AN ALGEBRAIC APPROACH TO APPROXIMATELY EQUIVARIANT NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Equivariant neural networks incorporate symmetries through group actions, embedding them as an inductive bias to improve performance. Prominent methods learn an equivariant action on the latent space, or design architectures that are equivariant by construction. These approaches often deliver strong empirical results but can impose architecture-specific constraints, large parameter counts, and high computational cost. We challenge the paradigm of complex equivariant architectures with a parameter-free approach grounded in representation theory. We prove that for an equivariant encoder over a finite group, the latent space must almost surely contain one copy of the regular representation for each linearly independent data orbit, which we explore with a number of empirical studies. Leveraging this foundational algebraic insight, we impose the regular representation as an inductive bias via an auxiliary loss, adding no learnable parameters. Our extensive evaluation shows that this method matches or outperforms specialized models in several cases, even those for infinite groups. We further validate our choice of the regular representation through an ablation study, showing it consistently outperforms a defining representation baseline.

1 INTRODUCTION

When we consider the problem of designing a neural network to solve a given task, we commonly observe the existence of a symmetry group G that acts naturally on the training data.[1] We illustrate a generic architecture in Figure 1, which we interpret broadly: E may be any sort of feature extractor, such as in an encoder or classifier; and D may be any final component that produces outputs from latent representations, such as a classifier head or decoder. On the input and output sets, the actions $\alpha_{\mathcal{X}}$, $\alpha_{\mathcal{Y}}$ transform the corresponding data, which we may want to be respected by our neural network.

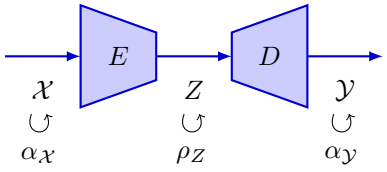

Figure 1: Generic architecture with input set $\mathcal{X}$, latent space Z and output set $\mathcal{Y}$, carrying group actions $\alpha_{\mathcal{X}}$, $\alpha_{\mathcal{Y}}$ on the input and output spaces, and potentially a representation ρ_Z on the latent space.

However, for certain tasks we can expect only *approximate equivariance*, where a transformation of the input vector corresponds inexactly, or nondeterministically, to a transformation of the outputs. This most general setup is typical of many real-world tasks, where we may encounter approximate scale-invariance or rotation-equivariance of turbulent dynamics (Holmes, 2012; Holl et al., 2020), and approximate reflection-invariance of pathologies in medical images (Yang et al., 2023).

A rich body of work in machine learning aims to learn a group representation ρ_Z that acts linearly on the latent space, satisfying a suitable equivariance property. This can be attractive, as it may reduce a complex nonlinear action on the training set to an easily-computable linear function. Furthermore, this approach has been shown to yield improved performance for invariant, equivariant, or approximately equivariant tasks (see Section 2 for a brief survey). However, the performance benefits of many of

[1] We give a formal definition of a group action on a vector space (group representation) in Section 3.

these state-of-the-art methods often come at the cost of high model complexity, increased training times, and significantly elevated parameter counts compared to their non-equivariant counterparts.

Our research is guided by the following question: **for a known finite symmetry group, can we leverage a theoretically-principled understanding of the latent algebraic structure to achieve the benefits of (approximate) equivariance, without the parameter and architectural costs of current methods?** Our core theoretical contribution is a proof that for any equivariant encoder the latent space must contain the regular representation almost surely. Based on this finding, we propose a new, lightweight training regime: we fix the latent representation to be a multiple of the regular representation, and enforce this algebraic prior with an auxiliary loss. This approach yields strong performance on a variety of invariant, equivariant, and approximately-equivariant tasks. We summarize our main contributions as follows.

- We present a new lightweight method with no additional learnable parameters for training neural networks to solve invariant, equivariant and approximately-equivariant tasks, where a finite group acts on the training set with a known action.
- We provide a theoretical characterization of latent space representations under data augmentation and an equivariant encoder, showing that the regular representation must appear almost surely. Building on this insight, we empirically validate that neural networks tend to learn a linear action aligned with this structure.
- We show that our method is competitive with or exceeds state-of-the-art in a range of benchmarks, despite having only a single tunable hyper-parameter, and no additional learnable parameters, while alternative approaches typically have large learnable parameter demands (in some cases 5-20 times baseline).

2 Related Work

A wide variety of methods have been developed to train neural networks to solve tasks in the presence of invariance, equivariance, or approximate equivariance. We give a brief summary here of those methods which are most relevant for our present work.

One of the most studied bodies of work derive from Convolutional Neural Networks (CNNs), which of course have strict translation invariance in their traditional form (LeCun & Bengio, 1998; Shorten & Khoshgoftaar, 2019). Weiler et al. employ the framework of steerable functions (Hel-Or & Teo, 1998) to construct a rotation-equivariant Steerable CNN architecture (**SCNN**) (Cohen & Welling, 2016), which strictly respects both translation and rotation equivariance; this was later generalised to develop a theory of general E(2)-equivariant steerable CNNs (**E2CNN**), which allow the degree of equivariance to be controlled by explicit choices of irreducible representation of the symmetry group (Weiler & Cesa, 2019). Such a network can avoid learning redundant rotated copies of the same filters. A similar method is that of Mobius Convolutions (**MC**) (Mitchel et al., 2022). Wang, Walters and Yu, utilise steerable filters to obtain convolution layers with approximate translation symmetry and without rotation symmetry (**RSteer**), and with approximate translation and rotation symmetry (**RGroup**) (Wang et al., 2022). These authors relax the strict weight tying of E2CNNs, replacing single kernels with weighted linear combinations of a kernel family, with coefficients that are not required to be rotation- or translation-invariant. A third approach named Probabilistic Steerable CNNs (**PSCNN**) was proposed recently by Veefkind & Cesa (2024), which allows SCNNs to determine the optimal equivariance strength at each layer as a learnable parameter. All these methods have demonstrated enhanced performance on a variety of tasks, albeit at the cost of increased training time and model complexity. They all require a significant increase in the number of learnable parameters compared to baseline (see parameter comparison in Section 6.)

We also discuss a family of approaches which are not based around the CNN architecture. Residual Pathway Priors **(RPP)**, due to Finzi et al. (2021), is a model where each layer is effectively doubled, yielding a first layer with strong inductive biases, and a second layer which is less constrained, with final output is obtained as the sum of these layers. Another architecture is Lift Expansion **(LIFT)**, which factorizes the input space into equivariant and non-equivariant subspaces, and applies different architectures to each (Wang et al., 2021).

A number of recent studies employ an equivariance term in the loss function. First we consider approaches where the symmetry group is known. Dupont et al. (2020) propose a parameter-free

method to learn equivariant neural implicit representations for view synthesis; while similar to our method in some respects, such as fixing the latent representation, their work strongly leverages the defining representation of the infinite group $O(3)$, is limited to latent spaces with the same geometrical structure as the input space, and does not apply to arbitrary latent encodings. Jin et al. (2024) present a similar method which learns non-linear group actions on the latent space using additional learnable parameters, augmented by an optional attention mechanism. In Neural Isometries (**NIso**) (Mitchel et al., 2024), the authors propose to learn an action on the latent space via its eigenbasis; in contrast, in our model the group acts linearly on the latent space with a fixed representation, and with no additional parameters needed. Other approaches that do not require the symmetry group to be known beforehand include Neural Fourier Transforms (**NFT**) (Koyama et al., 2024), which seeks to learn a suitable latent space transformation, and other work (Shakerinava et al., 2022; Winter et al., 2024).

While our work builds on these approaches, our contribution is distinct: by assuming a known symmetry, we leverage representation theory (Sections 3 and 4) to identify the regular representation as a principled and theoretically-motivated latent structure, which enables our simple pipeline without additional learnable parameters. Although our approach requires fixing a group structure, and only applies to finite groups, our experimental results show that this approach can often achieve superior performance compared to models without these constraints.

3 Background on group representations

We review essential aspects of group representation theory for our work. We consider a finite group G and work over a base field $\mathbb{K}$, assumed to be $\mathbb{R}$ or $\mathbb{C}$. The results presented are standard, for which we recommend canonical texts such as Fulton & Harris (2004) and James & Liebeck (2001). A glossary of notation and further background on group actions are available in Appendix B and C.

Regular representation. For the case of a finite group, the *regular representation* ρ_{reg} is defined as the linearisation of the action of G on itself. Explicitly, we first define $\mathbb{K}[G]$ as having elements given by linear combinations of group elements $\sum_i c_i g_i$ weighted by coefficients $c_i \in \mathbb{K}$. Then ρ_{reg} is defined as a representation on $\mathbb{K}[G]$ as follows:

$$\rho_{\text{reg}}(g)(\textstyle\sum_i c_i g_i) = \sum_i c_i (g g_i) \tag{1}$$

By construction we have $\dim(\rho_{\text{reg}}) = |G|$, the size of the group. A representation ρ on the vector space $\mathbb{K}^n$ is a *permutation representation* when for all $g \in G$, the matrix $\rho(g)$ is a permutation matrix. By construction, the regular representation is a permutation representation.

Irreducibility. Given vector spaces V and V' we can form their direct sum $V \oplus V'$, with elements which are ordered pairs of elements (v, v') of V and V' respectively. Given a representation ρ on V, and ρ' on V', we can form their *direct sum* $\rho \oplus \rho'$ acting on the vector space $V \oplus V'$, as defined as $(\rho \oplus \rho')(g)(v, v') := (\rho(g)(v), \rho'(g)(v'))$. For an integer n, we can similarly define the *n-fold multiple* of ρ, written $n \cdot \rho$, as $\rho \oplus \rho \oplus \cdots \oplus \rho$. If $\rho = \rho' \oplus \rho''$, we say that ρ' and ρ'' are *subrepresentations* of ρ.

A representation is *irreducible*, also called an *irrep*, if it is not isomorphic to a direct sum of other representations, except for itself and the zero representation. A finite group has finitely many irreducible representations up to isomorphism, and the regular representation is the direct sum of irreps, with each irrep taken with multiplicity given by its dimension. For example, the group S_3 has just the trivial (dim 1), sign (dim 2) and standard (dim 2) irreducible representations (with the same for D_3 as they are isomorphic groups); and the cyclic group C_n has n irreducible representations (all dim 1) over $\mathbb{C}$, one for each nth root of unity.

Orthogonality of representations. For a fixed group G, we may ask whether a representation ρ contains an irreducible representation ρ' as a direct summand, and if so with what multiplicity. This can be determined using the formula for *inner product of representations*:

$$\langle \rho, \rho' \rangle = \tfrac{1}{|G|} \textstyle\sum_{g \in G} \overline{\text{Tr}(\rho(g))} \text{Tr}(\rho'(g))$$

Given the knowledge of all irreducible representations of a finite group, this method allows us to determine their multiplicities as subrepresentations of ρ.

4 IDENTIFYING OPTIMAL REPRESENTATIONS

We suppose a network architecture as illustrated in Figure 1 is given, with training elements $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, and task loss $L_{\text{task}}(D(E(x_i)), y_i)$. We now suppose a finite symmetry group G is specified, which acts by fixed actions $\alpha_{\mathcal{X}}$, $\alpha_{\mathcal{Y}}$ on the input and output spaces respectively. We are interested to answer the following question: if we use additional learnable parameters to construct a third representation $\widehat{\rho}_Z$ of G on the latent space Z, which we co-train alongside the parameters for E, D with a suitable loss function, **what representation $\widehat{\rho}_Z$ does the model prefer to learn?** We first provide a theoretical analysis, which we then complement with an empirical exploration.

4.1 THE LATENT SPACE MUST CONTAIN THE REGULAR REPRESENTATION ALMOST SURELY

Adopting the notation above, we denote the G-orbit of a training sample $x \in \mathcal{X}$ as $\mathcal{O}_x := \{\alpha_{\mathcal{X}}(g)(x) \,|\, g \in G\}$. This contains all G-augmented versions of x, which we call the *data orbit of* x. We suppose x is a single data sample chosen such that all augmented versions are distinct, i.e. such that $\alpha_{\mathcal{X}}(g)(x) = \alpha_{\mathcal{X}}(h)(x)$ implies $g = h$, which is typical for data augmentation. As a consequence $|\mathcal{O}_x| = |G|$, and we conclude that G acts freely and transitively on $\mathcal{O}_x$ (nLab, 2024). Equivalently, we get that the action $\alpha_{\mathcal{X}}$ restricted to $\mathcal{O}_x$, written $\alpha_{\mathcal{X}}|_{\mathcal{O}_x}$, is isomorphic to the regular representation on $\mathcal{O}_x$. We now see that an injective equivariant encoder must preserve this representation structure when mapping the data orbit into the latent space, forcing the latent representation to contain the regular representation almost surely (proof in Appendix E).

Theorem 1. *Let G be a finite group acting on a set $\mathcal{A}$ with action $\alpha_{\mathcal{A}}$, and on a vector space Z with a representation ρ_Z, with $\dim(Z) \geq |G|$. Suppose that the group acts freely and transitively on some subset $\mathcal{S} \subseteq \mathcal{A}$. If $E : \mathcal{A} \to Z$ is an equivariant function which is injective on $\mathcal{S}$, then Z contains the regular representation almost surely.*[2]

The degree to which the latent representation faithfully instantiates the group structure depends critically on the capacity of the latent space $\dim(Z)$. When the latent space has capacity less than the size of the group, a representational collapse is unavoidable, forcing the representation ρ_Z to be a quotient of the representation. But when the latent space is large enough, such a quotient is non-generic, occurring only if the encoder's parameters are confined to a submanifold of Lebesgue measure zero. Therefore, a generic encoder with sufficient capacity will learn a representation that contains the full non-degenerate regular representation as a subspace. This principle extends across the training set: if the G-orbits $\mathcal{O}_{x_i}$ of multiple samples x_i are embedded into linearly independent subspaces by the encoder, each orbit will contribute a distinct copy of the regular representation.

Key theoretical insight: To achieve encoder equivariance in the presence of data augmentation, a sufficiently large latent space must contain a separate copy of the regular representation for each training sample with a linearly independent embedded orbit.

The question remains how many copies of the regular representation one obtains in practice, and we investigate this with the following empirical studies.

4.2 EMPIRICAL EXPLORATION

For our empirical investigation, we conduct experiments with the following loss function:

$$
\begin{aligned}
L_{\text{opt}} = {} & L_{\text{task}}(D(E(x_i)), y_i) && (i)\\
& + \lambda_t \, L_{\text{task}}\big(D(\widehat{\rho}_Z(g)(E(x_i))), \alpha_{\mathcal{Y}}(g)(y_i)\big) && (ii)\\
& + \lambda_e \, \text{MSE}\big(\widehat{\rho}_Z(g)(E(x_i)), E(\alpha_{\mathcal{X}}(g)(x_i))\big) && (iii)\\
& + \lambda_a(\text{ALG}_{G,d} + \text{REG}_{G,d}) && (iv)
\end{aligned}
$$

We motivate this as follows. Component (i) (the task loss) ensures that E, D are appropriately trained for the underlying task. Component (ii) (the equivariance loss between ρ_Z and $\alpha_{\mathcal{Y}}$) encourages E, D to be equivariant with respect to the learned representation $\widehat{\rho}_Z$ on the latent space and the fixed action $\alpha_{\mathcal{Y}}$ on the output space. Component (iii) (the equivariance loss between $\alpha_{\mathcal{X}}$ and ρ_Z) encourages E

[2] In our case, we can take $\mathcal{A}$ as the training set $\mathcal{X}$, $S = \mathcal{O}_x$ for suitable $x \in \mathcal{X}$, and Z as the latent space.

to be equivariant with respect to the fixed action $\alpha_{\mathcal{X}}$ on the input space and the learned representation $\widehat{\rho}_Z$ on the latent space. This component uses MSE rather than the task loss function, since it is defined on feature vectors in the latent space, rather than the output space. Component (iv) (the algebra loss) is a penalty term which packages together the necessary algebraic properties for $\widehat{\rho}_Z$ to become a high-quality representation of G; it includes a first part $\mathrm{ALG}_{G,d}$ arising from the group presentation, and a second regularisation term $\mathrm{REG}_{G,d}$. We give further insight into the algebra loss and additional details in Appendix D and F.

Suppose we are able to train a good solution with respect to L_{opt}, which in particular achieves good scores on component (iv), so that $\widehat{\rho}_Z$ is a true group-theoretic representation to high accuracy. We may then use techniques from representation theory to analyze what representation the model has learned. We explore this in toy settings with the MNIST (Deng, 2012), TMNIST (Magre & Brown, 2022) and CIFAR10 (Krizhevsky, 2009) datasets, for both autoencoder and classifier tasks, and for the groups C_2, D_3 and C_4. As this procedure is architecture-agnostic, we are able to use both CNN- and MLP-based architectures.

Drawing insight from Theorem 1, we expect to learn $\widehat{\rho}_Z$ that contains copies of the regular representation of the corresponding group. The number of copies is lower bounded by the number of linearly independent embedded data orbits, which must be empirically determined. In fact, the following studies demonstrate that the network prefers to learn a representation which consists *entirely* of linearly independent copies of the regular representation.

Table 1: Left, TMNIST autoencoder task, learned representations of C_2 on latent space. Right, MNIST autoencoder task, learned representations of D_3 on latent space.

	Irrep. counts				
Run	-1	$+1$	Alg. loss	Eq. loss	Orbs.
1	4	4	5.7×10^{-5}	4.6×10^{-3}	4
2	3	5	6.7×10^{-9}	6.6×10^{-6}	3
3	4	4	2.7×10^{-8}	2.5×10^{-5}	4
4	4	4	2.3×10^{-9}	4.2×10^{-6}	4
5	3	5	6.0×10^{-9}	1.9×10^{-5}	3

	Irrep. counts					
Run	Triv	Sgn	Std	Alg. loss	Eq. loss	Orbs.
1	3.01	3.01	5.99	1.2×10^{-3}	1.3×10^{-2}	3
2	2.98	2.98	6.01	6.1×10^{-4}	2.3×10^{-2}	3
3	3.32	3.36	5.66	3.1×10^{-2}	1.4×10^{-2}	3
4	3.03	3.31	5.69	1.4×10^{-2}	1.2×10^{-2}	3
5	2.98	2.98	6.02	8.5×10^{-4}	1.3×10^{-2}	3

4.2.1 TMNIST AUTOENCODER, CNN ARCHITECTURE, $G = C_2$

For our first experiment we use the TMNIST dataset, of digits rendered in a variety of typefaces. We choose a subset of two typefaces only, producing 20 images, augmenting with 180° rotations. For our group we choose $G = C_2$ presented as $\{1, a \,|\, a^2 = 1\}$. Since this is an autoencoder we have $\mathcal{X} = \mathcal{Y}$, and we choose $\alpha_{\mathcal{X}} = \alpha_{\mathcal{Y}}$, with the nontrivial element $\alpha_{\mathcal{X}}(a) = \alpha_{\mathcal{Y}}(a)$ acting to flip the choice of font, with rotation and scaling left invariant. For the algebra loss component (iv) we choose $\mathrm{ALG}_{C_2,d} = \mathrm{MSE}(\widehat{\rho}_Z(a)^2, \mathrm{I}_d)$ where $d = \dim(Z) = 8$.

We present our findings in Table 1, with each run giving one row of the table, and we give a visualisation in Figure 2. Low values in the algebra loss and equivariance loss columns show that we learn high-quality representations $\widehat{\rho}_Z$, which are strongly equivariant with respect to the representations $\alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}$. By mapping the eigenvalues of $\widehat{\rho}_Z(a)$ to the nearest value in $\{-1, +1\}$, we can determine the corresponding irreducible representation. For the group C_2 the regular representation contains exactly one copy of the -1 and +1 representations, and so we see from inspection that our learned representations are close to a multiple of the regular representation. Furthermore, we report the number of linearly independent embedded orbits, and observe that as expected this agrees with the number of copies of the regular representation which are present (see Section 4.1).

4.2.2 MNIST AUTOENCODER, MLP ARCHITECTURE, $G = D_3$

For our second experiment we choose the MNIST dataset of handwritten digits, augmented by arbitrary rotations. We choose the group $G = D_3$, the group of symmetries of an equilateral triangle with the generators r, s (120-degree rotation, flip) and the following presentation: $\{e, r, r^2, r^3, s, rs \,|\, r^3 = e, s^2 = e, rsrs = e\}$. We parameterize the linear maps $\widehat{\rho}_Z(r)$ and $\widehat{\rho}_Z(s)$ independently, and define the following algebra loss, where $d = \dim(Z) = 18$, and where summands

x $\rho_{\mathcal{X}}(a)(x)$ $D(E(x))$ $D\big(\widehat{\rho}_Z(a)\,E(x)\big)$ $D\big(\widehat{\rho}_Z(a)^2\,E(x)\big)$

Figure 2: Visualisation of our learned encoder E, decoder D and latent action $\widehat{\rho}_Z$ on input vector x with a non-geometric action. The algebraic loss correctly enforced $\widehat{\rho}_Z(a)^2 = I_d$.

correspond to constraints in the presentation:

$$\begin{aligned}\mathrm{ALG}_{D_3,d} = \mathrm{MSE}(\widehat{\rho}_Z(r)^3, \mathrm{I}_d) + \mathrm{MSE}(\widehat{\rho}_Z(s)^2, \mathrm{I}_d) \\ + \mathrm{MSE}(\widehat{\rho}_Z(r)\widehat{\rho}_Z(s)\widehat{\rho}_Z(r)\widehat{\rho}_Z(s), \mathrm{I}_d)\end{aligned}$$

For the nonabelian group D_3, we determined the learned representation's composition using orthogonality of characters (Section 3), as eigenvalues alone are insufficient for identification. The data in Table 1 confirms that the network learns a high-fidelity multiple of the regular representation, which contains the trivial, sign, and standard irreducible representations in the ratio 1:1:2. Consistent with the previous experiment, each linearly independent data orbit contributes one distinct copy of this representation. Furthermore, Figure 3 illustrates the eigenvalues of the generator $\widehat{\rho}_Z(r)$ dynamically clustering around the third roots of unity during training, despite an uneven initialization.

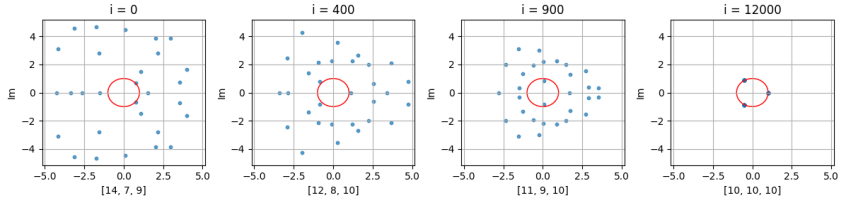

Figure 3: Complex eigenvalues of the real–valued matrix $\widehat{\rho}_Z(r)$ at different training steps i. Beneath each plot we show counts of eigenvalues nearest to each third root of unity.

4.2.3 CIFAR10 CLASSIFIER, CNN ARCHITECTURE, $G = C_4$

This experiment uses the CIFAR10 image dataset (Krizhevsky, 2009). We choose the group $G = C_4$ of 90-degree rotations, with the algebraic loss function $\mathrm{ALG}_{C_4,d} = \mathrm{MSE}\left(\widehat{\rho}_Z(1)^4, \mathrm{I}_d\right)$, where $d = \dim(Z) = 16$. For C_4 the regular representation contains exactly one copy of the $+1$, $+i$, -1 and $-i$ representations, and the results in Table 2 show that the network learns a representation which is close to a multiple of the regular representation. Furthermore, each linearly independent embedded data orbit contributes a distinct copy of this representation.

Table 2: CIFAR classifier task, representations of C_4 learned on latent space.

	Irreducible counts						
Run	$+1$	$+i$	-1	$-i$	Alg. loss	Eq. loss	Orbs.
1	4	4	4	4	1.5×10^{-4}	1.8×10^{-3}	4
2	3	4	5	4	7.2×10^{-5}	1.9×10^{-3}	3
3	3	5	3	5	9.4×10^{-5}	1.6×10^{-3}	3
4	4	4	4	4	1.1×10^{-4}	1.9×10^{-3}	4
5	4	4	4	4	8.4×10^{-5}	1.9×10^{-3}	4

Considering these three experiments together, we summarize the results of this section follows.

Key empirical insight: To achieve encoder equivariance in the presence of data augmentation, the network prefers to learn a multiple of the regular representation on the latent space.

5 METHOD

We present a novel parameter-free method to improve performance of neural networks to solve a variety of invariant, equivariant or approximately equivariant tasks, where a finite group G acts on the input and output layers with representations $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ respectively. Inspired by the theoretical and empirical results of Section 4, **instead of learning a representation on the latent space, we now fix ρ_Z to be a multiple of the regular representation of G.** Specifically, we use n copies where n is the maximum number of representations allowed by $\dim Z$. When $n|G| < \dim(Z)$, we pad by taking the direct sum with additional copies of the trivial representation, to ensure our representation on Z has the correct dimension. Our proposed representation is therefore given by:

$$\rho_Z := n \cdot \rho_{\text{reg}} + \max(\dim(Z) - n|G|, 0) \cdot \rho_{\text{triv}} \tag{2}$$

When the latent space is geometrically structured, for example as a product of features and channels, we choose an isomorphic form of the regular representation that preserves this structure (examples are the SMOKE and SHREC experiment in Section 6). We then train according to the following objective function, where $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ is an element of the training set, $g \in G$ is a group element, and $L_{\text{task}}(x_i, y_i)$ is the original task loss function:

$$\begin{aligned}
&\tfrac{1}{2}\, L_{\text{task}}\big(D(E(x_i)), y_i\big) && \text{Task loss} \\
&+ \tfrac{1}{2}\, L_{\text{task}}\big(D(E(\alpha_{\mathcal{X}}(g)x_i)),\ \alpha_{\mathcal{Y}}(g)y_i\big) && \text{Task loss shifted by } g \\
&+ \lambda\, \text{MSE}\big(E(\alpha_{\mathcal{X}}(g)x_i),\ \rho_{\mathcal{Z}}(g)\, E(x_i)\big) && \text{Equivariance loss from input to latent}
\end{aligned}$$

When used in a training loop, we select (x_i, y_i) and g uniformly at random. Here λ is a hyperparameter expressing the strength of the equivariance loss. We provide a sensitivity analysis for λ in Appendix H, which shows that model performance is robust across a range of values. Our model has no additional learned parameters above baseline, since the representation ρ_Z is now fixed. Our use of the g-shifted task loss means that our training dataset must be augmented by the action of G. This can be done either on-the-fly, or pre-computed to speed up training.

6 EXPERIMENTS

We benchmark our method against a variety of state-of-the-art methods for networks with approximate equivariance, considering four distinct tasks. We compare our results against the models SCNN, E2CNN, LIFT, RPP, RGroup, RSteer, PSCNN, NIso, NFT and MC, discussed in Section 2. All our experiments follow the setup of the original papers as far as possible. Notably, our method trains using a computational budget and wall-clock time at or below that of competing models. To ensure a fair comparison given our reliance on data augmentation, we provide both augmented and unaugmented CNN baselines. Full technical details for all reported runs, including hyperparameter selection and a sensitivity analysis for our equivariance coupling strength λ, are reported in the technical appendix. In the majority of cases our results are improved or comparable with state-of-the-art, while using fewer parameters and a simpler architecture.

For two experiments, as an ablation, we also report a comparison that replaces the regular representation in Equation 2 with the *defining representation*, the natural geometric action of the group by permutations (see Appendix C for a formal definition). The results further demonstrate the optimality of the regular representation. Further details can be found in the Appendix.

6.1 CLASSIFICATION TASK, DDMNIST, $G = C_2, C_4, D_4$

Following closely the procedure of (Veefkind & Cesa, 2024), for each of the chosen symmetry groups C_2, C_4 and D_4, we randomly and independently transform two MNIST images according to the group. Results are shown in Table 3. Because the transformations are local and independent, we apply our method using the product group. While for the groups C_2 and C_4 the regular and defining representation are isomorphic, for D_4 they are not, with the regular representation being more performant; this provides further empirical evidence for the optimality of the regular representation. Except for SCNN, we re-trained and re-evaluated all models. Further discussion is in the appendix.

Table 3: Higher is better. DDMNIST test accuracies. Mean over 3 runs; standard deviation in brackets. Parameter counts shown. Best result in each column is bold, second-best is underlined. For C_2, C_4 the defining representation is equivalent to the regular representation and so is omitted.

Model	C_4	Par M	C_2	Par M	D_4	Par M
CNN	0.907 (0.004)	**0.03**	<u>0.938</u> (0.006)	**0.03**	0.800 (0.001)	**0.03**
SCNN	0.484 (0.008)	<u>0.12</u>	0.474 (0.003)	**0.03**	0.431 (0.010)	<u>0.15</u>
Restriction	<u>0.914</u> (0.007)	<u>0.12</u>	0.890 (0.007)	0.33	0.837 (0.020)	0.17
RPP	0.908 (0.022)	0.79	0.903 (0.009)	0.08	0.827 (0.020)	1.73
PSCNN	0.909 (0.007)	0.51	0.871 (0.016)	0.04	<u>0.842</u> (0.011)	1.23
Defining rep	n/a		n/a		0.838 (0.010)	**0.03**
Ours (regular)	**0.915** (0.004)	**0.03**	**0.947** (0.004)	**0.03**	**0.868** (0.002)	**0.03**

6.2 Classification Task, MedMNIST3D, $G = \mathrm{Sym}_{\mathrm{cube}}$

We test our method on the Organ, Synapse and Nodule subsets of the MedMNIST3D dataset, using the same setup as the original authors (Veefkind & Cesa, 2024). We apply the group $\mathrm{Sym}_{\mathrm{cube}}$ of orientation-preserving symmetries of the cube, which is isomorphic to the permutation group S_4. All results, except for ours and the augmented CNN, are imported from the original authors. Table 4 shows classification accuracies on MedMNIST3D for different models and groups. For Nodule and Synapse, our method is comparable or outperforms other architectures, while having fewer parameters. We also note that the regular representation consistently outperforms the defining representation, providing further empirical evidence for its optimality. For the Organ dataset, canonical orientation is a key feature, and so the symmetry action to some extent conflicts with the task. This may explain our method's underperformance in this task (shared by the augmented CNN baseline).

Table 4: Higher is better. MedMNIST3D test accuracies. Mean over 3 runs; standard deviation in brackets. Parameter counts shown. Best result in each column is bold, second-best is underlined.

Group	Model	Nodule	Synapse	Organ	Par M
N/A	CNN	0.873 (0.005)	0.716 (0.008)	0.920 (0.003)	<u>00.19</u>
Aug	CNN	0.879 (0.007)	0.761 (0.008)	0.632 (0.005)	<u>00.19</u>
SO(3)	SCNN	0.873 (0.002)	0.738 (0.009)	0.607 (0.006)	**00.13**
SO(3)	RPP	<u>0.801</u> (0.003)	0.695 (0.037)	<u>0.936</u> (0.002)	18.30
SO(3)	PSCNN	0.871 (0.001)	**0.770** (0.030)	0.902 (0.006)	04.17
O(3)	SCNN	0.868 (0.009)	0.743 (0.004)	0.902 (0.006)	<u>00.19</u>
O(3)	RPP	0.810 (0.013)	0.722 (0.023)	**0.940** (0.006)	29.30
O(3)	PSCNN	0.873 (0.008)	<u>0.769</u> (0.005)	0.905 (0.004)	03.51
$\mathrm{Sym}_{\mathrm{cube}}$	Defining rep	0.837 (0.013)	0.756 (0.019)	0.560 (0.025)	<u>00.19</u>
$\mathrm{Sym}_{\mathrm{cube}}$	Ours (regular)	**0.887** (0.005)	**0.770** (0.002)	0.642 (0.056)	<u>00.19</u>

6.3 Autoregression Task, SMOKE, $G = C_4$

We evaluate our method on the SMOKE dataset, generated with PhiFlow (Holl et al., 2020) by Wang et al. (2022) (see Figure 6 for a visualisation). The task involves predicting future frames of a simulated smoke velocity field autoregressively. This task is only approximately equivariant to the symmetry group C_4 (90-degree rotations) due to the presence of non-equivariant buoyancy effects. Full details are provided in the appendix. Table 5(a) shows the test RMSE for each model on the metrics considered. All reported figures are imported from the original authors (Wang et al., 2022), except for ours, augmented CNN, and non-augmented CNN, for which we tune the learning rate. Our method outperforms all models except for PSCNN, which has slightly better scores, with more than 12 times the number of parameters. While our method uses the augmented training set, we note from comparing the two CNN baselines that this gives little advantage for this task.

Table 5: Mean over 3 runs; standard deviation in brackets. Parameter counts shown. Best result in each column is bold, second-best is underlined.

(a) Lower is better. Test RMSE for SMOKE dataset.

Group	Model	Future	Domain	Par M
N/A	CNN	0.81 (0.01)	0.63 (0.00)	**0.25**
Aug	CNN	0.83 (0.03)	0.67 (0.06)	**0.25**
N/A	MLP	1.38 (0.06)	1.34 (0.03)	8.33
C4	E2CNN	1.05 (0.06)	0.76 (0.02)	<u>0.62</u>
C4	RPP	0.96 (0.10)	0.82 (0.01)	4.36
C4	Lift	0.82 (0.01)	0.73 (0.02)	3.32
C4	RGroup	0.82 (0.01)	0.73 (0.02)	1.88
C4	RSteer	0.80 (0.00)	0.67 (0.01)	5.60
C4	PSCNN	**0.77** (0.01)	**0.57** (0.00)	3.12
C4	Ours	<u>0.78</u> (0.01)	<u>0.61</u> (0.01)	**0.25**

(b) Higher is better. Test accuracy for SHREC '11 dataset.

Model	Acc.
NIso Mitchel et al. (2024)	<u>90.26</u> (1.27)
NFT Koyama et al. (2024)	83.24 (2.03)
AE with aug	69.36 (2.81)
MC Mitchel et al. (2022)	86.5
Ours	**90.45** (2.1)

6.4 Autoencoding Task, 3D shapes, $G = O_h$

Finally, we test our method on the conformally transformed SHREC '11 dataset (Lian et al., 2011; Mitchel et al., 2022), following the pre-training and fine-tuning procedure of Mitchel et al. (2024). We use our parameter-free methodology with O_h augmentations (octahedral symmetries) to pre-train a baseline autoencoder before fine-tuning the encoder for classification. As this is an autoencoding task, we symmetrize the equivariance loss to the decoder. While we did not have access to the other models, we could check that NIso's kernel adds 18k parameters above our model, which has the same parameter count as the baseline autoencoder (AE). Results are given in Table 5(b). Our approach achieves 90.45% accuracy, outperforming the group-agnostic method NFT. Our method also surpasses NIso, a model capable of learning actions of infinite groups, even though our method uses only a finite subgroup.

7 Conclusions

Limitations and Future Work. Our theoretical framework is developed for finite groups, and a direct extension to infinite groups is a non-trivial challenge for future work. However, we empirically demonstrate that this theoretical limitation does not necessarily restrict the practical applicability of our method to tasks with continuous symmetries, noting that our method outperforms NIso, a state-of-the-art model capable of handling infinite groups, on the SHREC '11 dataset which involves continuous conformal transformations (Section 6.4). This strategy of employing a subgroup also directly addresses another potential limitation: tasks involving large finite groups, such as permutation groups, whose order may exceed the available latent space dimension. Our method requires data augmentation, although this is typically inexpensive when the group action on the input space is easy to construct, as for our demonstration tasks. We would also like to explore how our methodology could enable data augmentation in the latent space, while requiring augmentation of only a subset of the data at training time.

Conclusions. This work investigates an alternative path to building efficient equivariant models, focusing not on architectural design, but on the enforcement of a principled latent algebraic structure. We prove that for finite groups, this structure is the regular representation, which must appear almost surely in the latent space of any equivariant encoder. By enforcing this structure via a parameter-free auxiliary loss, our method achieves competitive or superior performance to state-of-the-art models, while requiring in some cases significantly fewer parameters. Furthermore, we empirically show the optimality of the regular representation by comparing it with the defining permutation representation. Ultimately, our work suggests that for tasks with inherent (approximate) symmetry, directly enforcing the correct latent algebraic structure can be a more effective and efficient path to equivariance than designing complex, highly-parameterized architectures.

Code of Ethics and Reproducibility Statement. We acknowledge and adhere to ICLR's code of ethics. We provide full reproducibility details in the appendix, which includes a link to our code

containing exact commands to reproduce our results. Full mathematical details are also given in the appendix.

REFERENCES

Taco S. Cohen and Max Welling. Steerable CNNs, 2016.

Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Emilien Dupont, Miguel Angel Bautista, Alex Colburn, Aditya Sankar, Carlos Guestrin, Josh Susskind, and Qi Shan. Equivariant neural rendering, 2020.

Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 30037–30049. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fc394e9935fbd62c8aedc372464e1965-Paper.pdf.

William Fulton and Joe Harris. *Representation Theory*. Springer New York, 2004. ISBN 9781461209799. doi: 10.1007/978-1-4612-0979-9.

Yacov Hel-Or and Patrick C Teo. Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9:83–95, 1998.

Philipp Holl, Vladlen Koltun, Kiwon Um, and Nils Thuerey. PhiFlow: A differentiable PDE solving framework for deep learning via physical simulations. In *NeurIPS workshop*, volume 2, 2020.

Philip Holmes. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 2012.

Gordon James and Martin Liebeck. *Representations and Characters of Groups*. Cambridge University Press, 2001. ISBN 9780511814532. doi: 10.1017/cbo9780511814532.

Yinzhu Jin, Aman Shrivastava, and Tom Fletcher. Learning group actions on latent representations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=HGNTcy4eEp.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Masanori Koyama, Kenji Fukumizu, Kohei Hayashi, and Takeru Miyato. Neural Fourier Transform: A General Approach to Equivariant Representation Learning, February 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Yann LeCun and Yoshua Bengio. *Convolutional networks for images, speech, and time series*, pp. 255–258. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262511029.

Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen. SHREC'11 track: Shape retrieval on non-rigid 3D watertight meshes. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*, 3DOR '11, pp. 79–88, Goslar, DEU, April 2011. Eurographics Association. ISBN 978-3-905674-31-6.

Nimish Magre and Nicholas Brown. Typography-MNIST (TMNIST): an MNIST-style image dataset to categorize glyphs and font-styles, 2022.

Thomas W. Mitchel, Noam Aigerman, Vladimir G. Kim, and Michael Kazhdan. Möbius Convolutions for Spherical CNNs, May 2022.

Thomas W. Mitchel, Michael Taylor, and Vincent Sitzmann. Neural Isometries: Taming Transformations for Equivariant ML, October 2024.

nLab. Free action, 2024. URL https://ncatlab.org/nlab/show/free+action. Accessed: 2025-08-02.

Mehran Shakerinava, Arnab Kumar Mondal, and Siamak Ravanbakhsh. Structuring Representations Using Group Invariants. In *Advances in Neural Information Processing Systems*, October 2022.

Christopher Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:60, 2019. doi: 10.1186/s40537-019-0197-0.

Lars Veefkind and Gabriele Cesa. A probabilistic approach to learning the degree of equivariance in steerable CNNs. In *41st International Conference on Machine Learning (ICML 2024)*, 2024. URL https://openreview.net/forum?id=49vHLSxjzy.

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant Q learning in spatial action spaces. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=IScz42A3iCI.

Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pp. 23078–23091. PMLR, 2022.

Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Robin Winter, Marco Bertolini, Tuan Le, Frank Noé, and Djork-Arné Clevert. Unsupervised Learning of Group Invariant and Equivariant Representations, April 2024.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1), January 2023. ISSN 2052-4463. doi: 10.1038/s41597-022-01721-8.

A CODE

The code to run all the experiments in this paper is available at the following location:

- https://anonymous.4open.science/r/parameter-free-approximate-equivariance-3352/

In the README file, we provide instructions to run the code and reproduce the results.

B NOTATION

Here we provide a comprehensive list of symbols and notational conventions used throughout the paper.

GENERAL MATHEMATICAL OBJECTS

G A finite group.

g, h Elements of the group G, e.g., $g \in G$.

$\mathbb{K}$ The base field, assumed to be either the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$.

$\mathcal{S}, \mathcal{A}$ General sets, denoted by calligraphic letters.

V, W General vector spaces, denoted by uppercase Roman letters.

v, w Elements (vectors) of a vector space, e.g., $v \in V$.

GROUP THEORY

α A group action on a set. The action of $g \in G$ on an element $s \in \mathcal{S}$ is written as $\alpha(g, s)$

ρ_V A group representation on the vector space V, which is a linear group action on V.

$\rho_V(g)$ The invertible linear map associated with the group element $g \in G$. The action of g on a vector $v \in V$ is written as $\rho_V(g)(v)$.

MACHINE LEARNING CONTEXT

$\mathcal{X}$ The input set.

x A single input data point, $x \in \mathcal{X}$.

$\mathcal{Y}$ The output or label set.

y A single output or label, $y \in \mathcal{Y}$.

Z The latent space, viewed as a vector space (e.g., $Z = \mathbb{R}^d$).

z A latent vector, $z \in Z$.

E An encoder network.

D A decoder network.

$\widehat{\rho}_Z$ A *learnable* representation on the latent space Z.

C GROUP ACTIONS AND REPRESENTATIONS

Groups. A group G is a set equipped with an associative and unital binary operation, such that every element has a unique inverse. Important families of groups include the following. The dihedral group D_n is the group of symmetries of the regular polygon with n sides, which we use in this paper for $n \geq 3$. The cyclic group C_n is the groups of integers $\{0, \ldots, n-1\}$ with addition modulo n. The permutation group S_n is the group of permutations of an n-element set. We may define groups by presentations, which give generators and relations for the product; for example, the group C_2 can be defined by the presentation $\{1, a \,|\, a^2 = 1\}$. For any two groups G, H, we write $G \times H$ for the product group, whose elements are ordered pairs of elements of G and H respectively.

Group representations. A *representation* ρ of a finite group G on a vector space V is a choice of linear maps $\rho(g) : V \to V$ for all elements $g \in G$, with the property that $\rho(e) = \mathrm{id}_V$ for the identity element $e \in G$, and such that $\rho(g)\rho(g') = \rho(gg')$ for all pairs of elements $g, g' \in G$. We define the *dimension* of ρ to be $\dim(V)$, the dimension of the vector space V. There is a notion of equivalence of representations: given representations ρ on V, and ρ' on V', they are *isomorphic* when there is an invertible linear map $L : V \to V'$ such that $L\rho(g) = \rho'(g)L$ for all $g \in G$. Given a subgroup $G \subseteq G'$, a representation of G' yields a *restricted representation* on G in an obvious way.

Defining representations. The concept of a defining representation is relevant for our ablation studies. While the term is context-dependent, it typically refers to a group's most natural or defining low-dimensional representation. For the permutation group S_n this is the linearisation of its permutation action on the n-element set; that is, the n-dimensional representation given by its action on $\mathbb{K}^n$ by permuting the basis vectors. For the dihedral group D_n ($n \geq 3$), the defining representation is the linearisation of its action on the n-element set of vertices. For the group $\mathrm{Sym}_{\mathrm{cube}}$ of orientation-preserving symmetries of the cube, the defining representation is the linearisation of its action on the 8-element set of vertices of the cube. We select these defining representations as a baseline as they provide a rich, geometrically intuitive alternative to the more abstract regular representation.

Group actions. A group may also have an *action* λ on a set $\mathcal{S}$, a choice of functions $\lambda(g) : \mathcal{S} \to \mathcal{S}$ for all elements $g \in G$, such that $\lambda(e) = \mathrm{id}_S$ and $\lambda(g)\lambda(g') = \lambda(gg')$. Such an action yields a representation of G on $\mathbb{K}[\mathcal{S}]$ by linearisation, the *free* $\mathbb{K}$*-vector space* generated by S.

Some simple examples of representations include the *zero representation* on the zero-dimensional vector space, and the *trivial representation* ρ_{triv} on the 1-dimensional vector space $\mathbb{K}$, where $\rho_{\mathrm{triv}}(g) = \mathrm{id}_{\mathbb{K}}$ for all $g \in G$.

D Insight into the algebra loss

To give further insight into component (*iv*), suppose our goal is to learn a representation $\widehat{\rho}_Z$ of the group C_2, which has group presentation $\{1, a \mid a^2 = 1\}$. Then $\widehat{\rho}_Z$ should satisfy $\widehat{\rho}_Z(1) = \mathrm{id}$ and $\widehat{\rho}_Z(a)\widehat{\rho}_Z(a) = \mathrm{id}$. To achieve this, we fix the parameter $\widehat{\rho}_Z(a) = \mathrm{id}$, and choose $\mathrm{ALG}_{C_2,d}$ and $\mathrm{REG}_{C_2,d}$ as follows, where $d = \dim(Z)$, the matrix I_d is the identity of size $d \times d$:

$$\mathrm{ALG}_{C_2,d} = \mathrm{MSE}\big(\widehat{\rho}_Z(a)^2,\ I_d\big)$$
$$\mathrm{REG}_{C_2,d} = \mathrm{MSE}\big(\widehat{\rho}_Z(a),\ \widehat{\rho}_Z(a)^{-1}\big).$$

We note that when $\mathrm{ALG}_{C_2,d}$ equals zero then $\widehat{\rho}_Z(a)^2 = \mathrm{I}_d$, and hence $\mathrm{REG}_{C_2,d}$ will equal zero. In this sense, the regularisation term is algebraically redundant, but is found to improve training.

E Proofs

Theorem 1. Let G be a finite group acting on a set $\mathcal{A}$ with action $\alpha_{\mathcal{A}}$, and on a vector space Z with a representation ρ_Z, with $\dim(Z) \geq |G|$. Suppose that the group acts freely and transitively on some subset $\mathcal{S} \subseteq \mathcal{A}$. If $E : \mathcal{A} \to Z$ is an equivariant function which is injective on $\mathcal{S}$, then Z contains the regular representation almost surely.

Proof. Let $\mathbb{R}[\mathcal{S}]$ denote the vector space of all formal linear combinations of $\mathcal{S}$ with coefficients in $\mathbb{R}$. Because $\alpha_{\mathcal{A}}|_{\mathcal{S}}$ is free and transitive it must be equivalent to the action of G on itself, and hence its linearisation $\mathbb{R}[\mathcal{S}]$ carries the structure of the regular representation. We write this representation explicitly as $\rho_{\mathbb{R}[\mathcal{S}]}(g)(\sum_s a_s s) := \sum_s a_s \alpha_{\mathcal{A}}(g)(s)$. Now, we can define the linear map $\tilde{E} : \mathbb{R}[\mathcal{S}] \to Z$ by $(\sum_s a_s s) \mapsto \sum_s a_s E(s)$. Because $E : \mathcal{A} \to Z$ is equivariant, we conclude that $\tilde{E} : \mathbb{R}[\mathcal{S}] \to Z$ is equivariant. Denote $V := \mathrm{Im}(\tilde{E}) = \mathrm{Span}_{\mathbb{R}}\{E(s) \mid s \in \mathcal{S}\}$, which is a linear subspace of Z of dimension at most $|G|$. By the first isomorphism theorem for representations (Fulton & Harris, 2004), we have $V \cong \mathbb{R}[\mathcal{S}]/\mathrm{Ker}(\tilde{E})$. Hence we have shown that Z contains a subspace V which is isomorphic to a quotient to the regular representation.

We must now show that Z contains the regular representation almost surely, i.e. that $\mathrm{Ker}(\tilde{E})$ is trivial almost surely. The linear function $\tilde{E}$ is fully specified by its action on the basis $\mathcal{S}$, i.e. $\{E(s) \mid s \in \mathcal{S}\}$.

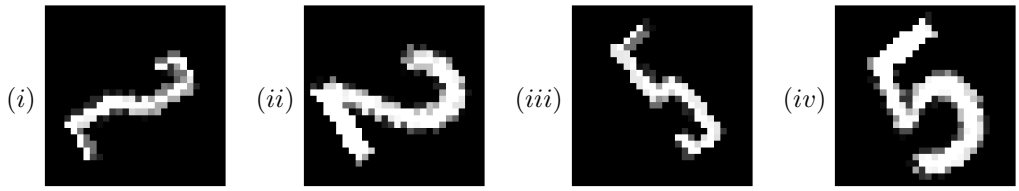

Figure 4: Examples of our augmented training dataset for the TMNIST experiment, from the chosen fonts 'Bahianita-Regular' (i), (iii) and 'IBMPlexSans-MediumItalic' (ii), (iv).

Since E is injective on $\mathcal{S}$, the set $\{E(s)|s \in S\}$ has cardinality $|G|$. Furthermore, because V has dimension at most G, we may embed each of the $E(s)$ into $y_s \in \mathbb{R}^{|G|}$. Therefore, in matrix representation, $\tilde{E}$ is obtained by collecting the vectors $\{y_s \,|\, s \in \mathcal{S}\}$ in a $|G| \times d$ matrix M. The condition for the kernel to be trivial is $\det M = 0$, which has Lebesgue measure 0 in $\mathbb{R}^{|G|\times|G|}$ (almost all matrices are invertible). Furthermore, we note that when $\dim Z = |G|$, it follows that Z must be isomorphic to the regular representation. □

F Exploratory experiments

Here we give details of the exploratory experiments we describe in Section 4. These use the TMNIST, MNIST and CIFAR10 datasets to determine the optimal representation on the latent space. Sections F.1, F.2 and F.3 provide details of the architectures and regularisation terms used for each of these experiments. In all runs, we use the Adam optimiser Kingma & Ba (2017) with default parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and report additional hyperparameters in Table 6. These were chosen through a manual tuning process.

F.1 TMNIST autoencoder, $G = C_2$

This experiment uses the TMNIST dataset Magre & Brown (2022) of digits rendered in a variety of typefaces. We select a data subset corresponding to just two typefaces 'IBMPlexSans-MediumItalic' and 'Bahianita-Regular', and augment with 180° rotations. We give some examples of our augmented dataset in Figure 4. The group we use here is $C_2 = \{1, a \,|\, a^2 = 1\}$ and, for a data point x, we define the group action $\rho_{\mathcal{X}}(a)(x)$ to be the data point with the font swapped, but the rotation and scaling unchanged. In particular, with reference to images Figure 4(i)–(iv), we have $\rho_{\mathcal{X}}(a)(i) = (ii)$, $\rho_{\mathcal{X}}(a)(ii) = (i)$, $\rho_{\mathcal{X}}(a)(iii) = (iv)$ and $\rho_{\mathcal{X}}(a)(iv) = (iii)$. For this experiment we set $L_{\text{task}} = \text{MSE}$, and we use a simple CNN autoencoder with hyperparameters given in Table 6. The architectural details can be found on the provided repository.

Experiment	Latent dim.	λ_a	λ_t	λ_e	LR	Batch Size
TMNIST C_2	6	1.0	0.5	1	0.003	64
MNIST D_3	18	0.5	0.495	0.005	0.003	64
CIFAR10 C_4	16	1.0	25	0.25	0.003	64

Table 6: Hyperparameters for exploratory experiments.

We use the following regularisation term:

$$\text{REG}_{C_2,d} = \text{MSE}(\widehat{\rho}_Z(a), \widehat{\rho}_Z(a)^{-1}) \tag{3}$$

Here $\widehat{\rho}_Z(a)^{-1}$ is computed with $\widehat{\rho}_Z(a)^{-1} = \texttt{torch.linalg.solve}(\widehat{\rho}_Z(a), \text{I}_d)$ for efficiency and numerical stability. We found empirically that this regularisation helps to stabilise the training of $\widehat{\rho}_Z(a)$, allowing us to achieve lower values for the algebra loss.

F.2 MNIST AUTOENCODER, $G = D_3$

This experiment uses the MNIST dataset Deng (2012) of handwritten digits. The group considered is $D_3 = \{e, r, r^2, r^3, s, rs \,|\, r^3 = e, s^2 = e, rsrs = e\}$, and on the input space we define the group action such that $\rho_{\mathcal{X}}(r)(x)$ is the counterclockwise rotation of x by 60 degrees, and $\rho_{\mathcal{X}}(s)(x)$ is the image generated by flipping x about the vertical axis. For this experiment, we set $L_{\text{task}} = \text{MSE}$, and use a simple MLP autoencoder with hyperparameters given in Table 6. The architectural details can be found on the provided repository.

We use the following regularisation term:

$$\text{REG}_{D_3,d} = -0.995 \text{ MSE}(\widehat{\rho}_Z(r)\widehat{\rho}_Z(s)\widehat{\rho}_Z(r)\widehat{\rho}_Z(s), \text{I}_d) \tag{4}$$

We determined empirically that this regularisation dampens the interaction between the matrices $\widehat{\rho}_Z(r)$ and $\widehat{\rho}_Z(s)$ in a way that improves training. Low final values of the algebra loss reported in Table 1 give evidence that we still obtain a high-quality representation despite this damping.

F.3 CIFAR10 CLASSIFIER, $G = C_4$

This experiment uses the CIFAR10 dataset Krizhevsky (2009) of 32x32 images organised in 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. The group considered is the cyclic group of size four C_4 of addition on the set $\{0, 1, 2, 3\}$ modulo 4. The element 1 is a generator for this group, and for an input vector x, we define the group action such that $\rho_{\mathcal{X}}(1)(x)$ is the rotation of x by 90 degrees counterclockwise. For this experiment we set $L_{\text{task}} = \text{CrossEntropy}$, and use a simple CNN classifier with hyperparameters given in Table 6. The architectural details can be found on the provided repository.

The regularisation term used is the following:

$$\text{REG}_{C_4,d} = \text{MSE}(\widehat{\rho}_Z(1)^3, \widehat{\rho}_Z(1)^{-1}) \tag{5}$$

Here, $\widehat{\rho}_Z(1)^{-1}$ is computed with $\widehat{\rho}_Z(1)^{-1} = \texttt{torch.linalg.solve}(\widehat{\rho}_Z(1), \text{I}_d)$ for efficiency and numerical stability. We determined empirically that this regularisation helps to stabilise the training of $\widehat{\rho}_Z(1)$ and the behaviour of its inverse.

G MAIN EXPERIMENTS

Here we give details of the main experiments we describe in Section 6, which test our model of Section 5 on tasks using the DDMNIST, MedMNIST, SMOKE and SHREC'11 datasets. Sections G.1, G.2, G.3 and G.4 provide details of the datasets, architectures and hyperparameters that we use. In all runs we use the Adam optimiser Kingma & Ba (2017) with default parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, with weight decay set to 0 for DDMNIST and MedMNIST, and set to 4×10^{-4} for SMOKE.

G.1 DDMNIST EXPERIMENTS

Data preparation. We follow closely the setup of the originators Veefkind and Cesa Veefkind & Cesa (2024). To generate this dataset, pairs of MNIST 28x28 images are chosen uniformly at random, and independently augmented according to the corresponding group action for $G \in \{C_4, C_2, D_4\}$ as per Table 7. We give an example in Figure 5. To ensure comparability of our results with the original paper, for $G \in \{C_4, D_4\}$ we follow their method of introducing interpolation artefacts by rotating each digit image by a random angle $\theta \in [0, 2\pi)$, and then rotating it back by $-\theta$; for $G = C_2$ these interpolation artefacts are not added, in line with the original paper. Finally, the two images are concatenated horizontally, and padded so that the final image is 56×56. In this way, we obtain a dataset of 10,000 images with labels in the set $\{(0, 0), (0, 1), \ldots, (9, 9)\}$.

Architecture. We use the same CNN architecture as in Veefkind and Cesa Veefkind & Cesa (2024), except that the final convolutional layer has an increased number of filters, from 48 to 66. We make this change so that we can fit a copy of the regular representation of $D_4 \times D_4$. To ensure a fair comparison, the results reported in Table 3, including those for SCNN, RPP, etc, are those obtained

Group	Type	Generators	Size
C_4	Cyclic	90° rotation	4
C_2	Dihedral	Horizontal reflection	2
D_4	Dihedral	Horizontal reflection and 90° rotation	8

Table 7: Symmetry groups and their actions on DDMNIST.

Before augmentation After augmentation

Figure 5: Examples of training data for the DDMNIST experiment with $G = D_4$. The left figure shows concatenated MNIST digits, and the right figure shows the result after a random augmentation. In this instance, the left digit is augmented with a reflection about the vertical axis, and the right digit is augmented with a clockwise 90-degree rotation.

with the increased number of filters, which we found marginally improved performance. Furthermore, we use a different learning rate for the CNN model, as we found that this increased performance and ensured a more meaningful baseline comparison. The CNN architectural details can be found on the provided repository.

Hyperparameters. We report the hyperparameters used for the CNN and our model for the DDMNIST experiments in Table 8. These hyperparameters were chosen after a grid search with the following values: learning rate $\in \{0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005\}$, and equivariance coupling strength $\lambda \in \{0.5, 1, 1.5, 2\}$. All other hyperparameters match those used by Veefkind and Cesa.

	C_4		C_2		D_4	
	LR	λ	LR	λ	LR	λ
CNN	0.0005	-	0.001	-	0.0005	-
Standard rep	-	-	-	-	0.0005	1
Ours (regular)	0.001	2	0.001	1	0.0005	1

Table 8: Hyperparameters for DDMNIST experiments.

G.2 MEDMNIST EXPERIMENTS

Data preparation. For this experiment, we use three subsets of the MedMNIST dataset Yang et al. (2023), in line with Veefkind and Cesa Veefkind & Cesa (2024): Nodule3D, Synapse3D and Organ3D, each containing 3D images of size 28x28x28. Nodule3D is a public lung nodule dataset, containing 3D images from thoracic CT scans; for this dataset, the task is to classify each nodule as benign or malignant. Synapse3D contains 3D images obtained from an adult rat with a multi-beam scanning electron microscope; the task is to classify whether a synapse is excitatory or inhibitory. Organ3D is a classification task for a 3D images of human body organs, with the following labels: liver, right kidney, left kidney, right femur, left femur, bladder, heart, right lung, left lung, spleen and pancreas.

For augmentations, we choose the octahedral group of orientation-preserving rotational symmetries of the cube, which is isomorphic to the permutation group S_4. We define its action $\rho_{\mathcal{X}}(g)$ on a 3D image x by applying the corresponding rotational symmetry of the cube. Specifically, we parameterise g as

a tuple (l, θ) where $l = (x, y, z)$ specifies a rotation axis and θ specifies the rotation angle about the axis l to obtain 24 rotation matrices each with size 3×3, one for each of the 24 elements of S_4. In summary, we have rotation matrices corresponding to the following tuples:

Identity (1) $(l, 0)$ for any l.
Coord-axis (9) (l, θ) for $l \in \{(1,0,0), (0,1,0), (0,0,1)\}$ and $\theta \in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.
Edge-mid (6) (l, θ) for $l \in \{(1,1,0), (1,-1,0), (1,0,1), (1,0,-1), (0,1,1), (0,1,-1)\}$ and $\theta = \pi$.
Body-diag (8) (l, θ) for $l \in \{(1,1,1), (1,1,-1), (1,-1,1), (-1,1,1)\}$ and $\theta \in \{\frac{2\pi}{3}, \frac{4\pi}{3}\}$.

Architecture. For these experiments we use the same CNN-based ResNet architecture as Veefkind and Cesa Veefkind & Cesa (2024). This is formed from seven 3D convolutional layers, formed into 3 blocks with residual connections, along with batch normalisation and pooling. The architectural details can be found on the provided repository.

Hyperparameters. We report the hyperparameters used for the baseline with S_4 augmentations, and for our model in the MedMNIST experiments in Table 9. These hyperparameters were chosen after a grid search with the following values: learning rate $\in \{0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005\}$, and equivariance coupling strength $\lambda \in \{0.5, 1, 1.5, 2\}$. All other hyperparameters are the same as those used by Veefkind and Cesa.

	Nodule3D		Synapse3D		Organ3D	
	LR	λ	LR	λ	LR	λ
CNN (Augmented)	0.00005	-	0.0001	-	0.0001	-
Ours	0.00005	1	0.0001	1	0.0001	2

Table 9: Hyperparameters for MedMNIST experiments.

G.3 SMOKE EXPERIMENTS

Data preparation. Here we use the SMOKE dataset of Wang et al. Wang et al. (2022), which consists of smoke simulations with varying intial conditions and external forces presented as grids of (x, y) components of a velocity field (see Figure 6 for a visualisation). The task is to predict the next 6 frames of the simulation given the first 10 frames only. We evaluate each model on two metrics: Future, where the test set contains future extensions of instances in the training set; and Domain, where the test and training sets are from different instances.

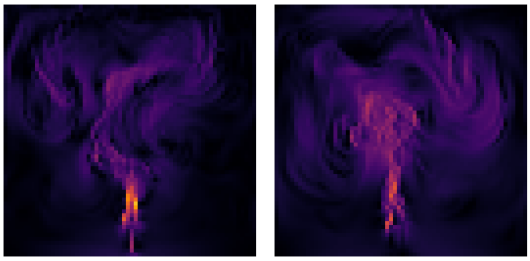

Figure 6: Approximately equivariant dynamics of smoke plumes Holl et al. (2020).

Architecture. We use the same CNN architecture, train and evaluation setups as in Veefkind and Cesa Veefkind & Cesa (2024), which they reproduced from Wang et al. Wang et al. (2022). The architectural details can be found on the provided repository. Because the latent space has the same geometric structure as the input data, i.e. $Z = \mathbb{R}^c \times \mathbb{R}^h \times \mathbb{R}^w$ (channels$\times$height$\times$width), we choose a representation of C_4 given by the regular representation in each channel separately.

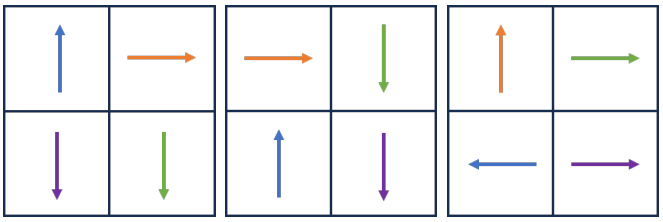

Figure 7: Examples of a velocity field and its augmentations with and without reorientation. Rotating by 90° counterclockwise without reorienting simply moves the spatial grid, but breaks the physical meaning of the underlying system.

Hyperparameters. For both CNN models, with C_4 augmentations and without, and for our model, we use a learning rate of 0.001. Additionally, for our model, we set $\lambda = 0.005$. These hyperparameters were chosen after a grid search with the following values: learning rate $\in \{0.001, 0.005, 0.0001, 0.0005\}$, and equivariance coupling strength $\lambda \in \{0.005, 0.05, 0.5, 1\}$. For all other hyperparameters, we copy the values used by Veefkind and Cesa.

G.4 SHREC '11 EXPERIMENTS

Data preparation. We use the SHREC '11 dataset Lian et al. (2011); Mitchel et al. (2022) where each 3D shape is also transformed with conformal transformations. We perform augmentation according to the group O_h of octahedral symmetries.

Architecture. We use the same architecture as the original authors Mitchel et al. (2024), which is a ResNet-based autoencoder. Similarly to the smoke experiment, the latent space retains a geometric structure. Therefore, we choose a representation of O_h given by the regular representation in each channel separately.

Hyperparameters. Due to computational constraints, we do not perform hyperparameter tuning, and we keep the same hyperparameters as the original authors Mitchel et al. (2024), except that we set the batch size to 4. We set $\lambda = 0.5$. Additionally, we symmetrize the equivariance loss to the decoder too, i.e., with $\lambda' = 0.8$,

$$\lambda' \, ||\rho_{\mathcal{X}}(g)(x) - D(\rho_Z(g)(E(x)))||$$

H SENSITIVITY ANALYSIS

To assess the practical usability of our method, we performed a sensitivity analysis on the hyperparameter λ, which controls the strength of the equivariance loss. We evaluated our model on the DDMNIST D_4 task across six different values for λ: $\{0, 0.05, 0.5, 1, 1.5, 2\}$, with $\lambda = 0$ being the baseline. The results, reported in Figure 8, show that while peak performance is achieved at $\lambda = 1$, the model maintains high accuracy and low variance across a wide range of values (0.5 to 2.0). This analysis demonstrates that our method is robust to the specific choice of λ.

Figure 8: Mean accuracy and standard deviation (over 5 runs) for different values of λ on the DDMNIST D_4 task. $\lambda = 0$ is the baseline.