3D Molecular Design with Covariant Neural Networks

Gregor N. C. Simm, Robert Pinsler, Gábor Csányi & José Miguel Hernández-Lobato
Department of Engineering, University of Cambridge, Cambridge, UK
{gncs2,rp586,gc121,jmh233}@cam.ac.uk

Abstract

Automating molecular design using deep reinforcement learning (RL) has the potential to greatly accelerate the search for novel materials. Despite recent progress on leveraging graph representations to design molecules, such methods are fundamentally limited by the lack of three-dimensional (3D) information. In light of this, we propose a novel actor-critic architecture for 3D molecular design that can generate highly symmetric molecular structures. This is achieved by exploiting the symmetries of the design process through a rotationally covariant state-action representation based on a spherical harmonics series expansion. We demonstrate that building in such symmetries allows our agent to generate molecules that were unattainable with previous approaches.

1 Introduction

The search for molecular structures with desirable properties is a challenging task with important applications in *de novo* drug design and materials discovery [1]. There exist a plethora of machine learning approaches to accelerate this search, including generative models based on variational autoencoders (VAEs) [2], recurrent neural networks (RNNs) [3], and generative adversarial networks (GANs) [4]. However, the reliance on a sufficiently large dataset for exploring unknown regions of chemical space is a severe limitation of such supervised models. Recent RL-based methods [5, 6, 7, 8, 9, 10, 11, 12, 13] mitigate the need for an existing dataset of molecules as they only require access to a reward function.

Most approaches rely on graph representations of molecules, where atoms and bonds are represented by nodes and edges, respectively. This is a strongly simplified model designed for the description of single organic molecules. It is unsuitable for encoding metals and molecular clusters as it lacks information about the relative position of atoms in 3D space. Further, geometric constraints on the design process cannot be included, e.g. those given by the active site of an enzyme. A more general representation closer to the physical system is one in which a molecule is described by its atoms' positions in Cartesian coordinates. In light of this, Gebauer et al. [14, 15] proposed a supervised generative model for sequentially placing atoms on a 3D grid. Similar to other supervised approaches, one further requires a dataset that covers the particular class of molecules to be generated. Hammer and coworkers [12, 16] employed a Deep Q-Network [17] to build planar compounds and crystalline surfaces by placing atoms on a grid.

In this work, we develop a novel RL approach for designing molecules in Cartesian coordinates that explicitly encodes the symmetries of the molecular design process. The agent builds molecules by consecutively placing atoms such that if the generated structure is rotated or translated, the agent's action is rotated and translated accordingly (see Fig. 1 (a)). We achieve this through a rotationally *covariant* state representation based on spherical harmonics, which we integrate into a novel actor-critic network architecture with an auto-regressive policy that maintains the desired covari-

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

ance. Building in this inductive bias enables us to generate highly symmetric molecular structures that were unattainable with previous approaches.

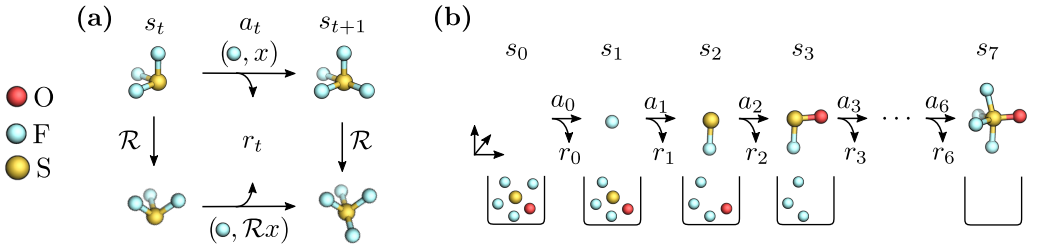

Figure 1: **(a)** Illustration of a rotation-covariant state-action representation. If the structure is rotated by $\mathcal{R}$, the position x of the action transforms accordingly. **(b)** Rollout with bag $\mathcal{B}_0 = \mathrm{SOF}_4$. The agent builds a molecule by repeatedly taking atoms from the bag and placing them onto the 3D canvas. Bonds connecting atoms are only for illustration and not part of the MDP.

2 Background

Following Simm et al. [13], we design molecules by iteratively picking atoms from a *bag* and positioning them on a 3D *canvas*. Such a sequential decision-making problem is described by an MDP where the state $s_t = (\mathcal{C}_t, \mathcal{B}_t)$ comprises both the canvas $\mathcal{C}_t$ and the bag $\mathcal{B}_t$. The canvas $\mathcal{C}_t = \mathcal{C}_0 \cup \{(e_i, x_i)\}_{i=0}^{t-1}$ is a set of atoms with chemical element $e_i \in \{\mathrm{H}, \mathrm{C}, \mathrm{N}, \mathrm{O}, \dots\}$ and position $x_i \in \mathbb{R}^3$ placed up to time $t-1$, where $\mathcal{C}_0$ can either be empty or contain a set of initially placed atoms. The number of atoms on the canvas is denoted by $|\mathcal{C}_t|$. The bag $\mathcal{B}_t = \{(e, m(e))\}$ is a multi-set of atoms yet to be placed, where $m(e)$ is the multiplicity of the element e. Each action $a_t = (e_t, x_t)$ consists of the element $e_t \in \mathcal{B}_t$ and position $x_t \in \mathbb{R}^3$ of the next atom to be added to the canvas. Placing an atom through action a_t in state s_t is modeled by a deterministic transition function $\mathcal{T}(s_t, a_t)$ that yields the next state $s_{t+1} = (\mathcal{C}_{t+1}, \mathcal{B}_{t+1})$ with $\mathcal{B}_{t+1} = \mathcal{B}_t \backslash e_t$. The reward function $r(s_t, a_t) = -\Delta E(s_t, a_t)$ is given by the negative energy difference between the resulting structure described by $\mathcal{C}_{t+1}$, and the sum of energies of the current structure $\mathcal{C}_t$ and a new atom of element e_t placed at the origin, i.e. $\Delta E(s_t, a_t) = E(\mathcal{C}_{t+1}) - [E(\mathcal{C}_t) + E(\{(e, \mathbf{0})\})]$. Intuitively, the reward encourages the agent to build stable, low-energy structures. An example of a rollout is shown in Fig. 1 (b). At the beginning of the episode, the agent observes the initial state $(\mathcal{C}_0, \mathcal{B}_0) \sim \mu_0(s_0)$, e.g. $\mathcal{C}_0 = \emptyset$ and $\mathcal{B}_0 = \mathrm{SOF}_4$. The agent then iteratively constructs a molecule by placing atoms from the bag onto the canvas until the bag is empty.[1]

3 Covariant Policy for Molecular Design

A function $f : \mathcal{X} \mapsto \mathcal{Y}$ is *invariant* under a transformation operator $T_g : \mathcal{X} \mapsto \mathcal{X}$ if $f(T_g[x]) = f(x)$ for all $x \in \mathcal{X}, g \in G$, where G is a mathematical group. In contrast, f is *covariant* with respect to T_g if there exists an operator $T'_g : \mathcal{Y} \mapsto \mathcal{Y}$ such that $f(T_g[x]) = T'_g[f(x)]$. We require a policy $\pi(a|s)$ with actions $a = (e, x)$ that is covariant under translation and rotation with respect to the position x, i.e., x should rotate (or translate) accordingly if the atoms on the canvas $\mathcal{C}$ are rotated (or translated). Since learning such a policy is difficult when working directly in global Cartesian coordinates, we instead follow Simm et al. [13] and use an action representation that is local with respect to an already placed *focal atom*. If the next atom is placed relative to the focal atom, covariance under translation of x is automatically achieved and only the rotational covariance remains to be dealt with.

As shown in Fig. 2, we model the action a through a sequence of sub-actions: (1) the index[2] $f \in \{1, \dots, |\mathcal{C}|\}$ of the focal atom around which the next atom is placed, (2) the element $e \in \{1, \dots, N_e\}$ of the next atom from the set of available elements, (3) a distance $d \in \mathbb{R}_+$ between the focal atom and the next atom, and (4) the orientation $\tilde{x} = (\vartheta, \varphi) \in \mathbb{S}^2$ of the atom on a unit sphere around the

[1] Hereafter, we drop the time index when it is clear from the context.

[2] If the canvas $\mathcal{C}_0$ is empty, the agent selects an element $e_0 \in \mathcal{B}_0$ and places it at the origin, i.e. $a_0 = (e_0, \mathbf{0})$.

focal atom. Denoting x_f as the position of the focal atom, we obtain action $a = (e, x)$ by mapping the local coordinates $(\tilde{x}, d, f)$ to global coordinates $x = x_f + d \cdot \tilde{x}$, where x is now covariant under translation *and* rotation. We choose these sub-actions using the following auto-regressive policy:

$$\pi(a|s) = \pi(\tilde{x}, d, e, f|s) = p(\tilde{x}|d, e, f, s)\, p(d|e, f, s)\, p(e|f, s)\, p(f|s). \tag{1}$$

Next, we present a novel actor-critic [18] neural network architecture that implements this policy. The parameters are learned with PPO [19].

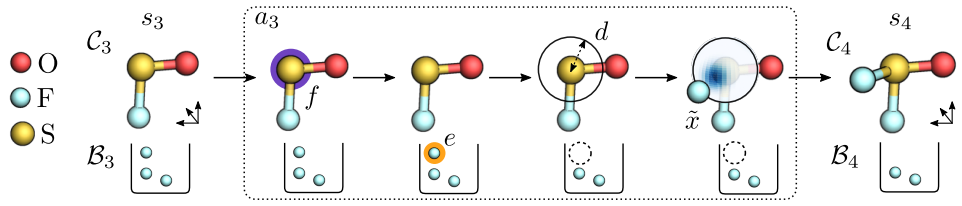

Figure 2: Action representation of the auto-regressive policy. The agent chooses focal atom f, element e, distance d, and orientation $\tilde{x}$. We then map back to global coordinates x to obtain action $a = (e, x)$. Bonds between atoms are only for illustration.

3.1 State Embedding

The state embedding network transforms canvas $\mathcal{C}$ and bag $\mathcal{B}$ to obtain a rotationally covariant and translationally invariant representation. For that, we employ CORMORANT [20], a neural network architecture for predicting properties of chemical systems that works entirely in Fourier space. A key feature is that each neuron is covariant to rotation but invariant to translation; further, each neuron explicitly corresponds to a subset of atoms in the molecule.

We concatenate a vectorized representation of the bag with each atom on the canvas and feed it into CORMORANT, i.e. $s^{\text{cov}} \leftarrow \text{CORMORANT}(\mathcal{C}, \mathcal{B})$, where $s^{\text{cov}} = \{s^{\text{cov}}_\ell\}_{\ell=0}^{L_{\max}}$, $s^{\text{cov}}_\ell \in \mathbb{C}^{|\mathcal{C}| \times \tau \times (2\ell+1)}$, and τ is the number of channels. For the sake of exposition, we assume a single channel for each element in the bag, i.e. $\tau = N_e$; in practice, we use up to four channels per element.

However, not every sub-action in Eq. (1) should transform equally under rotation and translation. While the orientation $\tilde{x}$ needs to be *covariant* under rotation, the choice of focal atom f, element e, and distance d has to be *invariant* to rotation and translation. For these sub-actions, we additionally require an invariant state representation. To obtain such a representation $s^{\text{inv}} \in \mathbb{R}^{|\mathcal{C}| \times k}$, we employ a combination of transformations from Anderson et al. [20], which we collectively denote as $\mathcal{T}_{\text{inv}}$.

3.2 Actor-Critic

Focal Atom and Element The distribution $p(f|s)$ over the focal atom f is modeled as categorical, $f \sim \text{Cat}(f; h_f)$, where h_f are the logits for each atom in $\mathcal{C}$ predicted by a multi-layer perceptron (MLP). Likewise, the distribution over the element e is given by $p(e|f, s) = \text{Cat}(e; h_e)$ with $h_e = \text{MLP}_e(s^{\text{inv}}_f)$, where s^{inv}_f is the invariant representation for the focal atom.

Distance We select the channel τ_e corresponding to element e from s^{cov}_f to obtain $s^{\text{cov}}_{f,e} := \{s^{\text{cov}}_{\ell_{f,e}}\}_{\ell=0,\dots,L_{\max}}$ and $s^{\text{inv}}_{f,e} \leftarrow \mathcal{T}_{\text{inv}}(s^{\text{cov}}_{f,e})$. Then we model the distribution over the distance d between the focal atom and the next atom to be placed as a mixture of M Gaussians, $p(d|e, f, s) = \sum_{m=1}^{M} \pi_m\, \mathcal{N}(\mu_m, \sigma_m^2)$, where π_m is the mixing coefficient of the m-th Gaussian $\mathcal{N}(\mu_m, \sigma_m^2)$. The mixing coefficients and the means are predicted by a mixture density network (MDN) [21], i.e. $\{\pi_m, \mu_m\}_{m=1}^{M} = \text{MDN}(s^{\text{inv}}_{f,e})$. The standard deviations $\{\sigma_m\}_{m=1}^{M}$ are global parameters.

Combining Invariant and Covariant Features We condition $s^{\text{cov}}_{f,e}$ on distance d through a non-linear and learnable transformation that preserves rotational covariance. We obtain a rotationally covariant representation $\hat{r} := \{\hat{r}_\ell\}_{\ell=0,\dots,L_{\max}}$ conditioned on all previous sub-actions as follows:

$$\hat{r}_\ell = \left[s^{\text{cov}}_{\ell_{f,e}} \oplus d \cdot s^{\text{cov}}_{\ell_{f,e}} \oplus (d \cdot s^{\text{cov}}_{f,e} \otimes_{\text{cg}} d \cdot s^{\text{cov}}_{f,e})_\ell \right] \cdot W_\ell \quad \forall \ell, \tag{2}$$

where $\oplus$ denotes the appropriate concatenation of matrices, $\otimes_{\text{cg}}$ is the channel-wise CG product [20], and W_ℓ is a learnable complex-valued matrix.

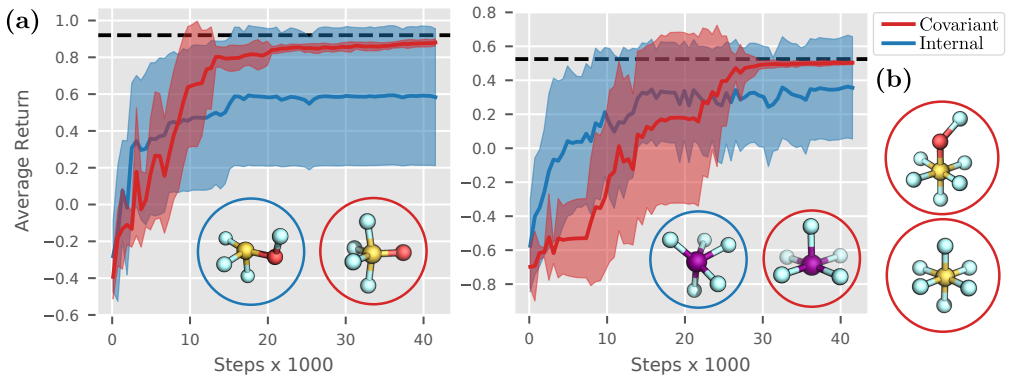

Figure 3: **(a)** Average offline performance on the bags SOF_4 (left) and IF_5 (right) across 10 seeds. In the lower right, molecular structures generated by the agents are shown. Dashed lines denote the optimal return for each experiment. Error bars show two standard deviations. **(b)** Further molecular structures generated by COVARIANT, namely SOF_6 and SF_6.

Orientation Next, we utilize $\hat{r}$ to obtain a rotationally covariant spherical distribution for the orientation $\tilde{x}$ based on a spherical harmonics series expansion. Taking inspiration from commonly used distributions [22], we propose to use the following expression:

$$p(\tilde{x}|d, e, f, s) = \frac{1}{Z} \exp\left(-\beta \left| \frac{1}{\sqrt{k}} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \hat{r}_\ell^m Y_\ell^m(\tilde{x}) \right|^2\right), \tag{3}$$

where $\beta \in \mathbb{R}$ is a scaling parameter, $1/\sqrt{k}$ with $k = \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} |\hat{r}_\ell^m|^2$ regularizes the distribution so that it does not approach a delta function, and Z is estimated via Lebedev quadrature [23, 24]. We sample from the distribution in Eq. (3) using rejection sampling [25] with a uniform proposal distribution $q(\tilde{x}) = (4\pi)^{-1}$. In contrast to more commonly used parametric distributions (e.g. von Mises-Fisher), this formulation allows to model multi-modalities.

Critic To compute a value V, we apply a permutation-invariant set encoding [26] over the atoms in s^{inv}, i.e. $V(s) = \text{MLP}_\rho\left(\sum_{i=1}^{|\mathcal{C}|} \text{MLP}_\phi(s_i^{\text{inv}})\right)$.

4 Experiments

We evaluate the ability to build stable molecules featuring high symmetry and coordination numbers (e.g. trigonal bipyramidal, square pyramidal, and octahedral) with bags SOF_4, IF_5, SOF_6, and SF_6. We compare our approach (COVARIANT) with the RL agent proposed by Simm et al. [13], which iteratively builds molecules on a 3D canvas by working in internal coordinates (INTERNAL). In particular, INTERNAL chooses a distance, angle, and dihedral angle with respect to already placed *reference atoms* which are determined based on their distance to the focal atom. As shown in Fig. 3 (a), COVARIANT can solve the task for SOF_4 and IF_5 within 30 000 to 40 000 steps, whereas INTERNAL fails to build low-energy configurations. That is because in symmetric states, there can be multiple, indistinguishable sets of reference atoms which lead to different actions. Further results in Fig. 3 (b) for SOF_6 and SF_6 show that COVARIANT can build such structures.

5 Conclusion

We proposed a novel covariant actor-critic architecture based on spherical harmonics for designing highly symmetric molecules in 3D. In future work, we aim to employ more accurate quantum-chemical methods (e.g., density functional theory) and improve sample-efficiency. Finally, we aim to explore reward functions specifically tailored towards drug design.

References

[1] Petra Schneider, W. Patrick Walters, Alleyn T. Plowright, Norman Sieroka, Jennifer Listgarten, Robert A. Goodnow, Jasmin Fisher, Johanna M. Jansen, José S. Duca, Thomas S. Rush, Matthias Zentgraf, John Edward Hill, Elizabeth Krutoholow, Matthias Kohler, Jeff Blaney, Kimito Funatsu, Chris Luebkemann, and Gisbert Schneider. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discovery*, pages 1–12, 2019.

[2] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.*, 4(2):268–276, 2018.

[3] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.*, 4(1):120–131, 2018.

[4] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[5] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.*, 9(1):48, 2017.

[6] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv preprint arXiv:1705.10843*, 2018.

[7] Evgeny Putin, Arip Asadulaev, Yan Ivanenkov, Vladimir Aladinskiy, Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik, and Alex Zhavoronkov. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J. Chem. Inf. Model.*, 58(6):1194–1204, 2018.

[8] Daniel Neil, Marwin Segler, Laura Guasch, Mohamed Ahmed, Dean Plumbley, Matthew Sellwood, and Nathan Brown. Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. *OpenReview*, 2018. URL `https://openreview.net/forum?id=HkcTe-bR-`.

[9] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Sci. Adv.*, 4(7):eaap7885, 2018.

[10] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. In *Advances in Neural Information Processing Systems*, pages 6410–6421, 2018.

[11] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of Molecules via Deep Reinforcement Learning. *Sci. Rep.*, 9(1):1–10, 2019.

[12] Mathias S. Jørgensen, Henrik L. Mortensen, Søren A. Meldgaard, Esben L. Kolsbjerg, Thomas L. Jacobsen, Knud H. Sørensen, and Bjørk Hammer. Atomistic structure learning. *J. Chem. Phys.*, 151(5):054111, 2019.

[13] Gregor NC Simm, Robert Pinsler, and José Miguel Hernández-Lobato. Reinforcement learning for molecular design guided by quantum mechanics. In *International Conference on Machine Learning*, 2020. URL `http://arxiv.org/abs/2002.07717`.

[14] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Generating equilibrium molecules with deep neural networks. *arXiv preprint arXiv:1810.11347*, 2018.

[15] Niklas W. A. Gebauer, Michael Gastegger, and Kristof T. Schütt. Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules. In *Advances in Neural Information Processing Systems*, pages 7564–7576, 2019.

[16] Søren A. Meldgaard, Henrik L. Mortensen, Mathias S. Jørgensen, and Bjørk Hammer. Structure prediction of surface reconstructions by deep reinforcement learning. *J. Phys.: Condens. Matter*, 32(40):404005, 2020.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[18] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 2000.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[20] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant Molecular Neural Networks. In *Advances in Neural Information Processing Systems 32*, pages 14537–14546. Curran Associates, Inc., 2019.

[21] Christopher M Bishop. Mixture density networks. Technical report, Aston University, 1994.

[22] S. Rao Jammalamadaka and György H. Terdik. Harmonic analysis and distribution-free inference for spherical distributions. *J. Multivar. Anal*, 171:436–451, 2019.

[23] V. I. Lebedev. Values of the nodes and weights of ninth to seventeenth order gauss-markov quadrature formulae invariant under the octahedron group with inversion. *Zh. Vychisl. Mat. Mat. Fiz.*, 15(1):44–51, 1975.

[24] V. I. Lebedev. Spherical quadrature formulas exact to orders 25–29. *Sibirsk. Mat. Zh.*, 18(1): 99–107, 1977.

[25] Christopher M. Bishop. Rejection sampling. In *Pattern Recognition and Machine Learning*, Information Science and Statistics, pages 528–530. Springer, New York, 2009. ISBN 978-0-387-31073-2.

[26] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, 2017.