Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
Sister Conferences Best Papers Track

On Learning Sets of Symmetric Elements (Extended Abstract)*

Haggai Maron[1†], **Or Litany**[1], **Gal Chechik**[1,2] and **Ethan Fetaya**[2]
[1]NVIDIA
[2]Bar-Ilan University

Abstract

Learning from unordered sets is a fundamental learning setup, recently attracting increasing attention. Research in this area has focused on the case where elements of the set are represented by feature vectors, and far less emphasis has been given to the common case where set elements themselves adhere to their own symmetries. That case is relevant to numerous applications, from deblurring image bursts to multi-view 3D shape recognition and reconstruction. In this paper, we present a principled approach to learning sets of general symmetric elements. We first characterize the space of linear layers that are equivariant both to element reordering and to the inherent symmetries of elements, like translation in the case of images. We further show that networks that are composed of these layers, called *Deep Sets for Symmetric elements* layers (DSS), are universal approximators of both invariant and equivariant functions, and that these networks are strictly more expressive than Siamese networks. DSS layers are also straightforward to implement. Finally, we show that they improve over existing set-learning architectures in a series of experiments with images, graphs, and point clouds.

1 Introduction

Learning with data that consists of unordered sets of elements is an important problem with numerous applications, from classification and segmentation of 3D data [Zaheer *et al.*, 2017; Qi *et al.*, 2017; Su *et al.*, 2015; Kalogerakis *et al.*, 2017] to image deblurring [Aittala and Durand, 2018]. In this setting, each data point consists of a set of elements, and the task is independent of element order. This independence induces symmetry structure, which can be used to design deep models with improved efficiency and generalization. Indeed, models that respect set symmetries, e.g. [Zaheer *et al.*, 2017; Qi *et al.*, 2017], have become the leading approach for solving such tasks.

Importantly, the elements of the set themselves often adhere to certain symmetries, as happens when learning with sets of images, sets of point clouds, and sets of graphs. It is still unknown what is the best way to utilize these additional symmetries.

A common approach to handle per-element symmetries is based on processing elements individually. First, one processes each set-element independently into a feature vector using a Siamese architecture [Bromley *et al.*, 1994], and only then fuses information across all feature vectors. When following this process, the interaction between the elements of the set only occurs after each element has already been processed, possibly omitting low-level details. Indeed, it has been recently shown that for certain visual tasks [Aittala and Durand, 2018; Sridhar *et al.*, 2019; Liu *et al.*, 2019], significant gain can be achieved with intermediate information-sharing layers.

Here, we present a principled approach to learning sets of symmetric elements. First, we describe the symmetry group of these sets, and then fully characterize the space of linear layers that are equivariant to this group. Notably, this characterization implies that information between set elements should be shared in all layers. For example, Figure 1 illustrates a DSS layer for sets of images. DSS layers provide a unified framework that generalizes several previously-described architectures for a variety of data types. In particular, it directly generalizes DeepSets [Zaheer *et al.*, 2017]. Moreover, other recent works can also be viewed as special cases of our approach [Hartford *et al.*, 2018; Aittala and Durand, 2018; Sridhar *et al.*, 2019].

A potential concern with equivariant architectures is that restricting layers to be equivariant to some group of symmetries may reduce the expressive power of the model [Maron *et al.*, 2019; Morris *et al.*, 2018; Xu *et al.*, 2019]. We address this concern by proving two universal-approximation theorems for invariant and equivariant DSS networks. Simply put, these theorems state that if invariant (equivariant) networks for the elements of interest are universal, then the corresponding invariant (equivariant) DSS networks on sets of such elements are also universal. One important corollary of these results is that DSS networks are strictly more expressive than Siamese networks.

*The original version of this paper was presented at the International Conference on Machine Learning (ICML), 2020.

†Contact Author

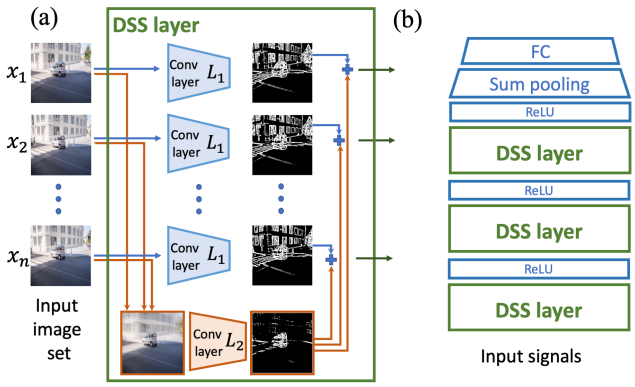

Figure 1: (a) A DSS layer for a set of images is composed of Siamese layer (blue) and an aggregation module (orange). The Siamese part is a convolutional layer (L_1) that is applied to each element independently. In the aggregation module, the *sum* of all images is processed by a different convolutional layer (L_2) and is added to the output of the Siamese part. (b) A simple DSS-based invariant network.

2 Preliminaries

2.1 Notation and Definitions

Let $x \in \mathbb{R}^\ell$ represent an input that adheres to a group of symmetries $G \leq S_\ell$, the symmetric group on ℓ elements. G captures those transformations that our task-of-interest is invariant (or equivariant) to. The action of G on $\mathbb{R}^\ell$ is defined by $(g \cdot x)_i = x_{g^{-1}(i)}$. For example, when inputs are images of size $h \times w$, we have $\ell = hw$ and G can be a group that applies cyclic translations, or left-right reflections to an image. A function is called G-equivariant if $f(g \cdot x) = g \cdot f(x)$ for all $g \in G$. Similarly, a function f is called G-invariant if $f(g \cdot x) = f(x)$ for all $g \in G$.

2.2 G-invariant and G-equivariant Networks

G-equivariant networks are a popular way to model G-equivariant functions. These networks are composed of several linear G-equivariant layers, interleaved with activation functions like ReLU, and have the following form:

$$f = L_k \circ \sigma \circ L_{k-1} \cdots \circ \sigma \circ L_1, \tag{1}$$

where $L_i : \mathbb{R}^{\ell \times d_i} \to \mathbb{R}^{\ell \times d_{i+1}}$ are linear G-equivariant layers, d_i are their feature dimensions and σ is a point-wise activation function. It is straightforward to show that this architecture results in a G-equivariant function. G-invariant networks are defined by adding an invariant layer on top of a G-equivariant function followed by a multilayer Perceptron.

2.3 Characterizing Equivariant Layers

The main building block of G-invariant/equivariant networks are linear G-invariant/equivariant layers. To implement these networks, one has to characterize the space of linear G-invariant/equivariant layers. For example, it is well known that for images with the group G of circular 2D translations, the space of linear G-equivariant layers is simply the space of all 2D convolutions operators [Puschel and Moura, 2008]. Unfortunately, such elegant characterizations are not available for most permutation groups. See [Wood and Shawe-Taylor, 1996; Ravanbakhsh *et al.*, 2017] for further details.

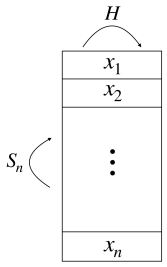

Figure 2: The input to a DSS layer is an $n \times d$ matrix, in which each row holds a d-dimensional element. $G = S_n \times H$ acts on it by applying a permutation to the columns and an element $h \in H$ to the rows.

2.4 Deep Sets

The current paper generalizes *DeepSets* [Zaheer *et al.*, 2017] and we summarize their main results for completeness. Let $\{x_1, \dots x_n\} \subset \mathbb{R}^d$ be a set, which we represent in arbitrary order as a matrix $X \in \mathbb{R}^{n \times d}$. [Zaheer *et al.*, 2017] characterized all S_n-equivariant linear layers for this case, which take the form:

$$L(X)_i = L_1(x_i) + L_2\left(\sum_{j \neq i}^{n} x_j\right), \tag{2}$$

where $L_1, L_2 : \mathbb{R}^d \to \mathbb{R}^d$ are general linear functions and the subscript represents the i-th row of the output. [Zaheer *et al.*, 2017; Qi *et al.*, 2017] established the universality of invariant networks that are composed of DeepSets Layers and [Segol and Lipman, 2019] extended this result to the equivariant case.

3 DSS Layers

Our main goal is to design deep models for sets of elements with non-trivial per-element symmetries. In this section, we first formulate the symmetry group G of such sets. The deep models we advocate are composed of linear G-equivariant layers (DSS layers); therefore, our next step is to find a simple and practical characterization of the space of these layers.

3.1 Sets with Symmetric Elements

Let $\{x_1, \dots x_n\} \subset \mathbb{R}^d$ be a set of elements with symmetry group $H \leq S_d$. We wish to characterize the space of linear maps $L : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$ that are equivariant to both the natural symmetries of the elements, represented by the elements of the group H, as well as to the order of the n elements, represented by S_n.

In our setup, H operates on all elements x_i in the same way. More formally, the symmetry group is defined by $G = S_n \times H$, where S_n is the symmetric group on n elements. This group operates on $X \in \mathbb{R}^{n \times d}$ by applying the permutation $q \in S_n$ to the first dimension and the same element $h \in H$ to the second dimension, namely $((q, h) \cdot X)_{ij} = X_{q^{-1}(i)h^{-1}(j)}$. Figure 2 illustrates this setup. Notably, this setup generalizes several popular learning setups: (1) DeepSets, where $H = \{I_d\}$ is the trivial group. (2) Tabular data [Hartford *et al.*, 2018], where $H = S_d$. (3) Sets

of images, where H is the group of cyclic translations [Aittala and Durand, 2018].

Considering the symmetry in the form of a direct product suggests implicitly that the elements $\{x_i\}$ are aligned, as they are in many popular tasks like image deblurring [Aittala and Durand, 2018] (also see figure 1). For other datasets and tasks, this assumption might not hold. In that case, it is natural to consider another setup, where the members of H applied to each element may differ[1]. Unfortunately, as we show in the full paper [Maron *et al.*, 2020], the corresponding equivariant layers are essentially Siamese layers. See also [Wang *et al.*, 2020].

3.2 Characterization of Equivariant Layers

This subsection provides a practical characterization of linear G-equivariant layers for $G = S_n \times H$. Our result generalizes DeepSets (equation 2) whose layers are tailored for $H = \{I_d\}$, by replacing the linear operators L_1, L_2 with linear H-equivariant operators. This result is summarized in the following theorem:

Theorem 1. *Any linear $G-$equivariant layer $L : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ is of the form*

$$L(X)_i = L_1^H(x_i) + L_2^H\left(\sum_{j\neq i}^{n} x_j\right),$$

where L_1^H, L_2^H are linear H-equivariant functions

Note that this is equivalent to the following formulations $L(X)_i = L_1^H(x_i) + L_2^H(\sum_{j=1}^n x_j) = L_1^H(x_i) + \sum_{j=1}^n L_2^H(x_j)$ due to linearity. Figure 1 illustrates Theorem 1 for sets of images. In this case, applying a DSS layer amounts to: (i) Applying the same convolutional layer L_1 to all images in the set (blue); (ii) Applying another convolutional layer L_2 to the sum of all images (orange); and (iii) summing the outputs of these two layers.

Relation to previous work. In the case of a set of images and translation equivariance, L_i^H are convolutions. In this setting, [Aittala and Durand, 2018; Sridhar *et al.*, 2019] have previously proposed using similar set-aggregation layers after convolutional blocks. This work provides a theoretical analysis of their setup and generalizes it all permutation groups.

4 A Universal Approximation Theorem

When restricting a network to be invariant (equivariant) to some group action, one may worry that these restrictions could reduce the network expressive power. We now show that networks that are constructed from DSS layers do not suffer from loss of expressivity. Specifically, we show that for any group H that induces a *universal* H-invariant (equivariant) network, its corresponding G-invariant (equivariant) network has high expressive power: we prove universal approximation for invariant and equivariant functions defined on any compact set with zero intersection with a specific low-dimensional set $\mathcal{E} \subset \mathbb{R}^{n\times d}$. The precise definition of $\mathcal{E}$ is given in the supplementary of the full paper [Maron *et al.*, 2020]. The following theorem summarizes these results:

[1]Here, the symmetry group is the wreath product of S_n and H.

Theorem 2. *Let $K \subset \mathbb{R}^{n\times d}$ be a compact domain such that $K = \cup_{g\in G} gK$ and $K \cap \mathcal{E} = \emptyset$. G-equivariant networks are universal approximators (in $\|\cdot\|_\infty$ sense) of continuous $\mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ G-equivariant functions on K if H-equivariant networks are universal.*

The invariant case is a direct corollary of Theorem 2.

5 Experiments

This section investigates the effectiveness of DSS layers in practice by comparing them to previously suggested architectures and different aggregation schemes. We use the experiments to answer two basic questions: **(1) *Early or late aggregation?*** Can *early aggregation* architectures like DSS and its variants improve learning compared to *Late aggregation* architectures, which fuse the set information at the end of the data processing pipeline? and **(2) *How to aggregate?*** What is the preferred early aggregation scheme?

Tasks. In the full paper, [Maron *et al.*, 2020], we evaluated DSS in a series of six experiments spanning a wide range of tasks: from classification ($\mathbb{R}^{n\times d} \to \mathbb{R}$), through selection ($\mathbb{R}^{n\times d} \to \mathbb{R}^n$) and burst image deblurring ($\mathbb{R}^{n\times d} \to \mathbb{R}^d$) to general equivariant tasks ($\mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$). The experiments also demonstrate the applicability of DSS to a range of data types, including point clouds, images and graphs.

Competing methods. We compare DSS to a narchitecture with a Siamese network followed by DeepSets (Siamese+DS). See full paper for more comparisons. We also compare several variants of our DSS layers: **(1) DSS(sum)**: our basic DSS layer from Theorem 1 **(2) DSS(max)**: DSS with max-aggregation instead of sum-aggregation **(3) DSS(Aittala)**: DSS with the aggregation proposed in [Aittala and Durand, 2018], namely, $L(x)_i \mapsto [L^H(x_i), \max_{j=1}^n L^H(x_j)]$ where [] denotes feature concatenation and L^H is a linear H-equivariant layer **(4) DSS(Sridhar)**: DSS layers with the aggregation proposed in [Sridhar *et al.*, 2019] ,i.e., $L(x)_i \mapsto L^H(x_i) - \frac{1}{n}\sum_{j=1}^n L^H(x_j)$.

Evaluation protocol. All models have roughly the same number of parameters for each particular task. In all experiments, we report the mean and standard deviation over five random initializations. Experiments were conducted using NVIDIA DGX with V100 GPUs.

5.1 Selection Tasks

In a selection task, we are given a set and wish to choose one element from the set that obeys a predefined property. Formally, each task is modeled as a G-equivariant function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^n$, where the output vector represents the probability of selecting each element. The architecture comprises three convolutional blocks employing Siamese or DSS variants, followed by a DeepSets block. We note that the *Siamese+DS* model was suggested for similar selection tasks in [Zaheer *et al.*, 2017].

Frame selection in images and shapes. The first selection task is to find a particular frame within an unordered set of frames extracted from a video/shape sequence. For videos,

Dataset	Data type	Late Aggregation	Early Aggregation				Random choice
		Siamese+DS	DSS (sum)	DSS (max)	DSS (Sridhar)	DSS (Aittala)	
UCF101	Images	36.41% ± 1.43	76.6% ± 1.51	76.39% ± 1.01	60.15% ± 0.76	**77.96%** ± 1.69	12.5%
Dynamic Faust	point clouds	22.26% ± 0.64	42.45% ± 1.32	28.71% ± 0.64	**54.26%** ± 1.66	26.43% ± 3.92	14.28%
Dynamic Faust	Graphs	26.53% ± 1.99	44.24% ± 1.28	30.54% ± 1.27	**53.16%** ± 1.47	26.66% ± 4.25	14.28%

Table 1: Frame selection tasks for images, point clouds and graphs. Numbers represent average classification accuracy.

Noise type and strength	Late Aggregation	Early Aggregation				Random choice
	Siamese+DS	DSS (sum)	DSS (max)	DSS (Sridahr)	DSS (Aittala)	
Gaussian $\sigma = 10$	77.2% ± 0.37	**78.48%** ± 0.48	77.99% ± 1.1	76.8% ± 0.25	78.34% ± 0.49	5%
Gaussian $\sigma = 30$	65.89% ± 0.66	**68.35%** ± 0.55	67.85% ± 0.40	61.52% ± 0.54	66.89% ± 0.58	5%
Gaussian $\sigma = 50$	59.24% ± 0.51	**62.6%** ± 0.45	61.59% ± 1.00	55.25% ± 0.40	62.02% ± 1.03	5%
Occlusion 10%	82.15% ± 0.45	83.13%± 1.00	**83.27** ± 0.51	83.21% ± 0.338	83.19% ± 0.67	5%
Occlusion 30%	77.47% ± 0.37	78% ± 0.89	78.69% ± 0.32	**78.71%** ± 0.26	78.27% ± 0.67	5%
Occlusion 50%	76.2% ± 0.82	**77.29%** ± 0.40	76.64% ± 0.45	77.04% ± 0.75	77.03% ± 0.58	5%

Table 2: Highest-quality image selection. Values indicate the mean accuracy.

we used the UCF101 dataset [Soomro *et al.*, 2012]. Each set contains $n = 8$ frames generated by randomly drawing a video, a starting position, and frame order. The task is to select the "first" frame, namely, the one that appeared earliest in the video. Table 1 details the accuracy of all compared methods in this task, showing that *DSS(sum)* and *DSS(Aittala)* outperform *Siamese+DS* and *DSS(Sridhar)* by a large margin.

In a second frame selection task, we demonstrate that DSS can handle multiple data types. Specifically, we showcase how DSS operates on point clouds and graphs. Given a short sequence of 3D human shapes performing various activities, the task is to identify which frame was the center frame in the original non-shuffled sequence. These human shapes are represented as point clouds in the first experiment and as graphs (point clouds + connectivity) in the second experiment.

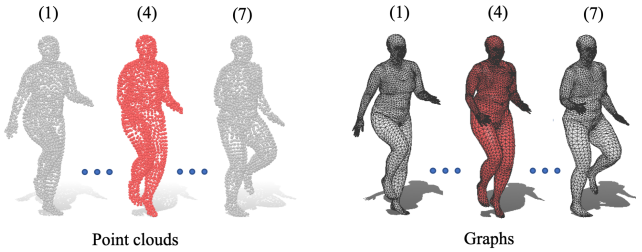

Figure 3: Shape-selection task on human shape sequences. Shapes are represented as graphs or as point clouds. The task is to select the central frame (red). Numbers indicate frame order.

To generate the data, we cropped 7-frame-long sequences from the Dynamic Faust dataset [Bogo *et al.*, 2017] in which the shapes are given as triangular meshes. In [Bogo *et al.*, 2017] the points of each mesh are ordered consistently, providing point-to-point correspondence across frames. When this correspondence is not available, a shape matching algorithm like [Litany *et al.*, 2017; Maron and Lipman, 2018] can be used as preprocessing. See Figure 3 for an illustration of this task. Results are summarized in Table 1, comparing DSS variants to a late-aggregation baseline (Siamese +DS) and to random choice.

Highest quality image selection. Given a set of $n = 20$ degraded images of the same scene, the task is to select the highest-quality image. We generate data for this task from the Places dataset [Zhou *et al.*, 2017], by adding noise and Gaussian blur to each image. The target image is defined to be the image that is the most similar in L_1 norm sense to the original image. Notably, DSS consistently improves over Siamese+DS with a margin of 1% to 3%. See Table 2.

5.2 Summary of Experiments

The above experiments demonstrate that applying early aggregation using DSS layers improves learning in various tasks and data types, compared with earlier architectures like *Siamese+DS*. This improvement can be attributed to the provably higher expressive power of DSS networks. More specifically, the basic DSS layer, *DSS(sum)*, performs well on all tasks, and *DSS(Aittala)* has also yielded strong results. *DSS(Sridhar)* performs well on some tasks but fails on others. See the full paper for additional experiments: signal classification, image deblurring, and multi-view reconstruction.

6 Conclusion

In this short paper, we have summarized the main results of [Maron *et al.*, 2020]. Specifically, we presented a principled approach for designing deep networks for sets of elements with symmetries. We characterized the space of equivariant maps for such sets, analyzed its expressive power, exemplified its benefits over standard set learning approaches over various tasks and data types, and shown that our approach generalizes several successful previous works. Using our framework as it is, one implicitly assumes that the set elements are aligned. A worthwhile direction for future work is to suggest approaches to circumvent this limitation.

Acknowledgments

This research was supported by an Israel science foundation grant 737/18. We thank Srinath Sridhar and Davis Rempe for valuable discussions.

References

[Aittala and Durand, 2018] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 731–747, 2018.

[Bogo *et al.*, 2017] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Bromley *et al.*, 1994] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[Hartford *et al.*, 2018] Jason S. Hartford, Devon R. Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *ICML*, 2018.

[Kalogerakis *et al.*, 2017] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3779–3788, 2017.

[Litany *et al.*, 2017] Or Litany, Tal Remez, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5659–5667, 2017.

[Liu *et al.*, 2019] Xiaofeng Liu, Zhenhua Guo, Site Li, Lingsheng Kong, Ping Jia, Jane You, and BVK Kumar. Permutation-invariant feature restructuring for correlation-aware image set-based recognition. *arXiv preprint arXiv:1908.01174*, 2019.

[Maron and Lipman, 2018] Haggai Maron and Yaron Lipman. (probably) concave graph matching. In *Advances in Neural Information Processing Systems*, pages 408–418, 2018.

[Maron *et al.*, 2019] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International conference on machine learning*, 2019.

[Maron *et al.*, 2020] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR, 2020.

[Morris *et al.*, 2018] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *arXiv preprint arXiv:1810.02244*, 2018.

[Puschel and Moura, 2008] Markus Puschel and José MF Moura. Algebraic signal processing theory: Foundation and 1-d time. *IEEE Transactions on Signal Processing*, 56(8):3572–3585, 2008.

[Qi *et al.*, 2017] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[Ravanbakhsh *et al.*, 2017] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. *arXiv preprint arXiv:1702.08389*, 2017.

[Segol and Lipman, 2019] Nimrod Segol and Yaron Lipman. On universal equivariant set networks. *arXiv preprint arXiv:1910.02421*, 2019.

[Soomro *et al.*, 2012] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[Sridhar *et al.*, 2019] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J Guibas. Multiview aggregation for learning category-specific shape reconstruction. *arXiv preprint arXiv:1907.01085*, 2019.

[Su *et al.*, 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[Wang *et al.*, 2020] Renhao Wang, Marjan Albooyeh, and Siamak Ravanbakhsh. Equivariant maps for hierarchical structures. *arXiv preprint arXiv:2006.03627*, 2020.

[Wood and Shawe-Taylor, 1996] Jeffrey Wood and John Shawe-Taylor. Representation theory and invariant neural networks. *Discrete applied mathematics*, 69(1-2):33–60, 1996.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[Zaheer *et al.*, 2017] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[Zhou *et al.*, 2017] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.