Group Invariant Global Pooling

Kamil Bujel [*1] **Yonatan Gideoni** [*1] **Chaitanya K. Joshi** [1] **Pietro Liò** [1]

Abstract

Much work has been devoted to devising architectures that build group-equivariant representations, while invariance is often induced using simple global pooling mechanisms. Little work has been done on creating expressive layers that are invariant to given symmetries, despite the success of permutation invariant pooling in various tasks. In this work, we present Group Invariant Global Pooling (GIGP), an invariant pooling layer that is provably sufficiently expressive to represent a large class of invariant functions. We validate GIGP on rotated MNIST and QM9, showing improvements for the latter while attaining identical results for the former. By making the pooling process over group orbits, this invariant aggregation method leads to improved performance, while performing well-principled group aggregations.

1. Introduction

Incorporating equivariance into architectures is a hard problem, with much of geometric deep learning being devoted to doing so. While architectures with specific equivariances have been successful (Sanchez-Gonzalez et al., 2020), their construction involves encoding geometric priors for each symmetry. Much effort has been devoted to building architectures that are equivariant to generic pre-specified symmetries (Finzi et al., 2020).

Leveraging symmetries in deep learning has yielded state-of-the-art results for various tasks. Scientific problems are prime examples, as they exhibit an abundance of natural symmetries. One architecture that was used for this is LieConv (Finzi et al., 2020), an architecture that is equivariant to generic continuous Lie groups. Finzi et al. (2020) demonstrate its efficacy on tasks such as classifying rotated images, predicting molecule's properties, and simulating physical systems.

However, most recent work has been devoted to the design of equivariant layers, with few trying to build expressive invariant layers. Invariance is often achieved through global or mean pooling the final layer's results, which causes a loss of information about the data's structure as all orbits are treated the same. Following recent developments on invariant pooling (Laptev et al., 2016a; Yang et al., 2020; Azizian & Lelarge, 2020) and to alleviate this loss of information, we present **Group Invariant Global Pooling** (**GIGP**), an orbit-aware invariant pooling layer, with **our contributions being as follows:**

1. We propose GIGP, group invariant global pooling, and prove that any invariant function that fulfills certain conditions can be expressed as a decomposition like that of GIGP's.
2. We use GIGP as a pooling method for LieConv and evaluate it on rotated MNIST and QM9. GIGP's orbit-aware invariant pooling leads to slightly improved results on both tasks.

2. Background

2.1. Groups and Equivariance

Most natural systems can be described by features that remain unaltered under certain actions, known as symmetries. This can be rigorously expressed using group theory. A group $(G, \circ)$ is a set G of elements $u \in G$ together with a closed associative binary operation $\circ : G \times G \to G$, an inverse $u^{-1} \in G$ and the identity element e. A canonical example is the rotation group $SO(2)$, which denotes all two-dimensional rotations in Euclidean space.

These abstract group elements can be represented by matrices and vectors in a way that retains their structure. For some groups each element $u \in G$ can be represented using a mapping $\rho : G \to GL(n)$ that associates it with an invertible matrix $\rho(u) \in \mathbb{R}^{n \times n}$. This representation must be a homomorphism to retain the group's structure. This allows expressing group elements as transformations over $\mathbb{R}^n$, thus making them easier to study. For $SO(2)$ the rep-

[*]Equal contribution [1]University of Cambridge. Correspondence to: Kamil Bujel <kdb36@cl.cam.ac.uk>, Yonatan Gideoni <yg403@cl.cam.ac.uk>.

Presented at the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

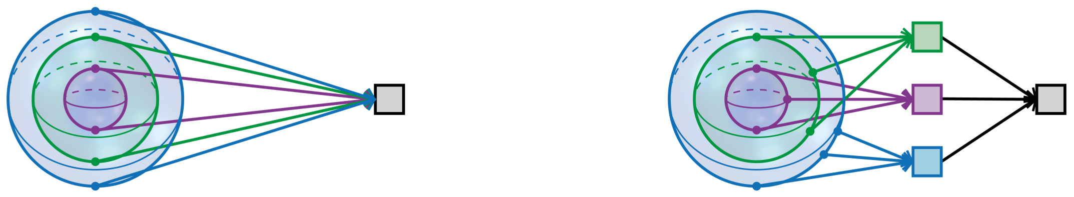

Figure 1: Visualization of the standard global pooling method (left) and our proposed Group-Invariant Global Pooling (right). Each sphere represents an orbit in $SO(3)$. GIGP builds group-invariant representations by aggregating each orbit separately, unlike global pooling. This leads to better-principled aggregation that is designed to respect the given symmetry.

resentation $\rho(u)$ yields the well known rotation matrices $R(t) = \left(\begin{smallmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{smallmatrix}\right)$ for an angle $t \in \mathbb{R}$.

In particular, group representations allow studying how functions change when transforming their inputs. An important class is **G-equivariant** functions, functions where a transformation in their input leads to a corresponding change in their output. Formally, $f : X \to Y$ is G-equivariant iff

$$\forall x \in X, g \in G: \; f(\rho_X(g)(x)) = \rho_Y(g)f(x), \quad (1)$$

where ρ_X, ρ_Y denote different representations of the group element. Invariance is then a special case when $\rho_Y = I$.

2.2. Group Convolution

The notion of equivariance through convolutions was generalized by Cohen & Welling (2016) to groups, yielding group convolutions. This allows the definition of convolutions that are equivariant to generic symmetries (Kondor & Trivedi, 2018), with them in this case being over functions of group elements $k, f : G \to \mathbb{R}$. It is defined as

$$(k * f)(u) \equiv \int_G k(v^{-1}u)f(v)d\mu(v) \quad (2)$$

where $d\mu(v)$ is the Haar measure, a measure that allows consistently defining integrals over groups.

2.3. Lifting

As most group-equivariant architectures operate on Euclidean coordinates $x \in \mathbb{R}^n$, not the group elements $u \in G$, one needs to convert coordinates to group elements. To do so, a *lifting* method Lift(x_i): $x_i \to u_{ik}$ is applied to convert coordinates into group elements (Kondor & Trivedi, 2018),

$$\text{Lift}(x) = \{u \in G : uo = x\}, \quad (3)$$

where $o \in \mathbb{R}^n$ is a chosen origin that maps to the identity. Each coordinate x_i is lifted into K group elements u_{ik} so it becomes non-ambiguous. For example, $K = 1$ for $SO(2)$ or $K = 2$ for $\mathbb{R}\times$ $SO(2)$.

Standard lifting can lead to a loss of information. For instance, given a pair of points sampled from 2 different rotation symmetric equations,[1] their lifted representations encode only the rotation angle. Finzi et al. (2020) proposed encoding group *orbits* during the lifting process. Intuitively, orbits represent a set of points that can all be reached through group transformations from the same origin o. Formally, a quotient space $Q = X/G$ contains all orbits $q \in Q$ of G present in the input coordinates $X \subset \mathbb{R}^n$, which are the information the lifting discards. To fix this, the lifting procedure is performed separately for each orbit to return pairs $(u_i, q_i) \in G \times Q$ that contain the group element u_i and the orbit identifier q_i. This ensures that no information is lost.

2.4. LieConv

To facilitate learning over Lie groups, Finzi et al. (2020) proposed *LieConv*, an architecture that uses Lie theory to perform group-equivariant convolutions over arbitrary Lie groups. This is done by mapping the input data, expressed as pairs of coordinates and values $\{(x_i, f_i)\}\, x_i \in \mathbb{R}^n$, to corresponding group elements and efficiently convolving them. A summary of this procedure is given below.

While group convolution could be performed as per Equation 2, with the kernel k being an MLP, this would yield poor results. This is because canonical neural networks perform poorly on non-Euclidean data, requiring one to convert the group elements to a Euclidean domain. The Lie algebra $\mathfrak{g}$ is one such space, where the group elements can be converted to it using the logarithm map $a = \log(u)$.

This lifting procedure and mapping to the Lie algebra elements results in the LieConv convolution equation

$$h(u, q_u) = \int_{v \in \text{nbhd}(u)} k(\log(v^{-1}u), q_u, q_v)f(v, q_v)d\mu(v) \quad (4)$$

where the integral is over a neighborhood of points to ensure locality, analogous to standard convolutions in deep learning. As the exact integral is intractable, it is approximated using

[1]e.g. $4 = x^2 + y^2$ and $9 = x^2 + y^2$

a Monte-Carlo estimate, thus yielding only approximate equivariance. This assumes that the group elements are locally distributed proportionally to the Haar measure.

Akin to standard convolutional networks, multiple layers of LieConv are then combined to form a ResNet architecture, with global pooling as the last layer.

3. Group Invariant Global Pooling

While the equivariant convolutions make the model respect a symmetry, they do not assure that the model's predictions are invariant. Thus, it is natural to introduce a group invariant global pooling layer, analogous to the aggregation functions in standard graph neural networks (GNNs). To make invariant global predictions it is sufficient to build invariant representations of each orbit. In this work, we propose a general framework to generate such orbit-aware invariant features.

We decompose a sum over the group elements $v \in G$ into a sum over orbits $q \in Q$ and group elements within each orbit $u \in G^q$:

$$\sum_{v \in G} f(v) = \sum_{q \in Q} \sum_{u \in G^q} f(u) \tag{5}$$

where $u \in G^q \subset G$ such that $[u, q] \in G \times Q$, indicating the u is observed within the orbit q. The inner sum is invariant to the group transformations if $f(u)$ is equivariant, as transformed elements always stay within the same orbit.

We can now use this decomposition to obtain an aggregation method that uses an MLP Φ to build invariant representations of orbits. We call this group invariant global pooling **(GIGP)**, defined as

$$\text{GIGP}(f, G) = C \sum_{q \in Q} w_q \Phi \left(\sum_{u \in G^q} f(u), q \right) \tag{6}$$

where w_q are learnable parameters and Φ is a neural network. C is a constant that can be tuned to make GIGP, for example, initialized as a standard pooling layer. w_q are used to weigh orbits according to their relevance.

As most architectures use a global mean-pooling aggregation layer, Φ and C are initialized so that GIGP before training performs global mean-pooling. This was found to improve results in practice. Moreover, to allow soft invariance, nearby orbits are aggregated together based on their distance from the main orbit q. This allows the application of GIGP to sparse data, such as point clouds and molecules.

Finally, we prove GIGP's expressivity by showing that any invariant continuous function has an identical decomposition to GIGP preceded by an equivariant network and followed by an MLP. To prove it, we generalize the Deep Sets proof by Zaheer et al. (2017) from permutation invariant operations to arbitrary non-homogeneous groups. The proof and exact theorem are given in Appendix A.

Theorem 3.1. *(Sufficient Expressivity of GIGP, Informal) Let $f : X \to F$ be some invariant function that takes a set of group elements as input. Then, there exist functions ρ, ϕ, ψ such that $f(X) \equiv \rho(\sum_{q \in Q} \phi[\sum_{x \in q} \psi(x)])$.*

For simplicity, GIGP was used without an appended network, thereby setting ρ to be proportional to the identity. Understanding when this form is also sufficient is left for future work.

4. Experiments

All experiments were run on 16GB Nvidia Tesla P100 GPUs. The rotMNIST/QM9 models took on average $4/16$ hours respectively to train for 500 epochs. For both architectures the hyperparameters given in Finzi et al. (2020) were used, which are a learning rate of 0.003, a batch size of 32, and a limit on each point's neighborhood to 32 nearest neighbors for rotMNIST and batch size of 75 and neighborhood of 100 for QM9. Hyperparameter tuning was not performed and is left for future work. $SO(n)$ was used as the symmetry group for both models, being the simplest non-homogeneous symmetry both exhibit. The models are implemented in PyTorch. The results are shown in Table 1.

4.1. Rotated Images

We use the rotMNIST dataset, which contains rotated images of digits from the popular image recognition MNIST task. These images are invariant under the rotation group $SO(2)$. We use the original splits of 10k for training, 2k for validation, and 50k for the test set.

To test GIGP, we evaluate it on LieConv, replacing its global mean pooling layer. Using the original hyperparameters and training procedure, we achieve comparable results, as shown in Table 1. We believe this result on rotMNIST, a saturated task, validates that our method is not only as expressive as standard pooling, but does not impair training.

4.2. Molecules

We also evaluate our proposed pooling method on the QM9 molecular property task. The dataset contains molecules as 3D coordinates of atoms and their respective atomic charges. The coordinates have no canonical origin, therefore they are $E(3)$ equivariant and well-performing models must use that geometric prior. To incorporate $SO(3)$ symmetry, all molecules were re-centered. We report results on the `Homo` task of predicting the energy of the highest occupied orbital. The dataset contains 100k molecules in the train set, 10% in the test set and the remaining ones in

Table 1: Results when appending GIGP to LieConv on various tasks. rotMNIST has an $SO(2)$ symmetry while pre-processed QM9 has $SO(3)$. QM9 had only a single run because of resource constraints.

	rotMNIST		QM9 ϵ_{HOMO}	
	Training Error %	Test Error %	Train MAE meV	Test MAE meV
LieConv + Simple Global Pooling	0.05 ± 0.05	1.42 ± 0.13	4.8	30.6
LieConv + GIGP	0.03 ± 0.05	1.38 ± 0.07	4.7	29.4

the validation set. As shown in Table 1, the resulting GIGP results are slightly better than those attained using normal global pooling. These results are promising, as they were achieved with no hyperparameter tuning.

5. Conclusion

We presented GIGP, an expressive pooling layer that is invariant to generic symmetries. It builds group-invariant representations of all elements on the same orbits, leading to pooling operation that is better aligned with the task at hand. This mechanism addresses the lack of invariant pooling methods for group-equivariant architectures. We proved GIGP is sufficiently expressive to model a large class of invariant functions and demonstrated it yields slight improvements on a popular image and molecular datasets.

Acknowledgements

We would like to thank Petar Veličković for helpful comments and discussions. We would also like to thank Jonas Jürß and Dulhan Jayalath for comments on an earlier version of this work. We would also like to thank Peini Yu for helping us to create Figure 1.

References

Azizian, W. and Lelarge, M. Expressive power of invariant and equivariant graph neural networks. In *ICLR*, 2020.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *ICML*, pp. 2990–2999. PMLR, 2016.

Cohen, T. S. and Welling, M. Steerable CNNs. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=rJQKYt5ll`.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *ICML*, pp. 3165–3176. PMLR, 2020.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, pp. 2747–2755. PMLR, 2018.

Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. Ti-pooling: Transformation-invariant pooling for feature learning in convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 289–297, 2016a.

Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 289–297, 2016b.

Ravanbakhsh, S., Schneider, J., and Poczos, B. Equivariance through parameter-sharing. In *ICML*, pp. 2892–2901. PMLR, 2017.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to simulate complex physics with graph networks. In *ICML*, pp. 8459–8468. PMLR, 2020.

van der Ouderaa, T. F., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters. In *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=5oEk8fvJxny`.

Weiler, M. and Cesa, G. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

Xu, J., Kim, H., Rainforth, T., and Teh, Y. Group equivariant subsampling. *Advances in Neural Information Processing Systems*, 34:5934–5946, 2021.

Yang, M., Shen, Y., Qi, H., and Yin, B. Breaking the expressive bottlenecks of graph neural networks. *ArXiv*, abs/2012.07219, 2020.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. Deep sets. In *NIPS*, 2017.

A. Proof of GIGP Expressivity

Theorem A.1. *Let $f : F^X \to \mathbb{R}$ be some function that is invariant to a group's action, so $\forall g \in G : f(gx) = f(x)$. Here F is some set representing the possible values given to each group element. Thus, assuming:*

1. *X, F are countable.*
2. *There are a finite number of orbits.*

Given these assumptions, there exist functions ρ, ϕ, ψ such that $f(X) = \rho(\sum_{q\in Q} \phi[\sum_{x\in q} \psi(x)])$, where with some abuse of notation we denote the value given to each group element over the domain with the group element directly.

Proof: The proof goes as follows. We shall show that there exists ψ such that $\sum_{x\in q} \psi(x)$ is group invariant and constitutes an invertible function from F^X to a countable set. Afterwards we'll show that there exists a function ϕ such that $\sum_{q\in Q} \phi(h(q))$ is an invertible function from $Q^{\mathbb{N}}$ to a countable set. Finally, we'll show that we can construct a function ρ such that $f(X') = \rho(\sum_{q\in Q} \phi[\sum_{x\in q} \psi(x)])$.

Step 1: As F is a countable domain, we have that $\exists c : c(x)$ is an invertible mapping to $\mathbb{N}$. Because X is countable each orbit has countably many elements. Thus, one can define $L(x) := \ln(p_{c(x)})$, where p_i denotes the i-th prime. Note that this constitutes a one-to-one correspondence between sets $\{x\}_{x\in q}$ and logarithms of numbers as each number has a unique prime representation and $\sum_{x\in q} L(x) = \ln(\prod_{x\in q} p_{c(x)})$. Thus, we have shown that there exists a function $\psi(x) := \ln(p_{c(x)})$ that is an invertible mapping between F^X and a countable set, specifically $\ln(\mathbb{N})$. This is trivially group invariant as the action of the group simply permutes the set.

Step 2: We denote an arbitrary function from an orbit to the real numbers as $h : Q \to \mathbb{R}$. We want to show that there exists a function ϕ such that $\sum_{q\in Q} \phi(h(q))$ is invertible from $Q^{\mathbb{N}}$, all the multisets of the form $\{\{h(q)\}\}_{q\in Q}$ to $\mathbb{N}$. As there is a finite number of orbits, one can use a similar construction to the previous step - $h(q)$ can be treated as a natural number (for example, calculating $\exp(h(q))$ if following the previous construction) and define $\phi(n) := \ln(p_n)$.

Step 3. Let $g(X) := \sum_{q\in Q} \phi[\sum_{x\in q} \psi(x)]$. This is an invertible group-invariant function from F^X to $\ln(\mathbb{N})$. Thus, setting $\rho := f \circ g^{-1}$ completes the proof. $\square$

B. Related Work

Group-Equivariant Convolution Cohen & Welling (2016) generalized the notion of architecture equivariance to arbitrary groups. Later works extended this by using irreducible group representations to perform convolution with respect to various continuous groups and their discrete subgroups (Cohen & Welling, 2017; Weiler & Cesa, 2019; Geiger & Smidt, 2022).

However, there is a lack of work that attempts to generate architectures equivariant to any symmetry group. Ravanbakhsh et al. (2017) achieve equivariance to any finite group by sharing weights over orbits. Finzi et al. (2020) present LieConv, which is equivariant to arbitrary Lie groups and scales well to larger datasets.

Group Invariant Pooling Laptev et al. (2016b) introduce transformation-invariant pooling for CNNs. Cohen & Welling (2016) propose coset pooling that is equivariant to any discrete group. **?** propose scale-invariant pooling in CNNs. van der Ouderaa et al. (2022) obtain G-invariant global pooling for non-stationary kernels. Xu et al. (2021) propose group equivariant/invariant subsampling for discrete groups. However, there is no work that implements group invariant global pooling for arbitrary groups. We attempt to fill this gap.