Lecture 3: Equivariant Convolutional Neural Networks on Homogeneous Spaces

Geometry and Topology in Machine Learning Seminar

June 16th, 2025

Recall: Representations and Bundles

1. Let G be a group and V be a vector space.
   - A G-representation is a group action of G on V where the actions are all linear maps.
   - Equivalently, this is the data of a map $\rho : G \to \mathrm{GL}(V)$ that respects the group structure on both sides[1].
2. A principal G-bundle is a fiber bundle $\phi : X \to B$ where the fibers are G-torsors, i.e. spaces where G acts freely and transitively on.
3. Let $\rho : G \to \mathrm{GL}(V)$ be an G-representation. The associated vector bundle of $\phi : X \to B$ with respect to ρ is

$$\phi' : E := X \times_\rho V := X \times V / \sim \to B$$

where $(x \cdot g, y) \sim (x, \rho(g) \cdot y)$. Each fiber over $b \in B$ is a copy of V, equipped with the action of G from ρ. E is the vector bundle associated to ϕ via ρ.

[1] i.e. a group homomorphism $\rho(gh) = \rho(g)\rho(h)$

Recall: Sections and Mackey Functions

4. Given an associated vector bundle $\phi' : E \to B$, a section[2] is a map $s : B \to E$ such that $\phi' \circ s = id_B$.
5. Let $H \leq G$ and V be an H-representation ρ, a Mackey function is a map

$$f : G \to V, f(gh) = \rho(h^{-1})f(g), \quad \forall g \in G, h \in H.$$

In the last lecture, we focused on the case where the base space B is a homogeneous space, meaning we can write $B = G/H$.

- The quotient map $G \to G/H$ is a principal H-bundle.
- Given an associated vector bundle $E \to G/H$ from an H-representation ρ, there is an isomorphism between

$\Gamma(E) = \{\text{sections of } E\}$ and $\mathcal{I}_G = \{\text{Mackey functions for } (H, \rho)\}$

[2] Also known as a field, or a stack of features

Recall: Induced Representations

6. Induced representations describe how features transform in a globally consistent vector bundle with G-actions.

Definition

Let $H \leq G$ and $\rho : H \to \mathrm{GL}(V)$ be an H-representation. The induced representation $\pi = \mathrm{Ind}_H^G \rho$ is the data:

1. The vector space of Mackey functions $\mathcal{I}_G$
2. An action of G on $\mathcal{I}_G$ by

$$g \cdot f(k) \coloneqq f(g^{-1}k)$$

Note in the case of $G \to G/H$, $\mathcal{I}_G$ is isomorphic to $\Gamma(E)$, so the induced representation describes how $\Gamma(E)$ transforms!

Outline

Let us talk more about ML in this lecture!

In the last lecture, we have seen the analogy of feature spaces and an example of spherical CNNs. Today, we will focus on layers between feature spaces in our set-up. Specifically, we should address the following questions:

1. Doing convolutions seems to require an integral - how do we integrate on groups?
2. Even if we can integrate on nice groups, what does convolution mean in this context?
3. How do we design convolutional neural networks from this?
4. How does convolution relate to equivariance?

Integration in $\mathbb{R}^1$ and $\mathbb{R}^n$

For example, let us look at the group $(\mathbb{R}^1, +)$ (the setting for 1D CNNs) where the operation is addition. Integration over $\mathbb{R}^1$ is fundamentally based on:

$$\int_{[a,b]} 1 dx = \text{length}([a,b]).$$

Observation: The length on $\mathbb{R}^1$ satisfies the following:

1. For any interval $[a, b]$, if we shift the entire interval by $t \in \mathbb{R}$, the length does not change.
2. The length of any non-empty open interval is not zero.
3. The length of any closed and bounded interval is finite.

If we move to the group $(\mathbb{R}^n, +)$ and length to n-dimensional volume, analogous statements to above would hold.

Going More Generally

1. For any interval $[a, b]$, if we shift the entire interval by $t \in \mathbb{R}$, the length does not change.
2. The length of any non-empty open interval is not zero.
3. The length of any closed and bounded interval is finite.

The general properties they should correspond to a group G are

1. Length $\to$ a $\mathbb{R}_{\geq 0}$-valued function μ on "nice" subsets of G (specifically a measure).
2. $t + [a, b]$ is an example of a left coset of $[a, b]$ with respect to $t \in \mathbb{R}$. Thus, we can more generally consider:

$$gS = \{gs \in G \mid s \in S\}, g \in G, S \subset G \text{ and } \mu(gS) = \mu(S).$$

3. Open intervals $\to$ Open sets.
4. Closed and bounded intervals $\to$ Compact sets.

Integration on Groups

Let G be a nice group[3], there is a way to define integration on G as follows:

Theorem

G admits a left Haar measure, which is a measure $\mu : F(G)^4 \subset \{\text{Subsets of } G\} \to \mathbb{R}$ such that:

1. μ is left-translation invariant, ie. $\mu(gS) = \mu(S)$ for all $g \in G$ and $S \in F(G)$. Here
2. $\mu(U) > 0$ for all non-empty open subset of G.
3. $\mu(K) < \infty$ for all compact subsets of G.

Note that left Haar measure is unique up to scaling.

[3] locally compact and Hausdorff, this is satisfied for all examples of groups we have seen so far

[4] This is the Borel σ-algebra, but you can just think of this as reasonable subsets.

Integration on Groups (Continued)

There is a similar way to define a right Haar measure with right cosets. We say G is unimodular if its left and right measures agree after scaling.

Definition

For unimodular nice groups G, we call μ the Haar measure of G.

1. If G is compact (ex. $\mathrm{SO}(n)$), then it is unimodular.
2. If G is abelian, then it is also unimodular.
3. (For Algebraic Enthusiasts): If more generally the abelianization of G is finite, then it is also unimodular.
4. (For Lie theory enthusiasts): For a Lie group G such that $\det(\mathrm{Ad}_g) = 1$ for all $g \in G$, G is unimodular.

Examples of Integration on Groups

Given a Haar measure μ on G, we can use it to define an integration $\int_G \bullet d\mu$ on G such that

$$\int_G 1_S d\mu := \mu(S).$$

1. If G is $(\mathbb{R}^n, +)$, this is just the usual (Lebesgue) integration from calculus.
2. If G is S^1 (group operation being rotation), for a function $f : S^1 \to \mathbb{R}$, this is the same as viewing f as a function $f' : [0,1] \to \mathbb{R}$ with $f(0) = f(1)$ and
$$\int_{S^1} f d\mu = \int_0^1 f' dx.$$
3. If G is a finite group (or more generally a discrete group[5]), then the integration is the same as discrete summation.

[5] Think $\mathrm{SL}_2(\mathbb{Z})$ for example

Convolution on Groups

In 1D functions, for two functions $f, g : \mathbb{R} \to \mathbb{R}$, convolution is given by

$$f * g(t) = \int_{\mathbb{R}} f(\tau) g(t - \tau) d\tau = \int_{\mathbb{R}} f(\tau) g((-\tau) + t) d\tau.$$

Another way to see this however is that the term $(-\tau) + t$ is really an example of

$$g^{-1} g' \in G \text{ for some elements } g, g' \in G.$$

Convolution on Groups

More generally, let V_1, V_2 be vector spaces (ie. codomain of features in CNNs).

Definition

Given a function $f : G \to V_1$ and $\kappa : G \to \mathrm{Hom}(V_1, V_2)$, the convolution of f and κ is the function

$$f \star \kappa : G \to V_2, (f \star \kappa)(g) := \int_G \kappa(g^{-1}g')f(g')d\mu(g')$$

Here the integration is taken with respect to the variable g'.

Here κ is often called the (one-argument) G-convolution kernel.

G-Convolutional Neural Network

Let G be a group.

Definition

A G-**Convolutional Neural Network Layer** (G-CNN Layer) is a layer $L_i : \mathcal{F}_i \to \mathcal{F}_{i+1}$ in a neural network where

1. $\mathcal{F}_i$ is the feature space of G with respect to some $(H_i \leq G, \rho_i : H_i \to \mathrm{GL}(V_i))$. In other words, $\mathcal{F}_i$ is the space of sections $\Gamma(E_i)$ of the associated vector bundle $E_i = G \times_{\rho_i} V_i \to G/H_i$. $\mathcal{F}_{i+1}$ is defined similarly.
2. The map $L_i : \Gamma(E_i) \to \Gamma(E_{i+1})$ is a convolution with respect to some κ_i where κ_i is to be optimized/learned.

Here we recall that $\Gamma(E_i)$ (resp. $\Gamma(E_{i+1})$) is isomorphic to the collection of Mackey functions $f : G \to V_i$ (resp. $f : G \to V_{i+1}$), so the convolution here makes sense.

Outline

G-Equivariant Neural Network

Let G be a group.

Definition

A G-**Equivariant Neural Network Layer** is a layer $L_i : \mathcal{F}_i \to \mathcal{F}_{i+1}$ in a neural network where

1. $\mathcal{F}_i$ is the feature space of G with respect to some $(H_i \leq G, \rho_i : H_i \to \mathrm{GL}(V_i))$. In other words, $\mathcal{F}_i$ is the space of sections $\Gamma(E_i)$ of the associated vector bundle $E_i = G \times_{\rho_i} V_i \to G/H_i$. $\mathcal{F}_{i+1}$ is defined similarly.
2. The map $L_i : \Gamma(E_i) \to \Gamma(E_{i+1})$ is a linear map that is equivariant with respect to the induced representation structure (π_i and π_{i+1}) on both sides.

Equivariance and Convolutions

Recall the following **slogan** from Lecture 1:

"Translation equivariant linear maps are convolutions."

Since we have defined convolutions to a broader generality now, the **new slogan** we would hope is that:

"G-equivariant linear maps are G-convolutions."

Convolution is All You Need

Theorem (Convolution is All You Need [Cohen et al., 2019])

Under reasonable assumptions[6], a G-equivariant neural network layer is a G-convolutional neural network layer.

In other words, under reasonable hypothesis, a (bounded) equivariant linear map ϕ can be expressed as some convolution with respect to G.

[6] which will be clarified in the proceeding slides

Why Should We Expect This? (Proof Sketch)

Let us consider a (bounded) equivariant linear map

$$\phi : \Gamma(E_i) \cong \mathcal{I}_G^i \to \Gamma(E_{i+1}) \cong \mathcal{I}_G^{i+1}.$$

(Here $\mathcal{I}_G^i$ refers to the respective Mackey function).

In many nice cases, it is possible to represent ϕ as an integral transform. In the sense that, for any $f : G \to V_i \in \mathcal{I}_G^i$, there exists a two-argument kernel $\kappa^+ : G \times G \to \mathrm{Hom}(V_i, V_{i+1})$ such that

$$(\phi f)(y) = \int_G \kappa^+(y,x) f(x) d\mu_G(x)^7 \quad (\diamondsuit)$$

If V_i and V_{i+1} are both finite dimensional, we can fix a choice of basis to make this a matrix-valued kernel.

[7] The right hand side is called an integral transform.

Why Should We Expect This? (Proof Sketch)

Let π_i, π_{i+1} be the induced representations for $\mathcal{I}_G^i$ and $\mathcal{I}_G^{i+1}$ respectively. Since ϕ is equivariant w.r.t the induced actions on the functions, this means that for all $u, x \in G$ and $f \in \mathcal{I}_G^i$

$$[\phi(\pi_i(u) \cdot f)](x) = [\pi_{i+1}(u) \cdot (\phi(f))](x) \quad (\spadesuit).$$

Recall the action of an arbitrary induced representation π on $f : G \to V$ is

$$g \cdot f(x) \coloneqq f(g^{-1}x), \forall g, x \in G.$$

Expanding both sides of ($\spadesuit$) out using the integral transforms, we have that

$$\int_G \kappa^+(y,x) f(u^{-1}x) d\mu_G(x) = \int_G \kappa^+(u^{-1}y, x) f(x) d\mu_G(x). \quad (\square)$$

Why Does (♠) Expand to (□)?

Starting with the equivariant condition

$$[\phi(\pi_i(u) \cdot f)](x) = [\pi_{i+1}(u) \cdot (\phi(f))](x) \quad (\spadesuit),$$

we have that

$$\begin{aligned}\mathrm{LHS} &= [\phi(\, \pi_i(u) \cdot f \,)](y) \\ &= \int_G \kappa^+(y,x) \left[\pi_i(u) \cdot f\right](x) \, d\mu_G(x) \quad \text{by} \quad (\diamondsuit). \\ &= \int_G \kappa^+(y,x) \, f(u^{-1}x)(x) \, d\mu_G(x)\end{aligned}$$

where the last equality follows from the induced action on functions (pull backs on f) $(\pi_i(u) \cdot f)(x) = f(u^{-1}x)$.

For the RHS, we first apply the induced action and then the integral transform,

$$\begin{aligned}RHS &= [\pi_{i+1}(u) \cdot (\phi(f))](y) = \phi(f)(u^{-1}y) \\ &= \int_G \kappa^+(u^{-1}y, x) \, f(x) \, d\mu_G(x)\end{aligned}$$

Why Should We Expect This? (Proof Sketch)

Now we have that

$$\int_G \kappa^+(y,x)f(u^{-1}x)d\mu_G(x) = \int_G \kappa^+(u^{-1}y,x)f(x)d\mu_G(x).$$

Consider a change of variables $x \mapsto ux$ on the left hand side. This is okay since we are integrating over all of G, and we have that

$$\int_G \kappa^+(y,ux)f(u^{-1}ux)d\mu_G(x) = \int_G \kappa^+(y,x)f(u^{-1}x)d\mu_G(x)$$

$$= \int_G \kappa^+(u^{-1}y,x)f(x)d\mu_G(x)$$

Why Should We Expect This? (Proof Sketch)

Thus, we have that

$$\int_G \kappa^+(y, ux) f(x) d\mu_G(x) = \int_G \kappa^+(u^{-1}y, x) f(x) d\mu_G(x).$$

Since this equality holds for all f's, it implies that $\kappa^+(y, ux) = \kappa^+(u^{-1}y, x)$.

Doing another round of variable substitutions [8], we have that

$$\kappa^+(uy, ux) = \kappa^+(y, x)$$

for all $x, y, u \in G$.

[8] Consider relabeling $y \to uy$ in the equality $\kappa^+(y, ux) = \kappa^+(u^{-1}y, x)$.

Why Should We Expect This? (Proof Sketch)

Now since we have the equality

$$\kappa^+(uy, ux) = \kappa^+(y, x) \quad (\dagger),$$

we observe that

$$\begin{aligned} \kappa^+(y, x) &= \kappa^+(y\,(e),\ (e)\,x) \\ &= \kappa^+(y(e),\ (y\,y^{-1})\,x) \\ &= \kappa^+(e,\ y^{-1}x) \quad \text{by } (\dagger)^9 \end{aligned}$$

Now we can define our one-parameter convolutional kernel as

$$\kappa(y^{-1}x) := \kappa^+(e, y^{-1}x).$$

[9] Here we take the symbol y in this step to play the role of u in the LHS of $(\dagger)$.

Why Should We Expect This? (Proof Sketch)

$$\kappa(y^{-1}x) := \kappa^{+}(e, y^{-1}x) = \kappa^{+}(y, x).$$

Now for the map $\phi : \Gamma(E_i) \cong \mathcal{I}_G^i \to \Gamma(E_{i+1}) \cong \mathcal{I}_G^{i+1}$, we have that

$$\begin{aligned}(\phi f)(y) &= \int_G \kappa^{+}(y, x) f(x) d\mu_G(x) \\ &= \int_G \kappa(y^{-1}x) f(x) d\mu_G(x) \\ &= (\kappa \star f)(y).\end{aligned}$$

<u>Thus, ϕ can be written as a convolution!</u>

Some Technical Notes: Integral Representations

Question:

When can we represent ϕ as an integral transform with κ^+?

Representations of this sorts are usually linked to Dunford-Pettis-like results (recall Lecture 1), such as:

Theorem [See Theorem 1.3 of [Arendt, 1994]]:

Let $\mathcal{K} : L^p(X) \to L^\infty(Y)$ be linear, bounded operator with $1 \leq p < \infty$ then $\mathcal{K}$ admits an integral representation.

The requirement for L^∞ in the codomain indicates that something might go wrong when $X = Y$ if the domain is not compact.

Some Technical Notes: Integral Representations

- In practice, many ML researchers treat the ability to represent ϕ as an integral transform as a given assumption (or what [Weiler et al., 2025] calls ansatz).
- We do ultimately want to optimize the parameters in a neural network layer. The integral representation gives a more descriptive parameter in terms of certain matrix-valued functions.
- For equivariant linear maps that do not admit the integral representation description, it is sometimes difficult to write down such linear maps.

From now on, we also adopt this ansatz.

Bi-Equivariant Kernels

Let $\mathcal{H}$ be the collection of equivariant linear maps $\mathcal{I}_G^i \to \mathcal{I}_G^{i+1}$. There is an alternative characterization of $\mathcal{H}$:

Theorem ([Cohen et al., 2019])

$\mathcal{H}$ is isomorphic to the space of bi-equivariant kernels $\kappa : G \to \mathrm{Hom}(V_i, V_{i+1})$, satisfying the condition

$$\kappa(h_2 g h_1) = \rho_{i+1}(h_2) \circ \kappa(g) \circ \rho_i(h_1).$$

Here recall ρ_{i+1} and ρ_i are the representations $H_{i+1} \to \mathrm{GL}(V_{i+1})$ and $H_i \to \mathrm{GL}(V_i)$ respectively.

Left-Equivariant Kernels

Although Mackey functions and sections are equivalent in this context,

- A Mackey function specifies a map $f : G \to V$.
- Whereas a section admits a map with domain being G/H.

In this perspective, a Mackey function contains redundant information and sections save more memories. We would therefore like a characterization of $\mathcal{H}$ with G/H.

Theorem ([Cohen et al., 2019])

$\mathcal{H}$ is isomorphic to the space of left equivariant kernels $\overleftarrow{\kappa} : G/H_i \to \mathrm{Hom}(V_i, V_{i+1})$ satisfying

$$\overleftarrow{\kappa}(h_2 x) := \rho_{i+1}(h_2) \circ \overleftarrow{\kappa}(x) \circ \rho_i(\mathrm{h}_1(x, h_2)^{-1})$$

for all $h_2 \in H_{i+1}, x \in G/H_i$. Here $\mathrm{h}_1(x, g) := s(gx)^{-1} g$, where s is a choice of local section.

Kernels on the Double Coset

Since there are two groups involved, we might as well consider the double coset:

$$H_{i+1} \backslash G / H_i := \text{right cosets of } H_{i+1} \text{ acting on } G/H_i.$$

Theorem ([Cohen et al., 2019])

$\mathcal{H}$ is isomorphic to the space $\mathcal{K}_D$ comprising of functions

$$\overline{\kappa} : H_{i+1} \backslash G / H_i \to \mathrm{Hom}(V_i, V_{i+1}),$$

$$\overline{\kappa}(x) = \rho_{i+1}(h) \overline{\kappa}(x) \rho_i^x(h)^{-1}, \forall x \in H_{i+1} \backslash G / H_i, h \in H_{i+1}^{\gamma(x) H_i}$$

Here $\gamma : H_{i+1} \backslash G / H_i \to G$ is a choice of coset representatives:

- $H_{i+1}^{\gamma(x) H_i} = \{h \in H_{i+1} \mid h\gamma(x) H_i = \gamma(x) H_i\} \leq H_i$.
- $\rho_i^x(h) := \rho_i(\gamma(x)^{-1} h \gamma(x))$.

The takeaway is that this theorem is the most memory-efficient way to represent the data.

Outline

Example: 3D Steerable CNNs

The work of 3D Steerable CNNs [Weiler et al., 2018] is concerned with the group $\mathrm{SE}(3)$ of orientation preserving rigid body motions (ie. rotations and translations) in $\mathbb{R}^3$. This is called the special Euclidean group for $\mathbb{R}^3$.

In their work, the layer is built with the choice

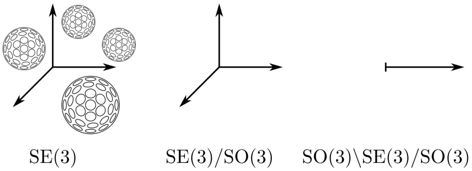
$$G = \mathrm{SE}(3) \text{ and } H = H_1 = H_2 = \mathrm{SO}(3).$$

and representations ρ_1 and ρ_2.

Note that G can be written as a certain twisted product[10] of $\mathrm{SO}(3)$ and $\mathbb{R}^3$.

[10] To be more precise, a semi-direct product.

Example: 3D Steerable CNNs



Picture from [Cohen et al., 2019].

Note in this diagram $G/H_1 \cong \mathbb{R}^3$ and $H_2\backslash G/H_1 \cong [0, \infty)$.

Theorem

The equivariant linear maps $\mathcal{H}$ can be identified with left-equivariant kernels $\overleftarrow{\kappa}$ such that

$$\overleftarrow{\kappa}(rx) = \rho_2(r) \circ \overleftarrow{\kappa}(x) \circ \rho_1(r^{-1}).$$

Example: 3D Steerable CNNs

Theorem

The equivariant linear maps $\mathcal{H}$ can be identified with left-equivariant kernels $\overleftarrow{\kappa}$ such that

$$\overleftarrow{\kappa}(rx) = \rho_2(r) \circ \overleftarrow{\kappa}(x) \circ \rho_1(r^{-1}), r \in \mathrm{SO}(3), x \in \mathbb{R}^3.$$

By the general characterization in the last section, we have that

$$\overleftarrow{\kappa}(rx) = \rho_2(r) \circ \overleftarrow{\kappa}(x) \circ \rho_1(\mathrm{h}_1(x, r)^{-1}),$$

and $\mathrm{h}_1(x, r) := s(rx)^{-1} r$ for s some local section. In this case, since $\mathrm{SE}(3)$ is a twisted product of $\mathbb{R}^3$ and $\mathrm{SO}(3)$, it is actually a trivial principal H-bundle, so we can actually choose s to be the identity section! Thus

$$\mathrm{h}_1(x, r)^{-1} = r^{-1}.$$

Outline

What About Practical Implementations?

Up to now we have developed the theory; now we will discuss some implementations.

Practical implementations ask: How do we discretize a (possibly continuous) group, parameterize kernels, and execute the resulting tensor operations efficiently?

Now we will briefly survey some works that have already translated G-convolution theory into code.

Examples

e2cnn[Weiler and Cesa, 2019] is a PyTorch add-on that builds $E(2)$-equivariant CNN layers whose kernels are automatically parameterized to rotations and reflections.

[Gupta et al., 2021] gives a rotation equivariant siamese network for tracker that estimates in-plane pose and lifts the SOTA on the Rotating-Object Benchmark.

Examples

e3nn [Geiger and Smidt, 2022] is also a PyTorch library providing $E(3)$-equivariant convolutional operators via steerable filters built from irreducible SO(3) representations.

escnn [Cesa et al., 2022] is a successor to e2cnn that proposes a general procedure to build arbitrary CNNs with respect to any compact group G.

- In practice, they considered a way to construct G-steerable (equivariant) kernels with any $G \leq O(3)$.
- This gave a way to build CNNs with respect to symmetries of platonic solids or choosing $G = \mathrm{SO}(2)$ in 3D to only have azimuthal symmetries.
- Achieve SOTA on volumetric datasets $\mathrm{ModelNet10}$ [Wu et al., 2015], a rotated version of it, and LBA [Townshend et al., 2022].

References I

Arendt, Wolfgang, B. A. V. (1994).
Integral representations of resolvents and semigroups.
Forum mathematicum, 6(1):111–136.

Cesa, G., Lang, L., and Weiler, M. (2022).
A program to build e(N)-equivariant steerable cnns.
International Conference on Learning Representations (ICLR) 2022.

Cohen, T. S., Geiger, M., and Weiler, M. (2019).
A general theory of equivariant CNNs on homogeneous spaces.
Curran Associates Inc., Red Hook, NY, USA.

Geiger, M. and Smidt, T. E. (2022).
e3nn: Euclidean neural networks.

References II

Gupta, D. K., Arya, D., and Gavves, E. (2021).
Rotation-equivariant siamese networks for tracking.

Townshend, R. J. L., Vögele, M., Suriana, P., Derry, A., Powers, A., Laloudakis, Y., Balachandar, S., Jing, B., Anderson, B., Eismann, S., Kondor, R., Altman, R. B., and Dror, R. O. (2022).
Atom3d: Tasks on molecules in three dimensions.

Weiler, M. and Cesa, G. (2019).
General $e(2)$-equivariant steerable cnns.

Weiler, M., Forré, P., Verlinde, E., and Welling, M. (2025).
Equivariant and coordinate independent convolutional networks.
WORLD SCIENTIFIC.

References III

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. S. (2018).
3d steerable cnns: Learning rotationally equivariant features in volumetric data.
In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015).
3d shapenets: A deep representation for volumetric shapes.