Learning Polynomial Problems with SL(2)-Equivariance

Hannah Lawrence [*1] **Mitchell Tong Harris** [*2]

Abstract

We introduce a set of polynomial learning problems that are equivariant to the non-compact group $SL(2,\mathbb{R})$. $SL(2,\mathbb{R})$ consists of area-preserving linear transformations, and captures the symmetries of a variety of polynomial-based problems not previously studied in the machine learning community, such as verifying positivity (for e.g. sum-of-squares optimization) and minimization. While compact groups admit many architectural building blocks, such as group convolutions, non-compact groups do not fit within this paradigm and are therefore more challenging. We consider several equivariance-based learning approaches for solving polynomial problems, including both data augmentation and a fully $SL(2,\mathbb{R})$-equivariant architecture for solving polynomial problems. In experiments, we broadly demonstrate that machine learning provides a promising alternative to traditional SDP-based baselines, achieving tenfold speedups while retaining high accuracy. Surprisingly, the most successful approaches incorporate only a well-conditioned subset of $SL(2,\mathbb{R})$, rather than the entire group. This provides a rare example of a symmetric problem where data augmentation outperforms full equivariance, and provides interesting lessons for other problems with non-compact symmetries.

1. Introduction

In recent years, there has been an explosion of learning problems on non-traditional data types: 3D vision applications often consider point clouds as input, partial differential equation (PDE) data involves time-evolution of functions over irregularly shaped domains, and diverse inputs ranging from social networks to molecules are represented as graphs, often embedded in 3D space. Especially valuable among these (often) scientific problems are those for which output correctness can be verified efficiently; this is the case for e.g. modeling catalysts, where a candidate conformation can be confirmed with a few DFT steps (Chanussot et al., 2021).[1]

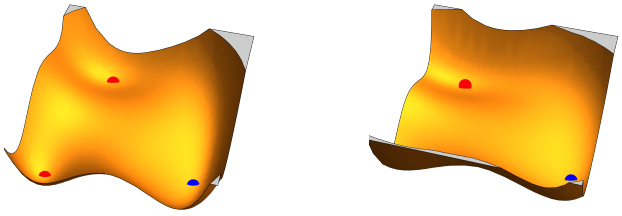
In this work, we introduce a new family of non-traditional inputs, ubiquitous in mathematics and engineering: *polynomials*. Primitives such as minimizing a polynomial or certifying its positivity via a sum of squares decomposition have attracted attention due to their direct applications in operations research and control theory (Henrion & Garulli, 2005). Polynomial inputs are particularly well-suited to $SL(2,\mathbb{R})$-equivariance because they lie in its finite-dimensional irreducible representations. As a result, it is possible to achieve *exact* equivariance to area-preserving linear transformations — subject only to numerical error from finite precision, and not to Monte Carlo integral approximations as in much previous work (Finzi et al., 2020). Therefore, we extend the equivariance research program for unusual input data to datasets of *polynomials*, taking into account the inherent symmetries of polynomial-based problems.

As shown in Figure 1, the minimizer of a polynomial tranforms equivariantly with respect to linear transformations of the input variables, while the minimum value itself is invariant. The general linear group, however, is noncompact. Unfortunately, existing techniques are largely tailored towards compact groups. There are several difficulties unique to non-compact groups, which arise due to the following essential fact: a non-compact group G does not have a finite, invariant (Haar) measure. For compact groups, it is nearly always assumed in prior work that all datapoints in a given orbit are equally likely. In contrast, any non-trivial probability distribution over $\mathcal{X}$ is not necessarily the same as $g\mathcal{X}$ for $g \in G$ when G is non-compact. **Therefore, data augmentation induces a distribution shift.** In this sense, non-compact equivariance is linked to the notion of "extrinsic equivariance" Wang et al. (2022), in which group transformations change the support of the data distribution. Further challenges associated with the ill-conditioning of matrices in our polynomial task are discussed in Appendix D. In response

[*]Equal contribution, author order determined by coin flip [1]Department of Computer Science, Massachusetts Institute of Technology [2]Department of Mathematics, Massachusetts Institute of Technology. Correspondence to: Hannah Lawrence <hanlaw@mit.edu>, Mitchell Tong Harris <mitchh@mit.edu>.

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

[1]Further inspirational examples are mentioned in Appendix A.



(a) $g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ (b) $g = \begin{pmatrix} 1.2 & 0 \\ 0 & \frac{1}{1.2} \end{pmatrix}$

Figure 1: **The action of $SL(2,\mathbb{R})$ on polynomials.** Each plot is the result of applying some $g \in SL(2,\mathbb{R})$ to a particular degree 6 bivariate polynomial, stretching or compressing the polynomial along each coordinate axis. The local minima are marked in red, and the global minimum in blue. While the minimum value is invariant, the minimizer transforms equivariantly under $SL(2,\mathbb{R})$.

to these challenges, we also consider the compact subgroup $SO(2,\mathbb{R})$, and data augmentation over well-conditioned subsets of $SL(2,\mathbb{R})$.

1.1. Our Contributions

In this work, our main contributions are twofold. First, we propose the application of machine learning methods to certain polynomial problems of interest to the optimization and control communities. To the best of our knowledge, this is the first application of machine learning to these problems. We demonstrate the potential for significant speedups over existing solvers for high-degree inputs, even with simple, fully-connected architectures. Second, we contextualize this problem through the lens of symmetries. We first build a universal $SL(2,\mathbb{R})$ -equivariant architecture suited to the non-compact input and output symmetries of polynomial tasks. However, we find that it is better to use *data augmentation* with respect to only *well-conditioned* elements of $SL(2,\mathbb{R})$. We hope our explorations provide a thought-provoking contribution to the broader equivariance literature, as a detailed case study of a non-compact group. By closely evaluating the behavior of equivariance with respect to a non-compact group on a concrete, richly-understood application problem, we provide a cautionary note on the incorporation of non-compact symmetries, and an invitation to lean more heavily on data augmentation in such cases.

2. Task: Sum-of-squares Certificate of Polynomial Nonnegativity

We first motivate the consideration of polynomial inputs with an important $SL(2,\mathbb{R})$-equivariant polynomial problem in the sum-of-squares literature, which will be our main focus: positivity verification.

One method to prove that a polynomial is nonnegative is by the following semidefinite programming (SDP) method (Parrilo, 2000). Define the $d-$lift as $\vec{x}^{[d]} = \begin{pmatrix} y^d & xy^{d-1} & \cdots & x^d \end{pmatrix}^T$. For $A \in SL(2,\mathbb{R})$, define the *induced* matrix $A^{[d]}$ as the $(d+1) \times (d+1)$ matrix for which $(A\vec{x})^{[d]} = A^{[d]}\vec{x}^{[d]}$ (Parrilo & Jadbabaie, 2008).

Suppose $p(\vec{x}) = \vec{x}^{[d]^T} Q \vec{x}^{[d]}$ for some positive semidefinite (PSD) $(d+1)\times(d+1)$ symmetric matrix Q. Then the right hand side quadratic form is nonnegative by definition of PSD. The equality implies p is nonnegative for all x and y. The problem of finding such a Q is traditionally relegated to a convex optimization interior point solver. One convenient formulation given to a solver is $f(p) = (\text{argmax} \log\det Q$ such that $p(\vec{x}) = \vec{x}^{[d]^T} Q \vec{x}^{[d]}$ and $Q \succeq 0)$, because this objective finds the analytic center of the feasible region. The optimizer $f(p) = Q^*$ satisfies the equivariance property $f(p(A\vec{x})) = A^{[d]^T} f(p(\vec{x})) A^{[d]}$.

This application is especially well-suited for machine learning, because we often only care if there exists some PSD Q. If the model predicts any Q satisfying $p(\vec{x}) = \vec{x}^{[d]^T} Q \vec{x}^{[d]}$ and $Q \succeq 0$, then p is automatically nonnegative; it is simple to validate this "machine-learned" certificate of nonnegativity. This could *accelerate* the search for sum of squares certificates in any of the problems mentioned in the introduction and Appendix A, because SDP solvers are only effective for moderate dimension (Mittelmann, 2003).

3. $SL(2,\mathbb{R})$-Equivariance for Polynomials

Now, we define key terms and concepts that will be used throughout the paper. The group $SL(2,\mathbb{R})$ consists of 2×2 real-valued matrices of determinant 1. A group G acts on a set $\mathcal{X}$ if $g(hx) = (gh)x$ and $ex = x$. In this case, the orbit of $x_0 \in \mathcal{X}$ is $\{gx_0 | g \in G\}$. A *representation* of G is a vector space V and a map $\rho : G \to GL(V)$ satisfying $\rho(g_1)\rho(g_2) = \rho(g_1 g_2)$ for all $g_1, g_2 \in G$. An *irreducible representation*, or irrep, is a representation for which there does not exist a subspace $W \subseteq V$ satisfying $\rho(g)w \in W$ for all $g \in G, w \in W$. An *equivariant* map is a map $f : V_1 \to V_2$ between representations (V_1, ρ_1) and (V_2, ρ_2) satisfying $f(\rho_1(g)v_1) = \rho_2(g) f(v_1)\ \forall\, g \in G, v_1 \in V_1$.

A reducible representation V_3 may be decomposed into a direct sum of irreps V_1 and V_2. When V_3 is itself a tensor product of irreps, this is known as the Clebsch-Gordan decomposition. The relation $V_3 \simeq V_1 \oplus V_2$ means that there exists an invertible, equivariant, linear map $T : V_3 \to V_1 \oplus V_2$.

The condition number κ of a matrix M is given by the ratio of its maximum to its minimum singluar values. A homogeneous polynomial, or *form*, is one in which every term has the same degree. We work with homogeneous polynomials in two real variables, or *binary forms*. Let

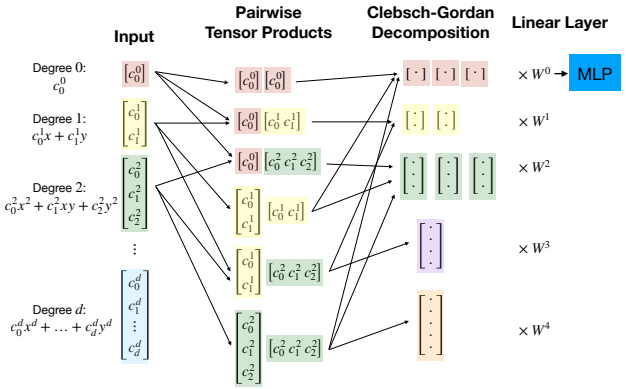


Figure 2: **Our $SL(2,\mathbb{R})$-equivariant layer.** The color indicates the irrep vector space of a given activation. The inputs to the layer live in the finite-dimensional irreps of $SL(2,\mathbb{R})$, corresponding to homogeneous polynomials of different degrees. The nonlinear layer takes all pairwise tensor products of these inputs, and the resultant matrices are decomposed back into elements of the irreps using the Clebsch-Gordan decomposition (T). The equivariant linear layer recombines elements within each irrep's vector space separately. Finally, we apply an MLP to the invariants. The outputs of the layer again reside in the irrep vector spaces, and so we can compose several layers of this form.

$V(d)$ denote the vector space of binary forms of degree d. The action of $g \in SL(2,\mathbb{R}) \subset \mathbb{R}^{2\times 2}$ on a binary form p is defined in Section 3.1 and shown in Figure 1.

3.1. Representation Theory of SL(2)

All finite-dimensional irreps of $SL(2,\mathbb{R})$ are given by $V(d)$, which is $(d+1)$-dimensional, for $d \in \mathbb{Z}^+$. Let $p \in V(d)$, and define the corresponding action on $V(d)$, ρ_d, by $\rho_d(g)p(\vec{x}) = p(g^{-1}\vec{x})$. These polynomial representations constitute all of $SL(2,\mathbb{R})$'s finite-dimensional irreps.

Given two irreps, $V(d_1)$ and $V(d_2)$, the tensor product decomposes as $V(d_1) \otimes V(d_2) \simeq \bigoplus_{n=0}^{\min(d_1,d_2)} V(d_1+d_2-2n)$. The linear map verifying this isomorphism is an invertible, equivariant map we call T, which is the Clebsch-Gordan decomposition for $SL(2,\mathbb{R})$ (Bohning & von Bothmer, 2010). Let $p(x_1,y_1) \in V(d_1)$ and $q(x_2,y_2) \in V(d_2)$. The d_1+d_2-2n component of $T(p\otimes q)$ is given by the classical nth order *transvectant*:

$$\psi_n(p,q) = \left(\left(\frac{\partial^2}{\partial x_1 \partial y_2} - \frac{\partial^2}{\partial x_2 \partial y_1}\right)^n p \cdot q\right)\Bigg|_{\substack{x=x_1=x_2 \\ y=y_1=y_2}}$$

3.2. Description of Equivariant Architecture

Each layer includes a nonlinear operation and a linear operation, which are described in Figure 2. The inputs to our polynomial network are conveniently already elements of the finite-dimensional irreps of $SL(2,\mathbb{R})$: each input channel has a list of polynomials $p_0 \in V(0), p_2 \in V(2), \ldots, p_d \in V(d)$. After composing these layers, it remains only to specialize the last layer to a given application.

Last layer The inputs to the last layer L are elements of the irreducible representation spaces, but the outputs may live in a reducible representation space V: $L : \bigoplus_{n=0}^{d} V(d) \to V$ for some reducible representation V. The decomposition of V into irreps dictates the structure of an equivariant last layer; see Appendix B.1 for details. For the positivity verification task, we furthermore are able to design the last layer so that the output Q for input p satisfies $p = \vec{x}^{[d]^T} Q x^{[d]}$.

The resultant equivariant architecture is universal, the proof of which follows from Bogatskiy et al. (2020) and is summarized in Appendix B.4. Note that, while one could hope to use invariant polynomials instead, a complete list of algebraically independent invariants is known only for low degrees (Popoviciu Draisma, 2014).

We also define an $SO(2)$-equivariant architecture (using more standard tools for compact groups); see Appendix B.3.

Model Name	Description
mlp-aug-rots	MLP with rotation augmentations
mlp-aug-a-b	MLP with augmentations by $g^{[d]}$, where $a \le \kappa(g) \le b$
so2net-aug-a-b	$SO(2)$ equivariant network, with data augmented by $g^{[d]}$ as above
so2net	$SO(2)$ equivariant architecture
sl2net	$SL(2,\mathbb{R})$ equivariant architecture
MLP	fully connected architecture

Table 1: Model names and meanings.

4. Experiments

Since there do not yet exist standardized benchmarks for these tasks, we design (and plan to release) our own tasks and datasets for polynomial minimization and positivity verification. Here we restrict our attention to the more interesting positivity verification problem, and defer an exploration of polynomial minimization to Appendix C. To generate synthetic data, we sample degree 6 polynomials p from a rotation-invariant distribution, and solve $f(p)$ from Section 2 with an interior point method (ApS, 2022).

We compare several instantiations of equivariant learning, summarized in Table 1. Of note is the use of random $SL(2,\mathbb{R})$ augmentations, with carefully controlled condition numbers, at train time. Dataset and experimental details, as well as equivariance tests, are included in Appendix C.

Degree	8	10	12	14
NMSE	6.0e-5	2.9e-5	2.3e-5	1.1e-5
MLP	0.082	0.17	0.22	0.29
Mosek (ApS, 2022)	17.50	12.35	13.27	12.35

Table 2: Runtimes (min) on a single CPU for 5,000 examples of specified input degree, and normalized mean-squared error (NMSE) of the trained networks on unseen data.

Timing comparison: network vs solver The primary impetus for turning to machine learning is to find faster ways of solving polynomial problems in practice. We compare the runtime of the traditional solves with that of a trained MLP. The MLP is about two orders of magnitude faster, while still very accurate, as reported in Table 2. These results provide strong motivation for applying machine learning to this task.

Comparison of equivariance and augmentation for out-of-distributional generalization In Figure 3, we report the test errors for each of the models. In addition to the original test dataset, we also evaluated increasingly ill-conditioned $SL(2,\mathbb{R})$-augmentations of the test set. The leftmost points are the test errors on the original test dataset.

On the untransformed test set, training an MLP with augmentations close to rotations yields the best results, followed by the so2net. An advantage of the so2net is that it has about half as many parameters (see Appendix C.1). Using high condition numbers for the *training* augmentation of an MLP increases the test error of the untransformed dataset, validating our earlier assertions about distribution shifts induced by highly non-unitary matrices. Some models have analogous behaviour when they encounter distribution shifts in the *test* set: the MLP, MLP augmented with rotations, and so2nets share a similar trend that the test error increases as the test distribution undergoes greater transformation.

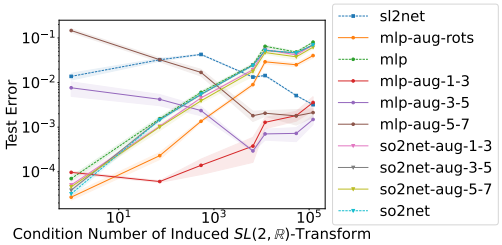


Figure 3: **Test errors.** The test dataset is transformed by random $A \in SL(2,\mathbb{R})$. The average value of the condition numbers of $A^{[d]}$ (d=6) is labeled on the x-axis. Error bars are one standard deviation over 4 independent runs.

5. Conclusion

In this work, we demonstrated for the first time the potential of machine learning for polynomial problems, particularly with equivariant learning techniques. An MLP with augmentations works best if the test and training distributions are similar, while the sl2net may only be useful to generalize very far beyond the original distribution.

As future work, it would be useful to apply a network trained on low degree to approximate high degree results, analogous to how graph neural networks apply to graphs with arbitrary number of vertices. Changing the polynomial basis we use may provide the sl2net more stability in training, while exploring polynomials in more variables or approaches to incorporate $GL(2,\mathbb{R})$ may be interesting as well. Other application areas of an $SL(2,\mathbb{R})$ equivariant architecture may also be of interest, e.g. classifying image silhouttes.

References

ApS, M. Mosek optimizer api for python. *Version*, 9(17): 6–4, 2022.

Bogatskiy, A., Anderson, B., Offermann, J. T., Roussi, M., Miller, D. W., and Kondor, R. Lorentz group equivariant neural network for particle physics. ICML'20. JMLR.org, 2020.

Bohning, C. and von Bothmer, H. G. A clebsch-gordan formula for sl3 (c) and applications to rationality. *Adv. Math*, 224:246–259, 2010.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., Riviere, M., Tran, K., Heras-Domingo, J., Ho, C., Hu, W., Palizhati, A., Sriram, A., Wood, B., Yoon, J., Parikh, D., Zitnick, C. L., and Ulissi, Z. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11 (10):6059–6072, 4 2021. doi: 10.1021/acscatal.0c04525.

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 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3165–3176. PMLR, 2020.

Henrion, D. and Garulli, A. *Positive polynomials in control*, volume 312. Springer Science & Business Media, 2005.

Mittelmann, H. D. An independent benchmarking of sdp and socp solvers. *Mathematical Programming*, 95(2): 407–430, 2003.

Parrilo, P. A. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

Parrilo, P. A. and Jadbabaie, A. Approximation of the joint spectral radius using sum of squares. *Linear Algebra and its Applications*, 428(10):2385–2402, 2008.

Popoviciu Draisma, M. I. *Invariants of binary forms*. PhD thesis, University_of_Basel, 2014.

Acknowledgements

HL is supported by the Fannie and John Hertz Foundation. MH and HL are supported by the NSF Graduate Fellowship under Grant No. 1745302. We thank Pablo Parrilo for many productive discussions.

A. Related Work

Broad application of machine learning The purview of machine learning has expanded far beyond its original wheelhouse of image and language domains, impacting diverse fields across science and algorithms (Pfau et al., 2020; Jumper et al., 2021). In scientific domains like PDE time-stepping and molecular dynamics, productive lines of work have arisen around learning to regress unknown parameters or forward-step in time — problems which are solvable by existing methods but could ideally be sped-up using large training datasets. In many such applications, machine learning has successfully provided a learnable model order reduction: given data generated by a high-accuracy, computationally expensive method, such as simulations based on density functional theory (DFT) for molecular dynamics or custom solvers for PDEs (Batzner et al., 2022; Han et al., 2018), a neural net is trained to approximate these costly solutions on a dataset of interest. Especially valuable are problems for which, one a solution has been found, its correctness can be verified efficiently; this is the case for e.g. modeling catalysts, where a candidate output conformation can be confirmed with a few DFT steps (Chanussot et al., 2021).

Equivariant Learning In other contexts of structured inputs, it is now relatively common practice to adapt neural architectures to the invariances of the input and output data types: neural networks for graph classification are invariant to permutation of the input nodes (Maron et al., 2019), for example, while architectures for point clouds are usually agnostic to translation, rotation, and permutation (Thomas et al., 2018; Anderson et al., 2019). Across these domains and many others, the incorporation of such problem-specific group symmetries into neural architectures has proven essential, lending greater sample efficiency as well as generalization to all of the appropriate symmetry transformations of the training data in both theory and practice (Cohen & Welling, 2016; Elesedy & Zaidi, 2021). A standardized toolkit has arisen for designing these "equivariant" architectures across wide families of groups, typically considering linear layers and nonlinear layers separately (Esteves, 2020). Group convolution provides one widespread method for building learnable linear equivariant layers, extending the translational convolution of traditional CNNs to more general transformations (Cohen & Welling, 2016). Indeed, such methods are readily applicable to any compact group. As an equivariant nonlinearity, a simple pointwise non-linearity often suffices for simple group transformations, while more elaborate nonlinearities based on tensor products are available for more complex group representations (Cohen et al., 2018; Kondor et al., 2018; Thomas et al., 2018).

Architectures enforcing equivariance to *compact* Lie groups abound in both theory and practice, often focusing on $SO(3)$ as it arises widely in practice. To name just a few, (Cohen et al., 2018) enforces spherical equivariance using real-space nonlinearities, while (Kondor et al., 2018) uses the tensor product nonlinearity. Several architectures designed for rotation-equivariant point cloud inputs also use tensor product nonlinearities (Thomas et al., 2018; Anderson et al., 2019). Methods for more general Lie groups based on Monte Carlo approximations of convolution integrals include Finzi et al. (2020) and MacDonald et al. (2021), while Finzi et al. (2021) presents a computational method for solving for an equivariant linear layer of an arbitrary Lie group. Since $SL(2, \mathbb{R})$ is a noncompact group, forming invariants by group averaging is impossible. As discovered in (Bogatskiy et al., 2020), if every layer activation lives in a finite-dimensional representation space, then averaging over the group is unnecessary. Our fully equvariant $SL(2, \mathbb{R})$ architecture can be viewed as a specialization of their insightful Lorentz group architecture. The exposition simplifies dramatically because our binary form inputs lie precisely in the group's finite-dimensional irreps. Note that our application area (and therefore the output layer) and empirical findings are distinct. Related to higher-level analyses of equivariances that induce distribution shift, Wang et al. (2022) coin the term "extrinsic" equivariance, and note that it can improve performance; in other settings, however (Wang et al., 2023), this kind of equivariance is not advantageous. Finally, Gerken et al. (2022) provides one other empirical example where augmentation outperforms equivariance, although they observe this phenomenon only for invariant tasks with respect to the compact group of rotations, whereas we work with a task equivariant to a non-compact group, and therefore hypothesize that the underlying mechanisms are quite different.

Polynomial problems Polynomials are ubiquitous in mathematics and engineering. Primitives such as minimizing a polynomial (Parrilo & Sturmfels, 2003; Jiang, 2013; Passy & Wilde, 1967; Nie, 2013; Shor, 1998) or certifying its positivity via a sum of squares decomposition (Lasserre, 2009; Prestel & Delzell, 2013; Parrilo, 2000), have attracted attention due to not only their abstract beauty, but also their direct applications to diverse problems ranging from operations research to control theory (Henrion & Garulli, 2005; Tedrake et al., 2010). Examples include certifying the stability of a dynamical system with a Lyapunov function, robot path planning, and designing nonlinear controllers (Ahmadi & Majumdar, 2016). There exist classical methods (ApS, 2022) for solving the resulting semidefinite program. Our polynomial nonnegativity task would be one way to accelerate the search for sum of squares certificates in any such problem because SDP solvers are only effective when the dimensions are not too large (Mittelmann, 2003). For global polynomial minimization, a comparison of abstract approaches, including sum of squares-based methods, are given in (Parrilo & Sturmfels, 2003). Methods such

as grid refinement and cutting plane algorithms can be used for a number of similar applications (Hettich & Kortanek, 1993), but the result is not certifiably feasible. One could check the feasibility by finding a certificate with our polynomial nonnegativity task.

B. Technical Background and Details

B.1. Details of $SL(2,\mathbb{R})$-equivariant architecture

Nonlinearity We first compute $p_i \otimes p_j$ for $0 \leq i, j \leq d$ within each channel. Decompose these tensor products using the linear map T from Section 3.1 to get $T(p_i \otimes p_j)$, which is a list of polynomials of degree at most $i + j$.

Learned linear layer After applying the Clebsch-Gordan decomposition T, we have a collection of polynomials ranging from degree 0 to degree $2d$. Gather all resulting polynomials of degree k resulting from any tensor product across any channel into the vector of polynomials v. Let $\#k$ be the number of such polynomials, and c the number of output channels. W is a $c \times \#k$ learnable matrix. Then, Wv are the inputs of degree k for the next layer.

Last layer Schur's Lemma then describes how to preserve equivariance, based on the decomposition of V into irreps. Because L is linear and invertible, we first compute L^{-1} given we know the decomposition of V into irreps. By linearity we only need to compute L^{-1} on a basis of V. Then we use the inverse of this transformation for L.

For the positivity verification task, we only need to design the last layer. Let v be a list of forms of degrees $0, \ldots, d$. Let $L : \bigoplus_{n=0}^{d} V(d) \to S^d$ be the last layer of the architecture. If L is equivariant and $L(v) = a \otimes b$ is rank one, we must have $L(A^{[0]^T} v_0, \ldots, A^{[2d]^T} v_{2d}) = (A^{[d]^T} L(v) A^{[d]}) = (A^{[d]^T}(a \otimes b) A^{[d]}) = (A^{[d]^T} a) \otimes (A^{[d]^T} b) = (A^{[d]^T} \otimes A^{[d]^T})(a \otimes b)$. Therefore L must be a map from irreps to the tensor product representation space. We proceed to calculate L via the procedure in Section 3.2. Let e_i be the standard ith basis vector of $\mathbb{R}^d$. The matrices $\frac{1}{2}(e_i \otimes e_j + e_j \otimes e_i)$ are a basis for the space S^d. We can interpret e_i and e_j as monomials of degree d. Then define $L^{-1} : \frac{1}{2}(e_i \otimes e_j + e_j \otimes e_i) \mapsto \frac{1}{2}(T(e_i \otimes e_j) + T(e_j \otimes e_i))$, with T from Section 3.1. Finally, we guarantee that our output Q satisfies $p = \vec{x}^{[d]^T} Q x^{[d]}$ by requiring that the degree $2d$ input to L is the original polynomial p.

B.2. Scale homogeneity and a comparison between $SO(2)$ equivariance and $SL(2)$ equivariance

In this work, we have evaluated both $SL(2)$ and $SO(2)$ equivariant architectures. Usually, for an $SL(2)$-equivariant problem, $SL(2)$ equivariance is naturally much stronger than $O(2)$ equivariance. However, the polynomial positivity verification problem explored in Section 4 has the additional structure that it is not only equivariant, but also 1-scale homogeneous: if homogeneous polynomial $p(\vec{x})$ has verifier matrix M of maximal determinant, then $cp(\vec{x})$ has verifier matrix cM of maximal determinant. (Similarly, polynomial minimization is also 1-homogeneous, although it is $SL(2)$-*invariant*, not equivariant.) The first implication of this structural property is that we may enforce it via data normalization, as discussed below:

Data normalization If a function f is 1-homogeneous, and one has an approximating function g, it is possible to make g 1-homogeneous via the following procedure:

$$\tilde{g}(x) := \|x\| g\Big(\frac{x}{\|x\|}\Big)$$

Indeed, this is the normalization we employ in all of our experiments, as data normalization (having all inputs roughly of the same scale) is a widespread and important technique to achieve good neural network performance during training. However, it is important to note that even if g is $SL(2)$-equivariant, $\tilde{g}$ may not be. If $A \in SL(2)$, then

$$\tilde{g}(Ax) = \|Ax\| g\Big(\frac{Ax}{\|Ax\|}\Big) \stackrel{?}{=} A \circ \tilde{g}(x) = A \circ \left[\|x\| g\Big(\frac{x}{\|x\|}\Big)\right] = \|x\| g\Big(\frac{Ax}{\|x\|}\Big)$$

However, $\|Ax\| \neq \|x\|$ in general; the uncertain equality does not hold unless g is already 1-homogeneous.

Even if one could parametrize the space of functions that are *both* $SL(2)$-equivariance and 1-homogeneous, one then might also reasonably ask whether $SL(2)$ and $SO(2)$ equivariance are actually that different, since scale equivariance implies that all interesting behavior in the function to be learned is captured by its behavior on the unit ball. In this section, we

expound upon the distinctions between $SO(2)$ equivariance and $SL(2)$ equivariance, even for scale-equivariant functions. Importantly, for our equivariant problem of polynomial positivity verification, the two notions are distinct — this is proven in the following proposition. Therefore, although the normalization procedure we employ slightly breaks equivariance (when there is no normalization), capturing both properties at once is a promising future direction. Moreover, as shown in subsequent plots, our architecture still captures equivariance better than other architectures.

We begin with the following example, which is our only example in which the two notions coincide. However, as explained below, it is a trivial case.

Proposition B.1 ($O(n)$ and $SL(n)$ *invariance* are equivalent for scale-invariant problems)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a scale-invariant function, i.e. $f(cx) = cf(x)$, and assume both $O(n)$ and $SL(n)$ act on $\mathbb{R}^n$. We assume G acts orthogonally. Then f is $SL(n)$-invariant if and only if it is $O(n)$-invariant.*

Proof. The statement follows from the simple fact that a function that is both scale-invariant *and* $O(n)$-invariant must be constant. This is because $O(n)$-invariance implies that $f(x)$ depends only on $\|x\|$, while scale invariance implies that $f(x)$ depends only on $\frac{x}{\|x\|}$. □

The above example is not very satisfying, however. In the next proposition, we prove that $SO(2)$ and $SL(2)$ equivariance are distinct for the polynomial positivity problem considered in the main text – even when considering 1-homogeneous functions via input normalization. Consider the type of input representation relevant for the maximum determinant problem, composed with data normalization: $p \mapsto \frac{A^{[d]}p}{\|A^{[d]}p\|}$. The following proposition states that this is not equivalent to an $SO(2)$ transformation of p.

Proposition B.2 (SO(2) *equivariance* is not equivalent to SL(2) *equivariance* and scale homogeneity)**.** *There exist p and there exists some element $A \in SL(2)$ such that there is no rotation $R \in SO(2)$ satisfying*

$$\frac{A^{[d]}p}{\|A^{[d]}p\|} = R^{[d]}p.$$

This proposition is somewhat counterintuitive. If A is $n \times n$, then for every $x \in \mathbb{R}^n$, there exists an orthogonal matrix R depending on R and x such that $\frac{Ax}{\|Ax\|} = Rx$. The reason this case is different is because even though induced matrices are size $(d+1) \times (d+1)$, they only have 4 degrees of freedom. This idea is formalized below.

Proof. Suppose for the sake of contradiction that for all $A \in SL(2)$, there is some rotation $R \in SO(2)$ and $\alpha \in \mathbb{R}$ satisfying

$$\frac{A^{[d]}p}{\|A^{[d]}p\|} = R^{[d]}p.$$

Left-multiplying both sides by $R^{[d]T}$, we obtain

$$R^{[d]T}\frac{A^{[d]}p}{\|A^{[d]}p\|} = R^{[d]T}R^{[d]}p.$$

Let $M^{[d]} = \frac{1}{\|A^{[d]}p\|}R^{T^{[d]}}A^{[d]}$. Such an M exists because $(R^TA)^{[d]} = R^{T^{[d]}}A^{[d]}$ as the induced matrix map is a homomorphism, and the map is homogeneous. So the condition is equivalent to $p(M^{-1}x) = p(x)$. For a generic p, it seems natural that there are only finitely many M such that $p(M^{-1}x) = p(x)$. In general, this is a system of $d+1$ degree d polynomial equations with 4 variables. For this proposition, however, we only need to show this system has finitely many solutions for a particular p.

In particular, we argue that if $p = x^{2d} + y^{2d}$, then there are only finitely many M. The terms other than $x^{2d} + y^{2d}$ of $(ax+by)^{2d}+(ex+fy)^{2d}$ would need to vanish. A generic coefficient in the expansion is a constant times $a^ib^{2d-i}+e^{2d-i}f^i$. Any term with even i and $2d-i$ would have no solution over the reals unless $ab = 0$ and $cd = 0$. The first and last expansion terms give us that $a^{2d} + e^{2d} = 1$ and $b^{2d} + f^{2d} = 1$. If for instance $a = b = 0$, then $e = \pm 1$ and $f = \pm 1$. There are only finitely many cases of which variables are zero to consider.

There exists some $A \in SL(2,\mathbb{R})$ such that MA^{-1} is not a multiple of an orthogonal matrix for any of these finite number of M. The matrix $\begin{pmatrix} r & 0 \\ 0 & \mathbb{R} \end{pmatrix}$ is in $SL(2,\mathbb{R})$ for every $x \in \mathbb{R}$. We will choose A so that A^{-1} is of this form. If $M = QR$ is the QR decomposition of M, and $R = \begin{pmatrix} u & v \\ 0 & w \end{pmatrix}$, then compute $-\frac{u}{v}$ for every M. Let x be much larger than any of these values. Then MA^{-1} scales vectors arbitrarily large and so is not orthogonal. □

Remark B.3. As a result of the above proposition, SO(2) and SL(2) *equivariance* are not equivalent for positivity verification. This is because representations of $SL(2,\mathbb{R})$ capture a richer variety of transformations than induced matrices of $O(2)$, even after normalization.

Although the proposition above proves that normalization and $SO(2)$-equivariance together still do not suffice to capture $SL(2)$-equivariance, the following reasoning still captures why we might expect $SO(2)$-equivariance to perform well. To approximate the $SL(2)$-equivariant, 1-homogeneous function f on an input x, first normalize x, and then input it to an $SO(2)$-equivariant architecture, and finally re-scale by $\|x\|$. (More precisely, the $SO(2)$ input representation in the architecture is merely the subrepresentation of the original $SL(2)$ input representation.) Although this technique is not $SL(2)$-equivariant, by the previous proposition, it still captures equivariance.

B.3. $SO(2)$ equivariant architecture

The main development for the so2net was making such a network compatible with the input and output types required for the tasks. The input for all tasks was a binary homogeneous polynomial of degree d.

Input layer The first layer maps the input to elements of the representation spaces of $SO(2)$. We map the homogeneous polynomial in (x, y) to a homogeneous polynomial in $(\cos\theta, \sin\theta)$ and then compute its Fourier coefficients.

Nonlinearity In the case of $SO(2)$, the tensor product simplifies immensely. Because the elements of the representation spaces of degrees j and k are scalars, their tensor product is simply their product that lives in the representation space of order $j + k$.

Learned linear layer We are free to add arbitrary linear combinations of scalars that live in the same representation spaces. These weights are learned parameters. Furthermore, we learn an MLP to apply to each set of invariants (Fourier modes of order 0). We send the results of this layer into either another nonlinear layer or into the final layer.

Final layer The final layer is output dependent. In the case of an invariant problem, the final layer should return an element of the representation space of order 0. For the problem of Section 2, we convert these Fourier modes into a sequence of irreps of $SL(2,\mathbb{R})$ and apply the final layer used for the sl2net. We convert a channel of Fourier modes to an ordinary binary form as follows. Every Fourier term of the form $a\cos(k\theta) + b\sin(k\theta)$ corresponds to a homogeneous polynomial in $\cos(\theta)$ and $\sin(\theta)$ of degree k. If $k < d$, where d is the degree of the binary form that we need, we multiply by the necessary power of $1 = \cos^2(\theta) + \sin^2(\theta)$ to lift to a form of degree d. Then we reverse-substitute (x, y) for $(\cos\theta, \sin\theta)$. This can be used as input for the last layer of the sl2net.

B.4. Universality

The construction of sl2net and so2net satisfy the requirements of Theorem D.1 of (Bogatskiy et al., 2020), which implies that any equivariant map for our tasks can be uniformly approximated by these networks. (This is expected because sl2net and so2net are in some sense special cases of their construction.) The important features of such networks are that they compute iterated tensor products of representations and they have non-polynomial activations of group invariants. The idea for universality is to show that a basis of invariant and equivariant polynomials can be generated. They argue how repeatedly taking tensor products yields this required basis.

C. Experiments

C.1. Experimental Setup

All experiments were run on Nvidia Volta V100 GPUs, using the AdaM optimizer with learning rate $3 \cdot 10^{-4}$. Roughly, training on a single GPU took 15-30 minutes for each MLP, 1.5-2.5 hours for each SL2Net, and 2.5-6 hours for each SO2Net. For runs with data augmentation, augmentations were performed using $10,000$ presaved 2×2 $SL(2)$ matrices (and their

Architecture	Hyperparameters	Parameters
MLP	hidden layers 100-1000 (dimension 100, then 1000)	117,816
SL2Net	5 layers, 50 channels, max internal deg. 12, invariant MLP 10-10	887,771
SO2Net	3 layers, 10 channels, max internal deg. 12, invariant MLP 10-10	58,110

Table 3: Architecture hyperparameters for the max determinant experiment. For multi-layer perceptron (MLP) architectures, "x-y" indicates x hidden units, followed by y hidden units, etc.

Architecture	Hyperparameters	Parameters
MLP	hidden layers 100-1000 (dimension 100, then 1000)	104,901
SL2Net	3 layers, 20 channels, max internal deg. 12, invariant MLP 10-10	163,628
SO2Net	3 layers, 10 channels, max internal deg. 12, invariant MLP 10-10	48,930

Table 4: Architecture hyperparameters for the minimization experiment. For multi-layer perceptron (MLP) architectures, "x-y" indicates x hidden units, followed by y hidden units, etc.

induced versions) with condition numbers in the specified range. Details of the distribution from which these matrices were generated, as well as from which the random augmentations were applied in our test error plots, can be found in the code.

Positivity Verification Setup Random positive degree d forms were generated as follows. A real Wigner matrix A of dimensions $(d+1) \times (d+1)$ is sampled. Each entry is an independently and identically distributed random normal variable with mean 0 and variance 1. Then $p = \vec{x}^{[d]^T}(A^T A + 10^{-8} I)\vec{x}^{[d]}$. The identity perturbation is added to ensure strict positivity. The maximum determinant Gram matrix was computed with (ApS, 2022). We used $5,000$ training examples, 500 validation examples, and 500 test examples.

Experiments were trained for 700 epochs across 4 random seeds. We used the hyperparameters shown in Table 3, which were chosen heuristically by comparing validation errors across a small number (< 20) of runs.

Minimization Setup For the purpose of computing the minimum of an inhomogeneous polynomial, the polynomial was generated as follows. Let m be sampled from a standard normal distribution. Let (x_0, y_0) each be the result of sampling independently from a standard normal and taking the absolute value. If a, b are sampled independently from a uniform distribution on $[0, 1]$, then let $(x_1, y_1) = ((2a-1)x_0, (2b-1)y_0)$. Then multiply a collection of polynomials from the five quadratics in $\{((x \pm x_0)^2 + (y \pm y_0)^2), ((x - x_1)^2 + (y - y_1)^2)\}$ to get a degree $d-2$ polynomial. Add together every product from this collection that has degree $d-2$ to get the polynomial p and then return $p \cdot ((x - x_1)^2 + (y - y_1)^2) + m$. The minimum of this polynomial is guaranteed to be m and occur at (x_1, y_1). After expanding the resulting polynomial, we can separate the polynomial into binary forms of degrees $0, \ldots, d$ and input all forms simultaneously into the first layer of the neural network. We used $5,000$ training examples, 100 validation examples, and 100 test examples.

Experiments were trained for 400 epochs across 2 random seeds. We used the hyperparameters shown in Table 4 which were chosen heuristically by comparing validation errors across a small number (< 20) of runs.

Normalization As noted in the previous section, we normalize input polynomials via $p \mapsto \frac{p}{\|p\|}$, where $\|p\|$ is equal to the L_2 norm of the polynomial's coefficients, in a monomial basis. Since our problems are 1-homogeneous, we rescale the outputs via $\|p\|$. This is done for all architectures we compare; therefore, all methods are scale equivariant (or in particular, 1-homogeneous) during training and validation.

C.2. Polynomial Minimization

In this section, we repeat our experiments for an alternative $SL(2)$-equivariant problem: polynomial minimization. If a bivariate polynomial (not necessarily homogeneous) has a unique minimum, then that global minimum is invariant to any invertible change of coordinates, including $SL(2, \mathbb{R})$. The $SL(2, \mathbb{R})$-equivariant architecture is amenable to this problem by the following adaptations.

- The input to the first layer is a sequence of forms of different degrees. The degree k form are all the terms of degree k of the original polynomial.
- The last layer returns an element of the degree 0 representation space – an invariant scalar.

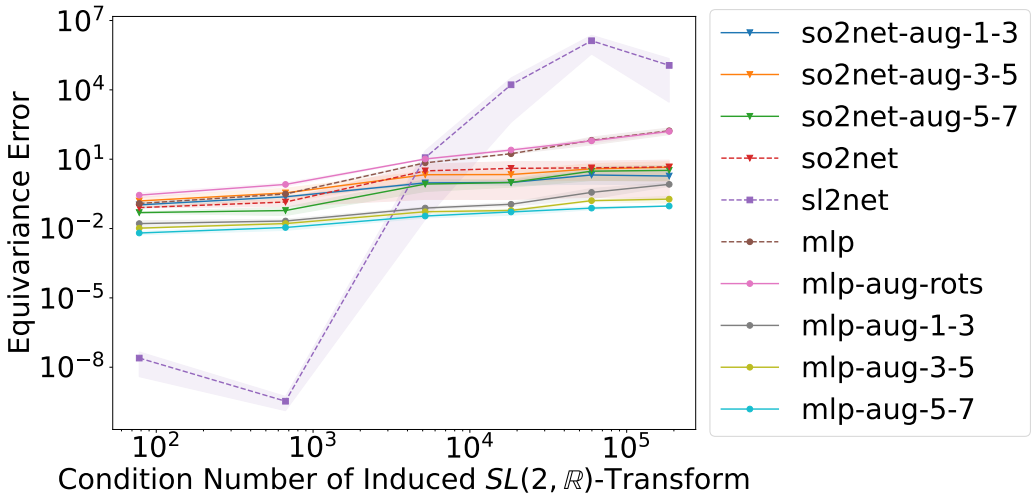


Figure 4: Equivariance errors for different architectures on the polynomial minimization problem, averaged over two random seeds with a standard-deviation error bar.

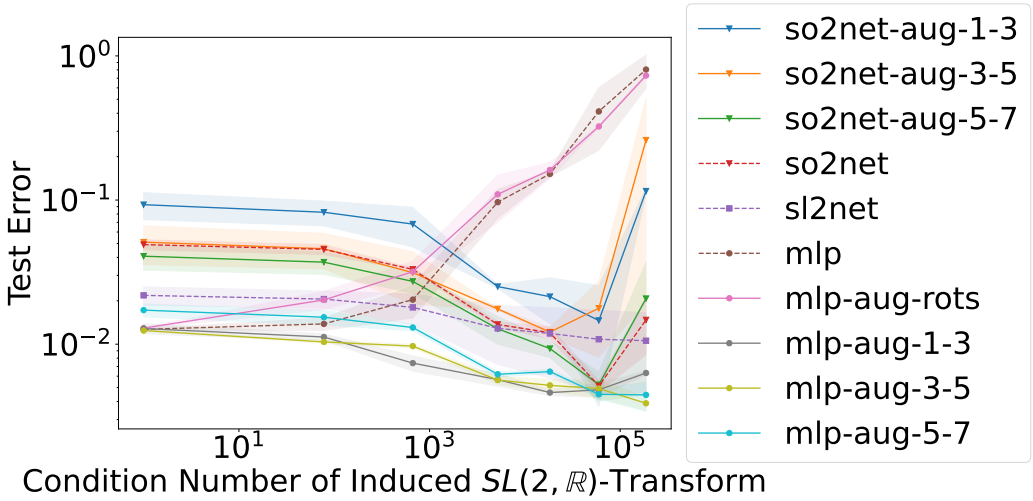


Figure 5: Normalized test errors for different architectures on the polynomial minimization problem, averaged over two random seeds with a standard-deviation error bar.

The changes to so2net are analogous:

- The input to the network is a sequence of forms of different degrees. We implement this as several input channels. We interpret each as a polynomial in $\cos(\theta)$ and $\sin(\theta)$ and compute the Fourier transform. We zero-pad the Fourier coefficients so that all channels have the same number of Fourier coefficients, and this collection of channels is the input to the first nonlinear layer.

- The output returns the 0th Fourier mode of the last layer, again an invariant scalar.

We again compare various architectures for applying machine learning to this problem. As shown in Figure 4, the SL2-equivariant architecture is far more equivariant than any other architecture at low condition numbers, but has high equivariance error at higher condition numbers. We discuss possible reasons for this at the end of this section. However, as shown in Figure 5, the so2 and sl2 equivariant architectures have an advantage over an unaugmented MLP only for very ill-conditioned $SL(2)$ transformations. Moreover, MLPs with $SL(2)$-augmentations perform the best in terms of test error, and this seems to be the best choice for this problem.

C.3. Equivariance Tests for Positivity Verification

Equivariance Error The equivariance error of the models in Table 1 after transformation of the input by random $A \in SL(2, \mathbb{R})$ with average condition number of $A^{[d]}$ given on the horizontal axis are reported in Figure 6. The equivariance error was calculated as

$$\frac{\|A^{[d']T}\mathcal{N}(p)A^{[d']} - \mathcal{N}(A^{[d]}p)\|}{\|A^{[d']T}yA^{[d']}\|},$$

averaged over all p in the test set (here, y is the correct minimum for input p).

As shown, the sl2net architecture had the highest degree of equivariance at lower transformation levels, but diverges along with other models for high condition numbers. Almost all of the MLPs also learned some level of equivariance. Controlling network normalization for the equivariant nets is an important direction for future work.

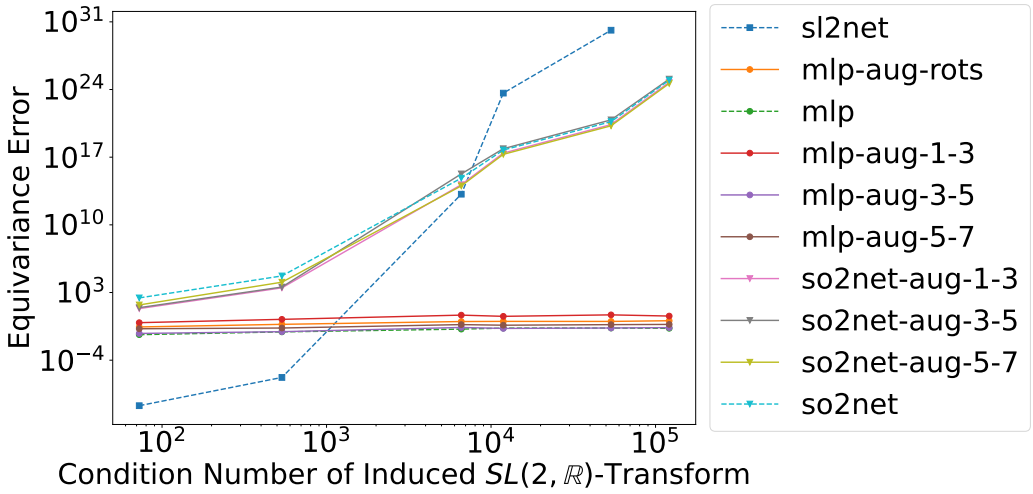


Figure 6: Equivariance errors

Impact of architecture In Figure 7, we also study the impact of the equivariant architecture hyperparameters on the equivariance error of the network. We find that the higher degree the internal irreps, the further the equivariance error is from numerical precision. In the following section, we hypothesize an explanation for this behavior.

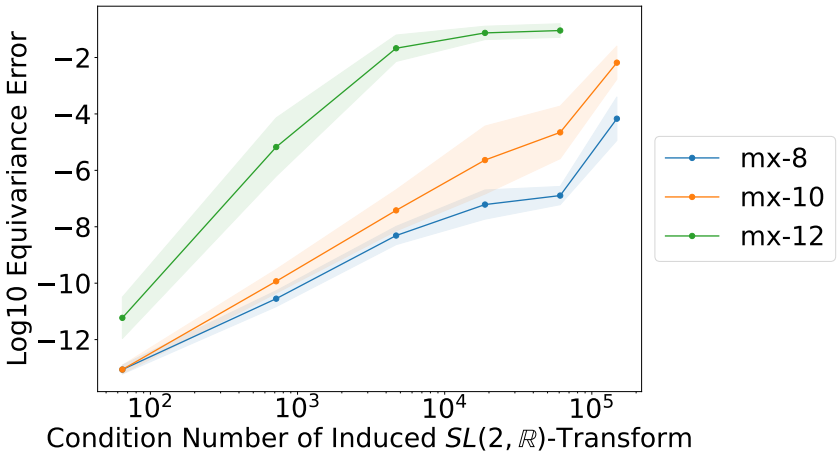


Figure 7: The impact of the $SL(2)$-equivariant architecture on equivariance error. Architectures with either channles in $\{10, 30, 50\}$, layers in $\{3, 4, 5\}$, and maximum internal irrep degree stored in $\{8, 10, 12\}$ (indicated by `mx-8`, `mx-10`, etc) are trained for a short time (30 epochs). The log equivariance error is then averaged across all architectures with the same maximum internal degree. As shown, architectures with a lower internal degree tend to be more equivariant. Another takeaway is that the variation in equivariance due to the other hyperparameters other than maximum irrep degree is minimal – changing the maximum irrep degree has the biggest affect on equivariance.

D. Conditioning considerations

D.1. Induced condition number study

First, we demonstrate in Figure 8 the exponential relationship between the condition number (denoted by κ) of A and that of $A^{[d]}$. Each line is a randomly generated matrix $A \in SL(2)$, and demonstrates a linear relationship between d and $\log(\kappa(A^{[d]}))$. This is likely a relationship that one can prove, but we defer such a theorem to future work.

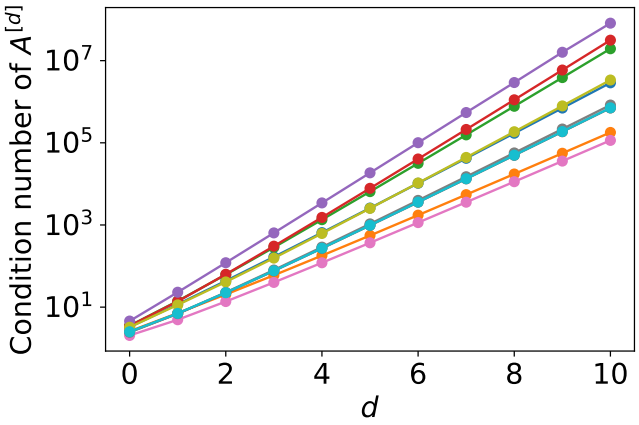


Figure 8: The condition number of $A^{[d]}$ as a function of d for 10 randomly generated matrices $A \in SL(2)$. Here, each colored line corresponds to a different sampled matrix.

D.2. What those induced condition numbers mean for us

Past work on noncompact Lie groups has largely focused on how to design linear equivariant layers, but not much on the practical difficulties that arise from working with a group of non-orthogonal matrices. Here, we expound upon some of these difficulties, in the hopes that it may be useful for future practitioners working with groups which are not subgroups of the orthogonal group.

Unlike elements of an orthogonal group, the elements of $SL(2,\mathbb{R})$ can be arbitrarily poorly conditioned. As shown in the previous subsection, Appendix D.1, even if $A \in SL(2,\mathbb{R})$ has a moderate condition number (e.g. less than 10), $A^{[10]}$ may have a condition number several orders of magnitude larger. This affects various aspects of training and testing. First, when $A^{[d]}$ is poorly conditioned, it means that computed values of $p(Ax)$ and therefore $M(p(Ax))$ may be untrustworthy. This means that there is a limit to how much data augmentation can be done, and how reasonable it is to test on transformed datasets.

Our architecture is designed to be fully equivariant. This means that it should handle arbitrary transformations of the input data. When testing equivariance of our model, it is apparent that the errors from poorly conditioned transformations compound with more Clebsch-Gordan layers and higher irrep degrees. High degree polynomials in the monomial basis are poorly conditioned, and low degree polynomials may have numerical issues when we repeatedly multiply them together. This means there is a practical limitation with how big the networks can be before the internal computations become unstable. This is consistent with Figure 7 from the previous section: networks with higher degree internal irrep activations are slighty less equivariant then those with lower degrees, as the architecture — by nature of its very equivariance — is constrained to apply an ill-conditioned matrix to its internal activations, as a result of the representation matrix applied to the input.

One could hope to choose a different basis for each vector space of homogeneous polynomials, such that the resultant induced representation is better-conditioned. This is an open problem for future work, and indeed may not always be possible for the entire group, as shown in the following general theorem about non-unitary representations.

Proposition D.1. *Let G be a group, and $p : G \to GL(V)$ a representation of G with associated n-dimensional vector space of V. A basis change $U \in O(n)$ of V induces a mapping[2] from p to p', where $p'(g)x = Up(g)U^T x$. Then, there is no basis change of p under which $\max_{g\in G} \kappa(p(g))$ changes.*

Proof. We wish to find a basis for the vector space V corresponding to a representation p such that $\max_{g\in G} \kappa(p(g))$ is as small as possible. A basis change here corresponds to the mapping $p(g) \mapsto Up(g)U^T$. Therefore, want to solve: $\min_{U\in O(n)} \max_{g\in G} \kappa(Up(g)U^T)$. If $Im(p(g))$ contains $O(n)$, then for any $U \in O(n)$, we can define h as the preimage of U: $p(h) = U$. Then, $Up(g)U^T = p(hgh^{-1})$, so $\max_{g\in G} \kappa(Up(g)U^T) = \max_{g\in G} \kappa(p(g))$ is independent of U. This implies that no basis is better than any other basis — at least from a worst-case perspective. □

Remark D.2. The standard, two-dimensional (irreducible) representation of $SL(2)$ satisfies the criterion of Proposition D.1, as $SL(2)$ contains $SO(2)$. Therefore, there is no basis under which this representation is better-conditioned – at least, in a worst-case sense over $SL(2)$.

In spite of Proposition D.1, there is still reason to believe a basis may exist that performs well in practice. First, the condition that $O(n) \subseteq Im(p(g))$ is restrictive; for representations like the induced representations $A^{[d]}$ of $G = SL(2,\mathbb{R})$ for $d > 2$, this does not hold. (Consider just that there are on the order of n free parameters for elements in $O(n)$ but only 4 for elements of $SL(2,\mathbb{R})$.) Moreover, perhaps we do not care about the metric $\max_{g\in G} \kappa(p(g))$, but rather something distributional: $\mathbb{E}_{g\sim\mu}\kappa(p(g))$. In this case, we could hope to find a basis change that is better-conditioned on high-probability group elements.

Finally, one practical consideration to do with $SL(2)$-equivariance and conditioning is the loss function. Although the L_2 norm is unchanged under orthogonal transformations — so that loss functions are often themselves invariant – the L_2 norm is changed by non-unitary representations, and therefore not invariant under our non-unitary $SL(2)$ representations. As shown below, for the positivity verification problem, the normalized loss function we use in our experiments may be distorted by a factor of up to κ^2. This means that, when we query the loss of an equivariant model on an $SL(2)$-transformed datapoint, the loss may vary in accordance with the condition number of the transformation.

Proposition D.3 (Variation of Loss Along Orbits)**.** *Let p be a homogeneous bivariate polynomial. Let $M(p)$ be the true solution to the problem in Section 2. Let $\mathcal{N}(p)$ be the equivariant neural network prediction. Then*

$$\epsilon_1 \leq \frac{\|\mathcal{N}(p) - M(p)\|_{\text{Fro}}}{\|M(p)\|_{\text{Fro}}} \leq \epsilon_2 \qquad \Longrightarrow \qquad \frac{\epsilon_1}{\kappa^2} \leq \frac{\|\mathcal{N}(gp) - M(gp)\|_{\text{Fro}}}{\|M(gp)\|_{\text{Fro}}} \leq \kappa^2\epsilon_2, \tag{1}$$

where κ is the condition number of $g^{[d]}$.

[2]We overload terminology somewhat, and refer to this as a basis change of the representation itself.

Proof. We calculate directly

$$\begin{aligned}\frac{\|\mathcal{N}(gp) - M(gp)\|}{\|M(gp)\|} &= \frac{\|g^{[d]}(\mathcal{N}(p) - M(p))g^{[d]^T}\|}{\|g^{[d]}M(p)g^{[d]^T}\|} \\ &= \frac{\|(g^{[d]} \otimes g^{[d]})\mathrm{vec}(\mathcal{N}(p) - M(p))\|}{\|(g^{[d]} \otimes g^{[d]})\mathrm{vec}(M(p))\|} \\ &\leq \frac{\sigma_{\max}(g^{[d]})^2\|\mathrm{vec}(\mathcal{N}(p) - M(p))\|}{\sigma_{\min}(g^{[d]})^2\|\mathrm{vec}(M(p))\|} \\ &\leq \kappa^2\epsilon_2.\end{aligned} \tag{2}$$

The lower bound is shown by taking the minimum singular value in the numerator and maximum in the denominator. □

Appendix Citations

Ahmadi, A. A. and Majumdar, A. Some applications of polynomial optimization in operations research and real-time decision making. *Optimization Letters*, 10:709–729, 2016.

Anderson, B., Hy, T.-S., and Kondor, R. *Cormorant: Covariant Molecular Neural Networks*. Curran Associates Inc., Red Hook, NY, USA, 2019.

ApS, M. Mosek optimizer api for python. *Version*, 9(17):6–4, 2022.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13 (1), 2022. doi: 10.1038/s41467-022-29939-5. URL https://par.nsf.gov/biblio/10381731.

Bogatskiy, A., Anderson, B., Offermann, J. T., Roussi, M., Miller, D. W., and Kondor, R. Lorentz group equivariant neural network for particle physics. ICML'20. JMLR.org, 2020.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., Riviere, M., Tran, K., Heras-Domingo, J., Ho, C., Hu, W., Palizhati, A., Sriram, A., Wood, B., Yoon, J., Parikh, D., Zitnick, C. L., and Ulissi, Z. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 4 2021. doi: 10.1021/acscatal.0c04525.

Cohen, T. S. and Welling, M. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 2990–2999. JMLR.org, 2016.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical cnns. *International Conference on Learning Representations (ICLR)*, 2018.

Elesedy, B. and Zaidi, S. Provably strict generalisation benefit for equivariant models. *arXiv preprint arXiv:2102.10333*, 2021.

Esteves, C. Theoretical aspects of group equivariant neural networks. *ArXiv*, abs/2004.05154, 2020.

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 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3165–3176. PMLR, 2020.

Finzi, M., Welling, M., and Wilson, A. G. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3318–3328. PMLR, 2021. URL http://proceedings.mlr.press/v139/finzi21a.html.

Gerken, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., and Persson, D. Equivariance versus augmentation for spherical images. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7404–7421. PMLR, 17–23 Jul 2022.

Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Henrion, D. and Garulli, A. *Positive polynomials in control*, volume 312. Springer Science & Business Media, 2005.

Hettich, R. and Kortanek, K. O. Semi-infinite programming: theory, methods, and applications. *SIAM review*, 35(3): 380–429, 1993.

Jiang, B. *Polynomial optimization: structures, algorithms, and engineering applications*. University of Minnesota, 2013.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kondor, R., Lin, Z., and Trivedi, S. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31:10117–10126, 2018.

Lasserre, J. B. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

MacDonald, L., Ramasinghe, S., and Lucey, S. Enabling equivariance for arbitrary lie groups. In *CVPR*, 11 2021.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. 2019.

Mittelmann, H. D. An independent benchmarking of sdp and socp solvers. *Mathematical Programming*, 95(2):407–430, 2003.

Nie, J. Polynomial optimization with real varieties. *SIAM Journal on Optimization*, 23(3):1634–1646, 2013.

Parrilo, P. A. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

Parrilo, P. A. and Sturmfels, B. Minimizing polynomial functions. *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60:83–99, 2003.

Passy, U. and Wilde, D. Generalized polynomial optimization. *SIAM Journal on Applied Mathematics*, 15(5):1344–1356, 1967.

Pfau, D., Spencer, J. S., Matthews, A. G., and Foulkes, W. M. C. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

Prestel, A. and Delzell, C. *Positive polynomials: from Hilbert's 17th problem to real algebra*. Springer Science & Business Media, 2013.

Shor, N. Z. *Nondifferentiable optimization and polynomial problems*, volume 24. Springer Science & Business Media, 1998.

Tedrake, R., Manchester, I. R., Tobenkin, M., and Roberts, J. W. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.

Thomas, N., Smidt, T., Kearnes, S. M., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://dblp.uni-trier.de/db/journals/corr/corr1802.html#abs-1802-08219.

Wang, D., Park, J. Y., Sortur, N., Wong, L. L. S., Walters, R., and Platt, R. The surprising effectiveness of equivariant models in domains with latent symmetry. *CoRR*, abs/2211.09231, 2022. doi: 10.48550/arXiv.2211.09231.

Wang, D., Zhu, X., Park, J. Y., Platt, R., and Walters, R. A general theory of correct, incorrect, and extrinsic equivariance. *CoRR*, abs/2303.04745, 2023. doi: 10.48550/arXiv.2303.04745.