,” in *Advances in Neural Information Processing Systems*, pp. 6737–6746, 2019. 2

[16] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, “Mask3D for 3D Semantic Instance Segmentation,” 2023. 2

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” *CoRR*, vol. abs/1801.07829, 2018. 3

[18] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3

[19] V. G. Satorras, E. Hoogeboom, and M. Welling, “E(n) equivariant graph neural networks,” *CoRR*, vol. abs/2102.09844, 2021. 4