Algebraic priors for approximately equivariant networks

Riccardo Ali, Pietro Liò & Jamie Vicary
University of Cambridge
{rma55,pl219,jv258}@cam.ac.uk

Abstract

Equivariant neural networks incorporate symmetries through group actions, embedding them as an inductive bias to improve performance. Existing 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 involve 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 group representation theory. We prove that for an equivariant encoder over a finite group, the latent space must almost surely contain one copy of its 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 group's 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 defining and trivial group representation baselines.

1 Introduction

Equivariance is a powerful inductive bias used to align model representations with the underlying group structure of the data (Bronstein et al., 2021), as shown in Figure 1. Strict equivariance assumes a level of symmetry that real-world data rarely possesses (Holmes, 2012; Holl et al., 2020; Yang et al., 2023), thus over-constraining architectures can degrade their performance (Petrache & Trivedi, 2025). *Approximate equivariance* offers a mechanism to better balance geometric structure with representational flexibility (Wang et al., 2021; Finzi et al., 2021).

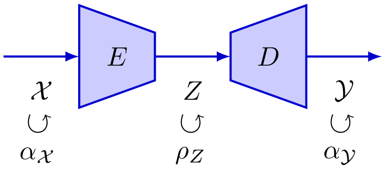


Figure 1: Generic architecture with input set $\mathcal{X}$, latent space Z and output set $\mathcal{Y}$, with group actions $\alpha_{\mathcal{X}}$, $\alpha_{\mathcal{Y}}$ on the input and output sets, and potentially a representation ρ_Z on the latent space.

Augmented architectures achieve approximate equivariance by modifying the model structure at the cost of increased computational complexity and often tied to specific architectures (Veefkind & Cesa, 2024). Avoiding such design constraints, loss-based approaches enforce equivariance via soft constraints as an additional penalty term (Dupont et al., 2020; Koyama et al., 2024; Jin et al., 2024). While flexible, these loss-based methods often rely on arbitrary priors or heuristics, leaving the factors determining the learned representations' geometry underexplored or confined to specific empirical settings (Bökman et al., 2024).

In this work, we provide a theoretical and empirical analysis of soft-constrained equivariance for linear representations of finite groups under data augmentations. We analytically prove that, in this setup, the latent space *must contain the regular representation almost surely*, and we give a lower bound of its multiplicity. Our empirical studies, validating the theory, prove the regular representation a principled inductive bias. We contrast this choice against both learnable and other fixed representations, showing competitive or improved performance. This yields a simple, scalable, and model-agnostic framework achieving strong results on invariant and approximately-equivariant benchmarks. We summarize our contributions as follows:

- We formally characterize latent space representations, proving that for an approximately equivariant encoder, the regular representation must emerge almost surely. We derive a lower bound on its multiplicity and empirically validate that neural networks tend to learn a linear representation aligned with this structure.
- We demonstrate that enforcing the regular representation yields a lightweight framework that matches or exceeds state-of-the-art performance. Crucially, our method requires no additional learnable parameters and only a single hyperparameter, whereas competing baselines often demand 5–$20\times$ more parameters.
- We empirically establish the optimality of the regular representation across a diverse suite of invariant and equivariant tasks. We validate this choice by contrasting it against both learnable approaches and alternative fixed representations, such as the trivial and defining representations.

2 Related Work

We provide a brief overview of methods most relevant to approximate equivariance here; an extended review of the broader literature is available in Appendix B.

Augmented architectures. While strict equivariance is well-established through steerable CNNs (Cohen & Welling, 2016; Weiler & Cesa, 2019), recent work focuses on relaxing these constraints to improve flexibility. Residual Pathway Priors (**RPP**) (Finzi et al., 2021) combine a strictly equivariant layer with an unconstrained one, while Lift Expansion (**LIFT**) (Wang et al., 2021) factorizes inputs into equivariant and non-equivariant subspaces. More recently, Probabilistic Steerable CNNs (**PSCNN**) (Veefkind & Cesa, 2024) propose learning the optimal equivariance strength per layer. However, these architectural relaxations often incur significant computational costs, requiring elevated parameter counts to match the performance of simpler baselines (see Section 5.1).

Loss-based approaches. An alternative paradigm enforces equivariance via the loss function on the latent space. Dupont et al. (2020) propose fixing the latent representation for neural rendering; however, their approach relies on the defining representation of $O(3)$ and assumes the latent geometry matches the input, limiting its generality. Other methods attempt to learn the transformation, such as Neural Fourier Transforms (**NFT**) (Koyama et al., 2024) or subspace steerers (Bökman et al., 2024).

We distinguish our work by identifying the *regular representation* of a finite group as the theoretically and empirically justified fixed target. This eliminates the need for additional learnable parameters or complex architectures. Empirically, this algebraic prior proves robust, frequently outperforming unconstrained baselines and even architectures specifically adapted for continuous symmetries.

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 C and D.

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: $\rho_{\mathrm{reg}}(g)(\sum_i c_i g_i) = \sum_i c_i(gg_i)$. By construction we have $\dim(\rho_{\mathrm{reg}}) = |G|$, the size of the group.

Irreducibility. Given representations ρ_1 on V_1 and ρ_2 on V_2, their *direct sum* $\rho_1 \oplus \rho_2$ acts on $V_1 \oplus V_2$ component-wise: $(\rho_1 \oplus \rho_2)(g)(v_1, v_2) = (\rho_1(g)v_1, \rho_2(g)v_2)$. The *dimension* of a representation is the dimension of its underlying vector space. A representation is *irreducible* (or an *irrep*) if it cannot be decomposed into such a sum of non-trivial subrepresentations. For finite groups, any representation ρ decomposes into a direct sum of irreps: $\rho \cong \bigoplus_i m_i \rho_i$, where m_i is the multiplicity. Over $\mathbb{C}$, the regular representation decomposes as $\rho_{\mathrm{reg}} \cong \bigoplus_i d_i \rho_i$, containing every irrep ρ_i with multiplicity equal to its dimension d_i. For example, the group S_3 has just the trivial (dim 1), sign

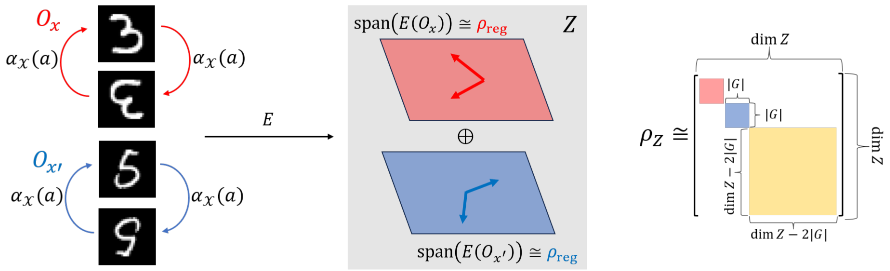

$\mathcal{O}_x$
$\alpha_{\mathcal{X}}(a)$
$\alpha_{\mathcal{X}}(a)$
$\mathcal{O}_{x'}$
$\alpha_{\mathcal{X}}(a)$
$\alpha_{\mathcal{X}}(a)$
E
$\text{span}(E(\mathcal{O}_x)) \cong \rho_{\text{reg}}$
Z
$\oplus$
$\text{span}(E(\mathcal{O}_{x'})) \cong \rho_{\text{reg}}$
$\rho_Z \cong$
$\dim Z$
$|G|$
$|G|$
$\dim Z - 2|G|$
$\dim Z - 2|G|$
$\dim Z$

Figure 2: Illustration of our theory for an approximately equivariant encoder E_θ and $G = C_2 = \{1, a\}$, with $\alpha_{\mathcal{X}}$ acting by horizontal flips. If $E(\mathcal{O}_x)$, $E(\mathcal{O}_{x'})$ are full rank and linearly independent, Z must contain a separate copy of the regular representation ρ_{reg} for each with probability 1.

(dim 1) and standard (dim 2) irreps (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. While the decomposition over $\mathbb{R}$ may yield different multiplicities, any real representation can be analyzed via its complexification (for instance, through eigendecomposition).

Orthogonality of representations. The presence and multiplicity of an irrep ρ' within a representation ρ is determined by the inner product of their characters $\chi_\rho(g) = \text{Tr}(\rho(g))$, which is defined as $\langle \chi_\rho, \chi_{\rho'} \rangle = \frac{1}{|G|} \sum_{g \in G} \overline{\chi_\rho(g)} \chi_{\rho'}(g)$. This formula and complexification allow us to decompose any representation into its irreducible components.

4 Identifying optimal representations

We consider the architecture in Figure 1 with data $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a finite group G acting on input/output spaces via $\alpha_{\mathcal{X}}, \alpha_{\mathcal{Y}}$. Let $E_\theta : \mathcal{X} \to Z \subseteq \mathbb{R}^d$ be an encoder parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$. If G acts on Z via a representation ρ_Z, we say E_θ is (ρ_Z, ε)-*equivariant* if $\sup_{x,g} \|E(\alpha_{\mathcal{X}}(g)(x)) - \rho_Z(g)E(x)\| \leq \varepsilon$, encoding the degree of equivariance of E_θ. We aim to answer the following research questions (RQs):

RQ1 Are there algebraic or geometric constraints for what representations can approximately equivariant models instantiate in the latent space?
RQ2 What representations do models tend to learn in practice, when free to learn one?
RQ3 Does explicitly enforcing this emerging structure yield practical benefits?

4.1 Algebraic constraints for approximate equivariance (**RQ1**)

Let $\mathcal{O}_x := \{\alpha_{\mathcal{X}}(g)(x) \,|\, g \in G\}$ denote the *data orbit* of a sample x, containing all G-augmented versions of x. We suppose x is a data sample 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). We focus on the *embedded orbit* $E_\theta(\mathcal{O}_x)$. When $\dim(Z) \geq |G|$, we say $E_\theta(\mathcal{O}_x)$ is *full rank* if its span has dimension $|G|$, corresponding to a strictly positive smallest singular value $\sigma_{\min} > 0$, and *rank deficient* otherwise. We now present our main theoretical results (proofs in Appendix F).

Theorem 1. *Let E_θ be a (ρ_Z, ε)-equivariant encoder and σ_{min} the smallest singular value of $E_\theta(\mathcal{O}_x)$. If $\sigma_{\min} > 0$, there exists a representation $(V, \tilde{\rho})$ with $V \subseteq Z$ and $\tilde{\rho} \cong \rho_{reg}$ such that for all $g \in G$:*

$$||\rho_Z(g)|_V - \tilde{\rho}(g)||_{op} \leq \frac{\varepsilon}{\sigma_{min}} \sqrt{|G|}$$

Theorem 2 (informal)**.** *If E_θ is also real analytic in its inputs and parameters, and trained by gradient descent, then for each training sample $x \in \mathcal{X}$, exactly one of the following holds:*

(i) for all possible parameterisations $\theta \in \Theta$, the vectors $E_\theta(\mathcal{O}_x)$ are rank deficient, $\sigma_{min} = 0$.

(ii) with probability 1, the vectors $E_\theta(\mathcal{O}_x)$ are full rank, $\sigma_{min} > 0$. Hence the latent space contains the regular representation, and the bound from Theorem 1 applies to E_θ.

Analyticity is detailed in Appendix F.1. Case *(i)* is restrictive, appearing when E_θ is structurally constrained (e.g., invariant by design). In contrast, Case *(ii)* is generic: for standard architectures, sampling θ yields a full-rank orbit $E_\theta(\mathcal{O}_x)$ with probability 1, confirming the presence of the regular representation. Importantly, this property extends across the dataset, where *each linearly independent orbit contributes a separate copy of ρ_{reg}* (see Appendix F).

Implications of Theorem 2. This result highlights a fundamental tension in learning approximate symmetries. Consider an encoder trained for invariance via a penalty $\|E(x) - E(\alpha(g)x)\|$. While the optimization objective drives the encoder toward rank deficiency (true invariance), Theorem 2 guarantees that the trajectory remains almost surely in the full-rank regime ($\sigma_{\text{min}} > 0$). Consequently, the network cannot achieve true information erasure; it is forced instead into a state of signal compression, where the regular representation is preserved but condensed ($\sigma_{\min} \ll 1$). A critical question is whether this theoretical persistence is merely a numerical artifact or representationally significant. To answer this, we employ *linear probes* (Alain & Bengio, 2018), a standard technique in interpretability literature for quantifying feature decodability, in a controlled setup (Rotated MNIST $C4$). Despite E being regularized for invariance, these probes recover the input rotation with 60.6% accuracy (vs. 25% chance). See Appendix G for more details. This confirms that the attenuated signal is not just numerical noise, but a distinguishing feature that persists despite the training objective. It underscores our core insight: the network does not structurally transition to the trivial representation; rather, it maintains the algebraic structure of the regular representation, instantiating the target geometry solely through geometric contraction.

> **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 linearly independent full rank embedded orbit. This is summarized in Figure 2.

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 of learned representations (**RQ2**)

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

$L_{\text{opt}} = L_{\text{task}}\big(D(E(x_i)),\, y_i\big)$	**Task loss.** Trains encoder and decoder on the supervised objective.
$+\, \lambda_t\, L_{\text{task}}\big(D(\widehat{\rho}_Z(g)E(x_i)),\, \alpha_{\mathcal{Y}}(g)(y_i)\big)$	**Latent→Output Equivariance.** Encourages $D(\widehat{\rho}_Z(g)E(x))$ to match $\alpha_{\mathcal{Y}}(g)(y)$.
$+\, \lambda_e\, \text{MSE}\big(\widehat{\rho}_Z(g)E(x_i),\, E(\alpha_{\mathcal{X}}(g)(x_i))\big)$	**Input→Latent Equivariance.** Encourages $\widehat{\rho}_Z(g)E(x)$ to match $E(\alpha_{\mathcal{X}}(g)x)$.
$+\, \lambda_a\big(\text{ALG}_{G,d} + \text{REG}_{G,d}\big)$	**Algebra Loss.** Encourages algebraic properties for $\widehat{\rho}_Z$ to be a group representation.

We give additional insight into the algebra loss in Appendix E. We further investigate the latent structure through exploratory studies. Theorems 1 and 2 imply that the learned representation contains copies of the regular representation, with a multiplicity lower-bounded by the number of linearly independent embedded data orbits. We empirically assess this multiplicity in controlled environments, ensuring coverage of invariant versus equivariant objectives, abelian and non-abelian groups (C_2, C_4, D_3), and geometric versus non-geometric actions. We observe that analytic encoders (initialized via $\mathcal{N}(\mathbf{0}, \mathbf{1})$) consistently converge to a representation composed *entirely* of linearly independent copies of the regular representation (methodology detailed in Appendix H.1). Further investigations into the algebra loss, alternative depths, and initialization schemes are provided in Appendices E, H.4, and H.5.

Non-analytic encoders. While Theorem 2 formally assumes analyticity, modern architectures can employ piecewise-analytic activations like ReLU. Theoretically, such functions can be arbitrarily well-approximated by analytic maps (Stone-Weierstrass), suggesting our results should transfer (see discussion in Appendix F.1). We empirically validate this in Appendix H.3, demonstrating that ReLU-based networks exhibit behavior identical to their analytic counterparts, consistently converging to linearly independent copies of the regular representation.

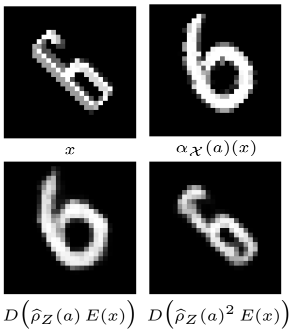
Table 1: Left (Right) TMNIST (MNIST) analytic autoencoder task, learned representations of C_2 (D_3) on the latent space. Z is taken as the middle layer, which carries the output of the encoder.

	Irrep. counts				
Run	-1	$+1$	Alg. loss	Eq. loss	Orbs
1	3	5	9.9×10^{-10}	1.6×10^{-3}	3
2	4	4	1.1×10^{-7}	1.3×10^{-3}	3
3	3	5	7.4×10^{-10}	1.2×10^{-4}	4
4	4	4	2.3×10^{-10}	9.1×10^{-5}	4
5	4	4	2.9×10^{-9}	1.5×10^{-3}	3

	Irrep. counts					
Run	Triv	Sgn	Std	Alg. loss	Eq. loss	Orbs.
1	2.98	3.1	5.98	1.1×10^{-4}	1.4×10^{-2}	3
2	3.03	2.98	6.01	5.8×10^{-3}	2.1×10^{-3}	3
3	3.1	2.97	5.70	1.6×10^{-4}	2.7×10^{-2}	3
4	2.96	3.15	5.75	2.3×10^{-3}	3.1×10^{-3}	3
5	2.91	2.99	6.02	7.5×10^{-2}	2.2×10^{-2}	3

CNN Equivariant task, abelian group, non-geometric action. For our first experiment we use the TMNIST (Magre & Brown, 2022)dataset, of digits rendered in a variety of typefaces. We choose a subset of two typefaces only, producing 20 images, augmenting with 180° rotations and random scaling in $(0.8, 1.2)$. 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 $d = \dim(Z) = 8$.

Table 1 shows our findings, with each run giving one row of the table, and Figure 3 shows a visualization. Low values in the algebra and equivariance loss columns reveal 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 one copy of the -1 and +1 representations, and the learned $\widehat{\rho}_Z$ are close to a multiple of the regular representation. Furthermore, we report the number of linearly independent embedded orbits and, as expected, this corresponds to the number of copies of the regular representation found (Section 4.1).



x $\quad$ $\alpha_{\mathcal{X}}(a)(x)$

$D\big(\widehat{\rho}_Z(a)\; E(x)\big)$ $\quad$ $D\big(\widehat{\rho}_Z(a)^2\; E(x)\big)$

Figure 3: Visualization of learned E, D and $\widehat{\rho}_Z$ on a sample x with $\alpha_{\mathcal{X}}$ swapping fonts. The algebraic loss correctly enforced $\widehat{\rho}_Z(a)^2 = I_d$ while retaining equivariance.

MLP Equivariant task, non-abelian group group, geometric action. For our second experiment we choose the MNIST (Deng, 2012) dataset, with augmentations given by arbitrary rotations. We choose the group $G = D_3$ of symmetries of an equilateral triangle with the generators r, s (acting as 120-degree rotation and flip, respectively) 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 correspond to constraints in the presentation: $\mathrm{ALG}_{D_3,d} = \mathrm{MSE}(\widehat{\rho}_Z(r)^3, \mathrm{I}_d) + \mathrm{MSE}(\widehat{\rho}_Z(s)^2, \mathrm{I}_d) + \mathrm{MSE}(\widehat{\rho}_Z(r)\widehat{\rho}_Z(s)\widehat{\rho}_Z(r)\widehat{\rho}_Z(s), \mathrm{I}_d)$.

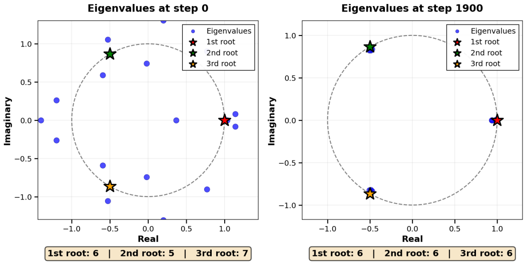


Figure 4: Eigenvalues of $\widehat{\rho}_Z(r)$ over training. Counts near third roots of unity (bottom) show convergence to a uniform distribution despite uneven initialization.

For the nonabelian group D_3, we determined the learned representation's composition using orthogonality of characters (Section 3). 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 4 illustrates the eigenvalues of the generator $\widehat{\rho}_Z(r)$ dynamically clustering around the third roots of unity during training, despite an uneven initialization.

Table 2: CIFAR classifier task with analytic encoder, representations of C_4 learned on latent space. Z is taken as the main feature layer before the final classification head.

	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

CNN Invariant task, abelian group, geometric action This experiment uses the CIFAR10 dataset (Krizhevsky, 2009). We choose the abelian 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 Table 2 shows that the network learns a representation close to a multiple of the regular representation. Furthermore, each linearly independent embedded data orbit contributes a distinct copy of this representation.

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 Fixing the regular representation (**RQ3**)

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 as an algebraic inductive bias. 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, yielding:

$$\rho_Z := n \cdot \rho_{\mathrm{reg}} + \max(\dim(Z) - n|G|, 0) \cdot \rho_{\mathrm{triv}} \tag{1}$$

When the latent space is geometrically structured, e.g., 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 5.1). Denoting $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ an element of the training set, $g \in G$ a group element, and $L_{\mathrm{task}}(x_i, y_i)$ the task loss function, we train according to:

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

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 J, 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.

5.1 Experiments

We benchmark our method across four distinct tasks against a comprehensive suite of state-of-the-art approximate equivariance methods, including SCNN, E2CNN, LIFT, RPP, RGroup, RSteer, PSCNN, NIso, NFT, and MC. For all comparisons, we adhere to the experimental setups described in the original papers. Our results demonstrate that our approach frequently outperforms these complex baselines while utilizing simpler architecture with fewer parameters and a comparable or lower computational budget. To isolate our contributions, we also include both augmented and unaugmented CNN baselines. We quantify the significance of our gains using Cohen's d-statistic (Hermann et al., 2024, p59), which consistently indicates very large performance improvements (see discussion in Appendix I.1). The equivariance loss is applied to the encoder output for autoencoding tasks and the penultimate layer for classification. Finally, to validate the optimality of the regular representation, we provide ablations replacing it with the trivial representation and the *defining representation* (defined formally in Appendix D). Full technical details, including hyperparameter selection and a sensitivity analysis for the coupling strength λ, are provided in Appendices I and J.

Invariant classification task, DDMNIST. 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. We also provide a comparison with the defining and trivial representations as an ablation study. While for the groups C_2 and C_4 the two representations 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 and effect size analysis can be found in Appendix I.2. These statistics show a very large effect size for our model over the CNN baseline, and a large effect size for our model compared to the majority of results for other architectures.

Table 3: 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	$C_4 \uparrow$	#Params(M) $\downarrow$	$C_2 \uparrow$	#Params(M) $\downarrow$	$D_4 \uparrow$	#Params(M) $\downarrow$
CNN	0.907 (0.004)	**0.03**	0.938 (0.006)	**0.03**	0.800 (0.001)	**0.03**
SCNN	0.484 (0.008)	0.12	0.474 (0.003)	**0.03**	0.431 (0.010)	0.15
Restriction	0.914 (0.007)	0.12	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	0.842 (0.011)	1.23
Trivial rep	0.874 (0.004)	**0.03**	0.938 (0.007)	**0.03**	0.819 (0.004)	**0.03**
Defining rep	–		–		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**

Approximately equivariant classification Task, MedMNIST3D. 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 MedMNIST3D accuracies for different models and groups. For Nodule and Synapse, our method is comparable or outperforms other architectures, while having fewer parameters. The regular representation consistently outperforms the defining and trivial representations, providing further empirical evidence for its optimality. For the Organ dataset, canonical orientation is a key feature, and so the symmetry action conflicts with the task. This may explain our method's underperformance in this task (shared by the augmented CNN baseline). Further discussion can be found in Appendix I.3, which shows our method has very large positive effect sizes for Nodule and Synapse datasets.

Table 4: 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 $\uparrow$	Synapse $\uparrow$	Organ $\uparrow$	#Params(M) $\downarrow$
N/A	CNN	0.873 (0.005)	0.716 (0.008)	0.920 (0.003)	00.19
Aug	CNN	0.879 (0.007)	0.761 (0.008)	0.632 (0.005)	00.19
SO(3)	SCNN	0.873 (0.002)	0.738 (0.009)	0.607 (0.006)	**00.13**
SO(3)	RPP	0.801 (0.003)	0.695 (0.037)	0.936 (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)	00.19
O(3)	RPP	0.810 (0.013)	0.722 (0.023)	**0.940** (0.006)	29.30
O(3)	PSCNN	0.873 (0.008)	0.769 (0.005)	0.905 (0.004)	03.51
$\mathrm{Sym}_{\mathrm{cube}}$	Trivial rep	0.867 (0.001)	0.743 (0.002)	0.571 (0.002)	00.19
$\mathrm{Sym}_{\mathrm{cube}}$	Defining rep	0.837 (0.013)	0.756 (0.019)	0.560 (0.025)	00.19
$\mathrm{Sym}_{\mathrm{cube}}$	Ours (regular)	**0.887** (0.005)	**0.770** (0.002)	0.642 (0.056)	00.19

Approximately equivariant autoregression task. We evaluate our method on the SMOKE dataset, generated with PhiFlow (Holl et al., 2020) by Wang et al. (2022) (see Figure 8 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. Further details can be found in Appendix I.4, showing very large positive effect sizes for all models except PSCNN.

Equivariant autoencoding task, 3D shapes. 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 apply our 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. 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. Further details can be found in Appendix I.5, with effect sizes showing equivalence between our method and NIso, and very large positive effect size for the other models.

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) Test RMSE for SMOKE dataset.

Group	Model	Future↓	Domain↓	#Params(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)	0.62
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	0.78 (0.01)	0.61 (0.01)	**0.25**

(b) Test accuracy for SHREC '11 dataset.

Model	Acc.↑
NIso Mitchel et al. (2024)	90.26 (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 Conclusions

Limitations and Future Work. Our method requires data augmentation, although this is typically inexpensive when the group action on the input space is easy to construct, and our ablations with an augmented baseline show that our model delivers benefits far beyond augmentation. We would also like to explore how our model could enable augmentation directly in the latent space. Another limitation is that our theoretical analysis is restricted to finite groups; however, we show empirically that applying our method to a finite subgroup yields competitive results. Future work could investigate the suitability of such subgroups and extend the theoretical analysis to infinite groups.

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 prior. We prove that for finite groups, this structure is the regular representation, which must appear almost surely in the latent space of any approximately equivariant encoder. By enforcing this structure via an auxiliary loss, our method achieves competitive or superior performance to SOTA models, while requiring in some cases significantly fewer parameters. Furthermore, we empirically show the optimality of the regular representation via ablations with the defining and trivial representations.

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.

Acknowledgments

The authors would like to acknowledge Ioannis Markakis, Francesco Caso, Christopher Irwin and Lorenzo Sani for their insightful feedback.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018. URL https://arxiv.org/abs/1610.01644.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances, 2021. URL https://arxiv.org/abs/2102.12452.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Georg Bökman, Johan Edstedt, Michael Felsberg, and Fredrik Kahl. Steerers: A framework for rotation equivariant keypoint descriptors. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4885–4895. IEEE, 2024. doi: 10.1109/cvpr52733.2024.00467. URL http://dx.doi.org/10.1109/cvpr52733.2024.00467.

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.

D.H. Fremlin. *Measure Theory*. Number v. 1 in Measure theory. Torres Fremlin, 2000. ISBN 9780953812905. URL https://books.google.co.uk/books?id=2_lHYXeEd7YC.

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

Odd Erik Gundersen, Saeid Shamsaliei, Håkon Sletten Kjærnli, and Helge Langseth. On reporting robust and trustworthy conclusions from model comparison studies involving neural networks and randomness. In *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, ACM REP '23, pp. 37–61, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701764. doi: 10.1145/3589806.3600044. URL https://doi.org/10.1145/3589806.3600044.

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

Katherine Hermann, Jennifer Hu, and Michael Mozer. Experimental design and analysis for AI researchers. Invited tutorial at NeurIPS 2024, 2024. URL https://neurips.cc/virtual/2024/tutorial/99528.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

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.

Jen-tse Huang, Man Ho Lam, Eric John Li, Shujie Ren, Wenxuan Wang, Wenxiang Jiao, Zhaopeng Tu, and Michael R. Lyu. Apathetic or empathetic? evaluating llms'emotional alignments with humans. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97053–97087. Curran Associates, Inc., 2024. doi: 10.52202/079017-3077. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b0049c3f9c53fb06f674ae66c2cf2376-Paper-Conference.pdf.

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.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael Curtis Mozer, and Rebecca Roelofs. Soft calibration objectives for neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=-tVD13hOsQ3.

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.

Brando Miranda, Patrick Yu, Saumya Goyal, Yu-Xiong Wang, and Sanmi Koyejo. Is pre-training truly better than meta-learning?, 2025. URL https://arxiv.org/abs/2306.13841.

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.

Giorgos Nikolaou, Tommaso Mencattini, Donato Crisostomi, Andrea Santilli, Yannis Panagakis, and Emanuele Rodolà. Language models are injective and hence invertible, 2025. URL `https://arxiv.org/abs/2510.15511`.

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

Mircea Petrache and Shubhendu Trivedi. Approximation-generalization trade-offs under (approximate) group equivariance, 2025. URL `https://arxiv.org/abs/2305.17592`.

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.

Max Welling and Taco S. Cohen. Transformation properties of learned visual representations. In *Proceedings of ICLR 2015*, 2014.

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

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Interpretable transformations with encoder-decoder networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5737–5746, 2017. doi: 10.1109/iccv.2017.611. URL `http://dx.doi.org/10.1109/iccv.2017.611`.

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.

APPENDIX

A CODE

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

https://github.com/rick-ali/parameter-free-approximate-equivariance

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

B EXTENDED 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). Cohen & Welling (2016) employ the framework of steerable functions (Hel-Or & Teo, 1998) to construct a rotation-equivariant Steerable CNN architecture (**SCNN**), 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 et al. (2022) use steerable filters to obtain convolution layers with approximate translation symmetry and without rotation symmetry (**RSteer**), and with approximate translation and rotation symmetry (**RGroup**). 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. While equivariant architectures may allow reduced parameter counts due to weight-tying, in practice many of these architectures require considerable additional parameter counts to achieve competitive performance (see parameter counts in Section 5.1).

We also discuss a family of approaches which are not based around the CNN architecture. Residual Pathway Priors **(RPP)** (Finzi et al., 2021), is a model where each layer is 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 previous studies have considered group representations on the latent space, sometimes governed via an equivariance term in the loss function. An early approach by (Welling & Cohen, 2014) shows how geometrical transformations can be encoded on the latent space via SO(3) representations on the latent space, while (Worrall et al., 2017) demonstrate disentanglement phenomena with similar methods. 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. Recent work of (Bökman et al., 2024) considers learned latent representations for a fixed group to solve certain geometrical tasks. 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 to identify the regular representation as a theoretically-motivated latent structure, which enables our simple pipeline without additional learnable parameters. Although our approach requires fixing a group structure, our experimental results show that this approach can often achieve superior performance compared to models without this constraint, including models specifically adapted for continuous symmetries.

C 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., $v \in V$.

$f|_{\mathcal{S}}$ If $f : \mathcal{A} \to \mathcal{B}$ is a function and $\mathcal{S} \subseteq \mathcal{A}$, $f|_{\mathcal{S}}$ denotes the restriction of the function to $\mathcal{S}$.

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.

D 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 $\{1, a \mid a^2 = 1\}$. 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 $g \in G$.

E 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 $\{1, a \,|\, a^2 = 1\}$. Then $\widehat{\rho}_Z$ should satisfy $\widehat{\rho}_Z(1) = \mathrm{id}$ and $\widehat{\rho}_Z(a^2) = (\rho_Z(a))^2 = \mathrm{id}$. To achieve this, we fix the parameter $\widehat{\rho}_Z(1) = \mathrm{id}$, 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 $d \times d$:

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

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.

F Proofs

We first introduce some basic definitions

Definition 3. Let V be a vector space, and $W, W' \subseteq V$ be subspaces of V. W and W' are *linearly independent* if $W \cap W' = 0$.

Definition 4. Let G act on a set $\mathcal{X}$ via the group action α_X. The *orbit of* $x \in \mathcal{X}$ is the set $\mathcal{O}_x = \{\alpha_{\mathcal{X}}(g)(x) \,|\, g \in G\}$. Given a vector space Z and a function $E : \mathcal{X} \to Z$, we call the set $E(\mathcal{O}_x) = \{E(\alpha_{\mathcal{X}}(g)(x) \,|\, g \in G\}$ the *embedded orbit of* x *along* E. Two embedded orbits $E(\mathcal{O}_x), E(\mathcal{O}_{x'})$ are *linearly independent* if their spans are linearly independent, that is if $\mathrm{Span}(E_\theta(\mathcal{O}_x)) \cap \mathrm{Span}(E_\theta(\mathcal{O}_{x'})) = \{0\}$.

Definition 5. Let G be a group acting on the sets $\mathcal{A}$ and $\mathcal{B} \subseteq \mathbb{R}^d$ with the actions $\alpha_{\mathcal{A}}$ and $\alpha_{\mathcal{B}}$, respectively. Let $E : \mathcal{A} \to \mathcal{B}$ be a function of sets. We say that E is $(\alpha_{\mathcal{A}}, \alpha_{\mathcal{B}}, \varepsilon)$*-equivariant* if

$$\sup_{g \in G\ x \in \mathcal{X}} ||E(\alpha_{\mathcal{A}}(g)(x)) - \alpha_{\mathcal{B}}(g)(E(x))|| \leq \varepsilon$$

This quantity measures how far E is from being equivariant with respect to the actions $\alpha_{\mathcal{A}}$ and $\alpha_{\mathcal{B}}$. When clear from the context, we will drop the dependency on $\alpha_{\mathcal{A}}$.

Definition 6. Let V, W be vector spaces with norms $||\cdot||_V$ and $||\cdot||_W$ respectively, and let $A : V \to W$ be a linear map between them. The *operator norm* is defined by

$$||A||_{\mathrm{op}} = \sup_{x \neq 0} \frac{||Ax||_W}{||x||_V} = \max_{||x||_V = 1} ||Ax||_W$$

Then, given two linear maps A, B, the quantity $||A - B||_{op}$ gives a distance of functions.

The proof of Lemma 9 adapts the argument in Nikolaou et al. (2025), which uses measure-theoretic properties of analytic functions to demonstrate that transformers are almost everywhere injective. Although our focus here is not on transformers, most of their results require only real analyticity, and thus can be easily adapted to our case. Intuitively, a measure μ on a set X quantifies the 'size' or 'volume' of subsets within X (see e.g., Fremlin (2000) for a foundational treatment). In the context of $\mathbb{R}^p$, the Lebesgue measure λ corresponds to the standard notion of Euclidean volume (e.g., it assigns the unit hypercube a measure of 1).

Notation. If $f : X \to X$ is a function, we will write $f^{\circ T}$ to indicate the consecutive application of f for T times. If Θ is a set equipped with a measure μ, we will write $\theta \sim \Theta$ to indicate that θ is a random draw of an element of Θ according to the measure μ.

The following are well-known mathematical results and definitions.

Proposition 7. *Let $U \subseteq \mathbb{R}^m$ be open and connected, and let $f : U \to \mathbb{R}^n$ be an analytic function. If f is not identically zero, then its zero set $Z(f) := \{x \in U \mid f(x) = 0\} = f^{-1}(0)$ has Lebesgue measure zero in $\mathbb{R}^m$, i.e. $\lambda(Z(f)) = 0$.*

Definition 8. Let μ, ν be Borel measures on $\mathbb{R}^p$. We say that μ is *absolutely continuous* with respect to ν, written $\mu \ll \nu$, if for every Borel set U we have

$$\nu(U) = 0 \implies \mu(U) = 0 \tag{2}$$

Since the Lebesgue measure λ is the standard notion of Euclidean volume, then we can intuitively understand that $\mu \ll \lambda$ just when the measure μ assigns zero measure to every set with zero Euclidean volume. In this way, measures μ with $\mu \ll \lambda$ may behave in ways that accord with our intuition. Any standard sampling measure μ used to generate initial parameters for a neural network (e.g. from the normal or uniform distributions) will be likely to satisfy this property.

The following critical lemma underlies our theoretical results, and we can explain it intuitively as follows. If we have some set of parameters $\mathcal{W} \subseteq \Theta$ for our neural network which is measure zero, then of course if we initialise the network with some parameters θ at random, the probability that $\theta \in \mathcal{W}$ is zero. But we then ask, if we update the parameters by gradient descent for finitely many steps, what is the probability that the optimised parameters are within the set $\mathcal{W}$?

Lemma 9. *Let E_θ be a parametrized function that is analytic for all parameters $\theta \in \Theta \subseteq \mathbb{R}^p$. Assume that the parameters are randomly initialized according to a distribution μ that is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^p$, i.e. $\theta_0 \sim \mu$ with $\mu \ll \lambda$. Furthermore, assume an analytic loss function $\mathcal{L}$, and that the parameters are updated via gradient descent, i.e. $\theta_{t+1} := \Phi(\theta_t) := \theta_t - \eta \nabla \mathcal{L}(\theta_t)$, with $\eta \in (0, 1)$.*

Let $\mathcal{W} \subseteq \Theta$ with $\lambda(\mathcal{W}) = 0$. Then, for all $T \in \mathbb{N}$, $\mu(\{\theta_0 \mid \Phi^{\circ T}(\theta_0) \in \mathcal{W}\}) = 0$.

Proof. The proof uses standard analytic and measure-theoretic tools, and is an adaptation of the argument in the proof of (Nikolaou et al., 2025, Theorem C.1). Let $\mathcal{W} \subseteq \Theta$ with $\lambda(\mathcal{W}) = 0$. First, we apply (Nikolaou et al., 2025, Lemma C.6), that $\lambda(\Phi^{-1}(\mathcal{W})) = 0$. (Note that in this paper, Lemma C.6 assumes the context of a transformer; however the proof only uses analyticity of the components, and thus the result holds in our case). Then, because $\mu \ll \lambda$, it follows that $\mu(\{\theta_0 \mid \Phi(\theta_0) \in \mathcal{W}\}) = \mu(\Phi^{-1}(W)) = 0$. By applying the same argument T times, we find $\mu(\{\theta_0 \mid \Phi^{\circ T}(\theta_0) \in \mathcal{W}\}) = 0$. □

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 $\mathcal{S} \subseteq \mathcal{A}$. Let E_θ be a (ρ_Z, ε)-equivariant encoder and σ_{min} the smallest singular value of $E(\mathcal{S})$. If $\sigma_{\min} > 0$, there exists a representation $(V, \tilde{\rho})$ with $V \subseteq Z$ and $\tilde{\rho} \cong \rho_{\text{reg}}$ such that for all $g \in G$:

$$||\rho_Z(g)|_V - \tilde{\rho}(g)||_{\text{op}} \leq \tfrac{\varepsilon}{\sigma_{\text{min}}} \sqrt{|G|}$$

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 linearization $\mathbb{R}[\mathcal{S}]$ carries the structure of the regular representation. We write this representation explicitly as $\rho_{\mathbb{R}[\mathcal{S}]}(g)(\sum_s a_s s) := \sum_s a_s \alpha_\mathcal{A}(g)(s)$.

Define the linearization of E as the linear map $\tilde{E} : \mathbb{R}[\mathcal{S}] \to Z$ by mapping basis elements $e_s \mapsto E(s)$. The matrix representation of $\tilde{E}$ is exactly $M^\mathcal{S} = E(\mathcal{S})$, the matrix whose columns are given by $\{E(s)\}_{s \in \mathcal{S}}$. Since $\sigma := \sigma_{\min}(M^\mathcal{S}) > 0$ by assumption, $\tilde{E}$ is a linear isomorphism onto its image $V = \text{Im}(\tilde{E})$.

We define the action $\tilde{\rho}$ on V by pushing forward the regular representation through $\tilde{E}$:

$$\tilde{\rho}(g) := \tilde{E} \circ \rho_{\text{reg}}(g) \circ \tilde{E}^{-1}$$

By construction, $(V, \tilde{\rho}) \cong (\mathbb{R}[\mathcal{S}], \rho_{reg})$, so V carries the exact regular representation structure.

Let $\rho_V(g) := \rho_Z(g)|_V$ be the restriction of the linear map ρ_Z to $V \subseteq Z$. We now bound the difference between the actions $\tilde{\rho}$ and ρ_V. For any vector $v \in V$ with $\|v\| = 1$, we can write $v = \tilde{E}u$ for a unique $u \in \mathbb{R}[\mathcal{S}]$. Note that $\|u\| \leq \frac{1}{\sigma}\|v\| = \frac{1}{\sigma}$ by definition of σ.

We compare the actions:

$$\begin{aligned}\|\rho_V(g)v - \tilde{\rho}(g)v\| &= \|\rho_V(g)\tilde{E}u - \tilde{E}\rho_{reg}(g)u\| \\ &= \|(\rho_V(g)\tilde{E} - \tilde{E}\rho_{reg}(g))u\|\end{aligned}$$

The term in the parenthesis is the 'equivariance error' of the linear map $\tilde{E}$. For a basis vector e_s:

$$\|\rho_V(g)\tilde{E}e_s - \tilde{E}\rho_{reg}(g)e_s\| = \|\rho_V(g)E(s) - E(g \cdot s)\| \leq \varepsilon$$

where the last inequality is given by assumption. For a general vector $u = \sum a_s e_s$, we have:

$$\|(\rho_V(g)\tilde{E} - \tilde{E}\rho_{reg}(g))u\| \leq \sum |a_s|\varepsilon = \varepsilon\|u\|_1$$

Using the inequality $\|u\|_1 \leq \sqrt{|G|}\|u\|$, we get:

$$\|(\rho_V(g)\tilde{E} - \tilde{E}\rho_{reg}(g))u\| \leq \varepsilon\sqrt{|G|}\|u\| \leq \varepsilon\sqrt{|G|}\frac{1}{\sigma}$$

for all $g \in G$. Thus, the operator norm difference is bounded by $\frac{\varepsilon}{\sigma}\sqrt{|G|}$. □

We now combine Lemma 9 and Theorem 1 to obtain guarantees on the existence of regular representations in the latent space.

Theorem 2. Let G be a finite group acting on a set $\mathcal{A}$ and on a vector space Z with representation ρ_Z, where $\dim(Z) \geq |G|$. Let $\mathcal{S} \subseteq \mathcal{A}$ be an orbit on which G acts freely. Consider an encoder $E_\theta : \mathcal{A} \to Z$ parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$. Suppose E_θ is analytic with respect to θ and that the parameters are initialized $\theta_0 \sim \mu$ (with μ absolutely continuous w.r.t. Lebesgue measure) and updated via gradient descent on an analytic loss.

Then, exactly one of the following holds:

(i) The encoder orbit $E_\theta(\mathcal{S})$ is rank-deficient (rank $< |G|$) for all possible parameter values $\theta \in \Theta$.

(ii) With probability 1, the latent subspace $V = \text{span}(E_\theta(\mathcal{S}))$ is isomorphic to the regular representation. Specifically, the orbit forms a basis for V, and the induced action $\tilde{\rho}$ on V satisfies the bound from Theorem 1: $\|\rho_Z(g)|_V - \tilde{\rho}(g)\|_{\text{op}} \leq \frac{\varepsilon}{\sigma_{\min}}\sqrt{|G|}$

Proof. We analyze the rank of the matrix $M_\theta^{\mathcal{S}} = E_\theta(\mathcal{S}) \in \mathbb{R}^{\dim(Z)\times|G|}$ formed by the encoder outputs. The regular representation is instantiated if and only if these vectors are linearly independent, i.e., rank$(M_\theta^{\mathcal{S}}) = |G|$.

Let $d_I(\theta)$ denote the determinants of all possible $|G| \times |G|$ minors of $M_\theta^{\mathcal{S}}$. The condition of rank deficiency corresponds to the zero set $\mathcal{W} = \{\theta \in \Theta \mid \forall I, d_I(\theta) = 0\}$. Since the encoder E_θ is analytic, each determinant $d_I(\theta)$ is an analytic function of θ.

By the properties of analytic functions, the common zero set $\mathcal{W}$ is either the entire domain Θ or a set of Lebesgue measure zero.

(i) If $\mathcal{W} = \Theta$, the rank is strictly less than $|G|$ everywhere. The vectors are always linearly dependent, preventing the formation of the regular representation.

(ii) If $\mathcal{W} \neq \Theta$, then $\mathcal{W}$ has measure zero. Since the initialization μ is absolutely continuous and the gradient update map is analytic (preserving null sets), the probability of the trajectory entering or remaining in $\mathcal{W}$ is 0. Thus, with probability 1, rank$(M_\theta^{\mathcal{S}}) = |G|$ and V is isomorphic to the regular representation.

In case (ii), the full rank implies $\sigma_{\min}(M_\theta^{\mathcal{S}}) > 0$. By Theorem 1, this guarantees the existence of the subspace isomorphism and the validity of the stability bound. □

F.1 The analyticity condition

We remark that, as observed by Nikolaou et al. (2025), most standard modules used in neural network, such as linear layers, layer norm, skip connections, convolutions, attention, and

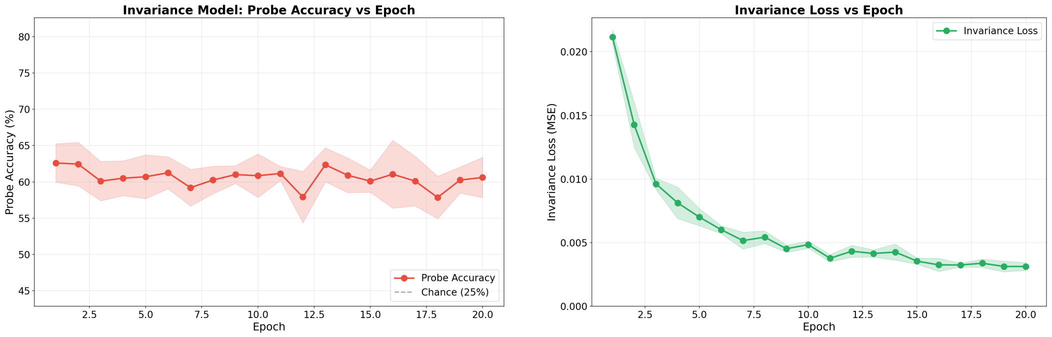


Figure 5: (Left) Global probe accuracy and (Right) invariance loss throughout the training of an invariant MLP. Despite the low invariance loss, the global probe maintains a significant accuracy (60.6%$\pm$2.8%), confirming that the regular representation persists in a decodable state rather than being structurally erased. Averages over 5 runs.

others are analytic. The same holds for many commonly used activation functions, such as tanh, sigmoid, softplus, softmax, SiLU, GELU, SwiGLU. Therefore, the analytic condition does not heavily restrict our analysis. For example, Nikolaou et al. (2025) highlight that decoder-only transformers are analytic. However, others activation functions are only piece-wise analytic, e.g. ReLU, LeakyReLU, ELU. For this reason, we repeat the TMNIST and MNIST experiments from Section 4.2 with non-analytic encoders to empirically test whether our conclusions hold for this class of networks. We find this to be the case, and we discuss it in Appendix H.3.

G Representational persistance

To evaluate the tension between the invariance objective and the theoretical persistence of the regular representation, we analyze the latent space of an encoder trained with an explicit invariance penalty: $\mathcal{L}_{\text{inv}} = \|E(x) - E(gx)\|^2$. We take Z to be the layer before the classification head, and set $d = \dim(Z) = 64$. For a given input x, we denote the latent embedding of its g-th transformation as $z_g = E(\alpha_{\mathcal{X}}(g)(x))$ and the orbit mean as $\bar{z} = \frac{1}{|G|}\sum_{g \in G} z_g$. We use these to define the residual vectors $r_g = z_g - \bar{z}$, which represent the portion of the signal that the encoder has failed to make invariant. We investigate whether the representation undergoes a structural transition to the trivial (invariant) representation. To quantify this, we use linear probes, a canonical framework in the interpretability literature (Alain & Bengio, 2018; Hewitt & Manning, 2019; Belinkov, 2021) used to assess whether specific latent properties are linearly decodable from the representation.

Local linear probes. We verify the existence of the regular representation by training a Local Linear Probe on each orbit. Given an orbit $\mathcal{O}_x = \{z_0, z_1, z_2, z_3\}$ in $\mathbb{R}^{64}$, we attempt to learn a linear mapping $W \in \mathbb{R}^{4\times 64}$ to the canonical basis $\{e_1, e_2, e_3, e_4\} \subset \mathbb{R}^4$. The probe is given via the normal equation:

$$W = YX^T(XX^T + \lambda I)^{-1}$$

where X is the matrix of orbit features and Y is the identity matrix. We find that for all orbits, the reconstruction accuracy is 100%, and the weights W remain well-conditioned. This empirically confirms that the residual vectors $\{z_g - \bar{z}\}$ are linearly independent, satisfying the full-rank condition $\sigma_{\min} > 0$ required by Theorem 2.

Global Linear Probes. To assess the consistency of this representation across the manifold, we train a Global Linear Probe. Unlike the local probe, which is fit to a single orbit, the global probe uses a single weight matrix W_{global} trained on the entire training set to predict group elements from embeddings. Specifically, they are trained to predict g from r_g at each training epoch. On the held-out set, the global probe achieves 60.6%$\pm$2.8% accuracy (vs. 25% chance), as shown in Figure 5. This performance gap between local (100%) and global (60.6%) probes indicates that while the symmetry information is preserved in every orbit, its orientation is not globally aligned. This suggests a 'twisted' geometry where the basis of the regular representation varies depending on the input x.

These results confirm that the attenuated signal is not just numerical noise, but a distinguishing feature that persists despite the training objective. It underscores our core insight: the network does not structurally transition to the trivial representation; rather, it maintains the full algebraic regular structure, instantiating the target geometry solely through geometric contraction.

H EXPLORATORY EXPERIMENTS

This section is organized as follows:

- Section H.1 describes how we extract the embedded orbits and check their linear independence to get the number of linearly independent orbits.
- Section H.2 contains further details for the exploratory experiments, including hyperparameters and regularization terms for the algebra loss.
- Section H.3 repeats the TMNIST and MNIST experiments from Section 4.2 but for non-analytic encoders.
- Section H.4 repeats the TMNIST experiment from Section 4.2 by varying the depth of the layer considered as Z.
- Section H.5 repeats the TMNIST experiment from Section 4.2 by changing the initialization scheme for the learnable group action $\widehat{\rho}_Z$.

H.1 EXTRACTING EMBEDDED ORBITS AND CHECKING THEIR LINEAR INDEPENDENCE

We describe how we extract the embedded orbits and how we compute their linear independence. Let $E : \mathcal{X} \to Z$ denote the encoder, and $G = \{g_1, \dots, g_n\}$ the finite group considered. First, we compute the embedded orbit as $E(\mathcal{O}_x) = \{\widehat{\rho}_Z(g_i)(E(x))\}_{i \in \{1,\dots,n\}} \subseteq Z$ with $|E(\mathcal{O}_x)| = |G|$. Then, given embedded orbits $E(\mathcal{O}_{x_1}), \dots, E(\mathcal{O}_{x_m})$, we collect all vectors in their union in a matrix $K \in \mathbb{R}^{m|G| \times d}$. These orbits are linearly independent if the matrix K is full rank, which is computed by checking that all its singular values are non-zero.

Each number in the 'Orbits' columns in the Tables from Sections 4.2 and Appendix H.3, H.4 and H.5 is the maximum number of linearly independent orbits found by randomly sampling combinations of training samples $x \in \mathcal{X}$. For each run, we sample 500 different combinations.

H.2 FURTHER DETAILS FOR THE 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 H.2.1, H.2.2 and H.2.3 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.

H.2.1 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 6. 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 6(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 $L_{\text{task}} =$ MSE, and we use a simple CNN autoencoder with hyperparameters given in Table 6. The architectural details can be found on the provided repository.

We use the following regularization term:

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

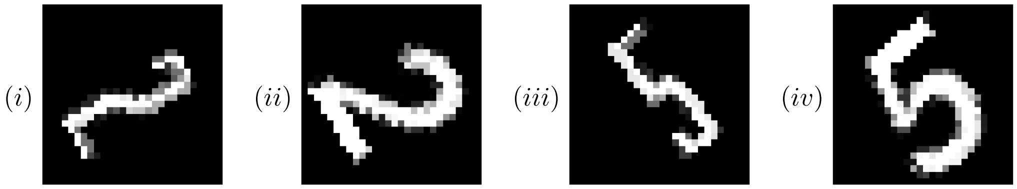


Figure 6: Examples of our augmented training dataset for the TMNIST experiment, from the chosen fonts ‘Bahianita-Regular’ (i), (iii) and ‘IBMPlexSans-MediumItalic’ (ii), (iv).

Table 6: Hyperparameters for exploratory experiments.

Experiment	Latent dim.	λ_a	λ_t	λ_e	LR	Batch Size
TMNIST C_2	8	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

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

H.2.2 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 120 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 regularization term:

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

We determined empirically that this regularization 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.

H.2.3 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 regularization 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 $\widehat{\rho}_Z(1)^{-1} = \texttt{torch.linalg.solve}(\widehat{\rho}_Z(1), \mathrm{I}_d)$ for efficiency and numerical stability. We determined empirically that this regularization helps to stabilize the training of $\widehat{\rho}_Z(1)$ and the behavior of its inverse.

H.3 Exploratory experiments for non-analytic encoders

In this section, we repeat the same experiments for TMNIST C_2 and MNIST D_3 from Section 4.2 but for non-analytic encoders. In particular, we use the same architecture but we replace the $\tanh$

activation function with ReLU. Table 7 shows similar results as the fully analytic encoders (Table 1), suggesting empirically that the optimization process avoids any potentially degenerate regions.

Table 7: Piece-wise analytic encoder experiments. 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

H.4 Exploratory experiments at different layer depths

In this section, we repeat the TMNIST experiment from Section 4.2 for different layer depths. In the original experiment, we choose to study equivariance with respect to the layer Z chosen as the central hidden layer (the output layer of the encoder). Table 8 shows the results of choosing Z as the first or last hidden layer. The results are similar to those in Section 4.2: each linearly independent embedded orbit corresponds to a copy of the regular representation, and the network tends to learn a multiple of it.

Table 8: TMNIST experiment with Z at different depths. Left: Z is taken as the first hidden layer; Right: Z is taken as the final hidden layer.

	Irrep. counts				
Run	-1	$+1$	Alg. loss	Eq. loss	Orbs.
1	3	5	4.9×10^{-10}	1.1×10^{-4}	3
2	3	5	4.2×10^{-9}	1.6×10^{-4}	3
3	4	4	1.0×10^{-10}	6.2×10^{-5}	4
4	4	4	2.3×10^{-6}	3.0×10^{-5}	4
5	3	5	9.0×10^{-10}	1.2×10^{-4}	3

	Irrep. counts				
Run	-1	$+1$	Alg. loss	Eq. loss	Orbs.
1	3	5	6.8×10^{-9}	4.5×10^{-4}	3
2	4	4	9.3×10^{-10}	3.7×10^{-4}	4
3	3	5	1.8×10^{-8}	4.5×10^{-4}	3
4	4	4	6.9×10^{-10}	4.6×10^{-4}	4
5	4	4	2.3×10^{-8}	4.4×10^{-4}	4

H.5 Exploratory experiment with different initialization

In this section, we repeat the TMNIST experiment from Section 4.2 but with a different initialization scheme. While Table 1 shows results for $\widehat{\rho}_Z$ initialized according to a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, Table 9 shows results for the same experiment with $\widehat{\rho}_Z$ initialized close to the identity as $\boldsymbol{I}_d + \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$.

The results confirm Theorem 1, as each linearly independent embedded orbit contributes one copy of the regular representation. However, the network typically does not learn a representation that consists entirely of a multiple of the regular representation. We observe that the trivial representation, corresponding to the eigenvalue $+1$ of $\widehat{\rho}_Z$ is over-represented. We hypothesize that the strong priming given by the initialization prevents a full exploration of the parameter space. To establish the practical advantage of the regular representation, we provide ablations with the trivial representation in controlled settings (Sections 5.1 and 5.1).

I Main experiments

Here we give details of the main experiments described in Section 5.1, which test our model of Section 5 on tasks using the DDMNIST, MedMNIST, SMOKE and SHREC‘11 datasets. Section I.1 discusses Cohen’s d-statistic, which we use to assess the effect size of our intervention. Sections I.2, I.3, I.4 and I.5 provide details of the datasets, architectures and hyperparameters that we use, together

Table 9: TMNIST experiment with $\widehat{\rho}_Z$ initialized close to the identity.

	Irrep. counts				
Run	-1	$+1$	Alg. loss	Eq. loss	Orbs.
1	2	6	3.1×10^{-5}	2.9×10^{-4}	2
2	2	6	9.5×10^{-4}	0.3×10^{-4}	2
3	2	6	9.5×10^{-4}	1.6×10^{-4}	2
4	2	6	4.6×10^{-5}	1.8×10^{-4}	2
5	2	6	2.4×10^{-5}	9.1×10^{-5}	2

with an effect size analysis. In all runs we use the Adam optimizer 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.

I.1 Cohen's d-statistic

Cohen's d-statistic is a widely-adopted metric (Miranda et al., 2025; Huang et al., 2024; Gundersen et al., 2023; Karandikar et al., 2021; Hermann et al., 2024) to assess effect size, i.e. the meaningfulness of the difference between distributions. In particular, Cohen's d quantifies the difference between two distributions in standard deviation units. Commonly used thresholds in machine learning are the following (Hermann et al., 2024):

- $|d| < 0.5$, small effect
- $0.5 \leq |d| < 0.8$, medium effect
- $0.8 \leq |d| < 1.2$, large effect
- $1.2 \leq |d|$, very large effect

Suppose we are given n_1 and n_2 observations of two distributions, with means $\overline{x}_1$ and $\overline{x}_2$, and standard deviations s_1 and s_2 respectively. Cohen's d is then defined as follows:

$$d = \frac{\overline{x_1} - \overline{x}_2}{s}, \qquad s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{6}$$

To assess the effect size of our model, we choose $\overline{x}_1$ to be the mean result of our model on a particular task, and $\overline{x}_2$ to be the mean result of a benchmark model. When reported in the tables below, we choose the sign of the effect value so that a positive value indicates our model performed better.

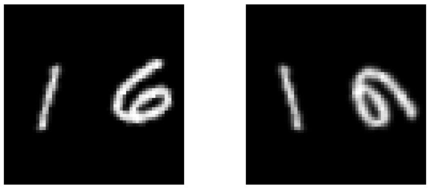
I.2 DDMNIST experiments

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 10. We give an example in Figure 7. 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 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.

Table 10: Symmetry groups and their actions on DDMNIST.

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



Before augmentation After augmentation

Figure 7: 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.

Hyperparameters. We report the hyperparameters used for the CNN and our model for the DDMNIST experiments in Table 11. 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.

Table 11: Hyperparameters for DDMNIST experiments.

	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

Effect size analysis. We report the effect size of our intervention in Table 12. For each model, the 'Effect' column reports the Cohen d-value, comparing that model against 'Ours' with the regular representation. We observe that, for each model considered, there is at least one task where the difference with our model is very large according to Cohen's d statistic (Appendix I.1).

I.3 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

Table 12: DDMNIST test accuracies and effect sizes. Mean over 3 runs; standard deviation in brackets. 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. Effect values compare to 'Ours (regular)', and a positive value means ours performed better. The annotations *, **, *** indicate medium, large and very large effect sizes respectively.

Model	$C_4 \uparrow$	Effect	$C_2 \uparrow$	Effect	$D_4 \uparrow$	Effect
CNN	0.907 (0.004)	2.0***	0.938 (0.006)	1.8***	0.800 (0.001)	43.0***
SCNN	0.484 (0.008)	68.2***	0.474 (0.003)	133.8***	0.431 (0.010)	60.6***
Restriction	0.914 (0.007)	0.2	0.890 (0.007)	10.0***	0.837 (0.020)	2.2***
RPP	0.908 (0.022)	0.4	0.903 (0.009)	6.3***	0.827 (0.020)	2.9***
PSCNN	0.909 (0.007)	1.1**	0.871 (0.016)	6.5***	0.842 (0.011)	3.3***
Trivial rep	0.874 (0.004)	10.0***	0.938 (0.007)	1.6***	0.819 (0.004)	15.5***
Defining rep	–		–		0.838 (0.010)	4.2***
Ours (regular)	**0.915** (0.004)		**0.947** (0.004)		**0.868** (0.002)	

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 13. 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.

Table 13: Hyperparameters for MedMNIST experiments.

	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

Effect size analysis. We report the effect size of our intervention in Table 14. For each model, the 'Effect' column reports the Cohen d-value comparing that model against 'Ours' with the regular representation. We observe that, for each model considered, there is at least one task where the difference with our model is very large according to Cohen's d statistic (Appendix I.1).

I.4 SMOKE EXPERIMENT

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

Table 14: MedMNIST3D test accuracies and effect sizes. Mean over 3 runs; standard deviation in brackets. Parameter counts shown. Best result in each column is bold, second-best is underlined. Effect values compare to 'Ours (regular)', and a positive value means ours performed better. The annotations *, **, *** indicate medium, large and very large effect sizes respectively.

Group	Model	Nodule ↑	Effect	Synapse ↑	Effect	Organ ↑	Effect
N/A	CNN	0.873 (0.005)	2.80***	0.716 (0.008)	9.26***	0.920 (0.003)	-7.01***
Aug	CNN	0.879 (0.007)	1.32***	0.761 (0.008)	1.54***	0.632 (0.005)	0.25
SO(3)	SCNN	0.873 (0.002)	3.68***	0.738 (0.009)	4.91***	0.607 (0.006)	0.88**
SO(3)	RPP	0.801 (0.003)	20.86***	0.695 (0.037)	2.86***	0.936 (0.002)	-7.42***
SO(3)	PSCNN	0.871 (0.001)	4.44***	**0.770** (0.030)	0.00	0.902 (0.006)	-6.53***
O(3)	SCNN	0.868 (0.009)	2.61***	0.743 (0.004)	8.54***	0.902 (0.006)	-6.53***
O(3)	RPP	0.810 (0.013)	7.82***	0.722 (0.023)	2.94***	**0.940** (0.006)	-7.48***
O(3)	PSCNN	0.873 (0.008)	2.10***	0.769 (0.005)	0.26	0.905 (0.004)	-6.62***
$\mathrm{Sym}_{\mathrm{cube}}$	Trivial rep	0.867 (0.001)	5.55***	0.743 (0.002)	13.50***	0.571 (0.002)	1.79***
$\mathrm{Sym}_{\mathrm{cube}}$	Defining rep	0.837 (0.013)	5.08***	0.756 (0.019)	1.04**	0.560 (0.025)	1.89***
$\mathrm{Sym}_{\mathrm{cube}}$	Ours (regular)	**0.887** (0.005)		**0.770** (0.002)		0.642 (0.056)	

(x, y) components of a velocity field (see Figure 8 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. In line with Wang et al. (2022), we consider the group C_4 acting on the data by 90° rotations and reorientation of the velocity field, as illustrated in Figure 9.

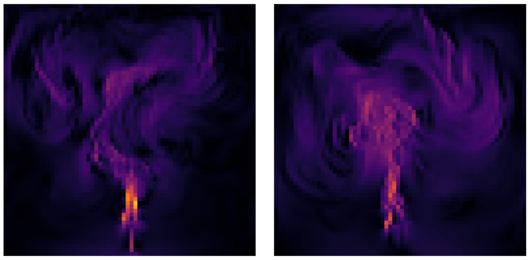


Figure 8: Approximately equivariant dynamics of smoke plumes Holl et al. (2020).

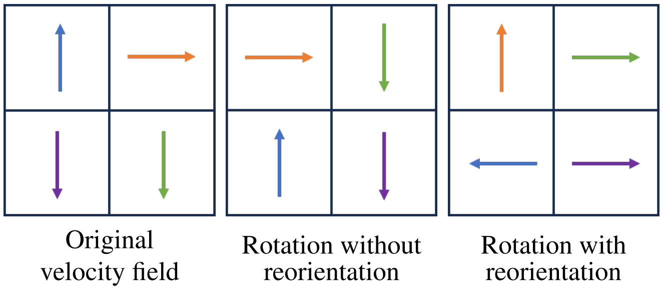


Figure 9: Examples of a velocity field and its augmentations with and without reorientation. Rotating by 90° counterclockwise without reorienting simply moves the spatial grid, but breaks the physical meaning of the underlying system.

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×height×width), we choose a representation of C_4 given by the regular representation in each channel separately.

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.

Effect size analysis. We report the effect size of our intervention in Table 15. For each model, the 'Effect' column reports the Cohen d-value comparing that model against 'Ours' with the regular representation. We observe that, for each model considered, there is at least one metric where the difference with our model is very large according to Cohen's d statistic (Appendix I.1).

Table 15: Test RMSE and effect for the SMOKE dataset. Effect values compare to 'Ours', and a positive value means ours performed better. The annotations *, **, *** indicate medium, large and very large effect sizes respectively.

Group	Model	Future ↓	Effect	Domain ↓	Effect
N/A	CNN	0.81 (0.01)	3.0***	0.63 (0.00)	2.8***
Aug	CNN	0.83 (0.03)	2.2***	0.67 (0.06)	1.4***
N/A	MLP	1.38 (0.06)	14.0***	1.34 (0.03)	32.6***
C4	E2CNN	1.05 (0.06)	6.3***	0.76 (0.02)	9.5***
C4	RPP	0.96 (0.10)	2.5***	0.82 (0.01)	21.0***
C4	Lift	0.82 (0.01)	4.0***	0.73 (0.02)	7.6***
C4	RGroup	0.82 (0.01)	4.0***	0.73 (0.02)	7.6***
C4	RSteer	0.80 (0.00)	2.8***	0.67 (0.01)	6.0***
C4	PSCNN	**0.77** (0.01)	-1.0**	**0.57** (0.00)	-5.7***
C4	Ours	0.78 (0.01)		0.61 (0.01)	

I.5 SHREC '11 EXPERIMENT

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 $\lambda' = 0.8$,

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

Effect size analysis. We report the effect size of our intervention in Table 16. For each model, the 'Effect' column reports the Cohen d-value comparing that model against 'Ours' with the regular representation. We observe that, for the augmented baseline and NFT, the difference with our model is very large according to Cohen's d statistic (Appendix I.1). The same analysis reveals that NIso and our model are essentially equivalent on this task.

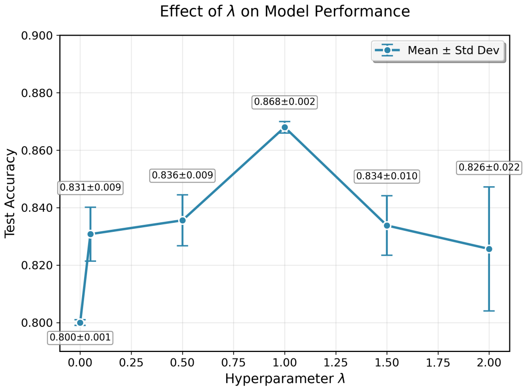
Table 16: Test accuracies and effect for the SHREC '11 dataset. Effect values compare to 'Ours', and a positive value means ours performed better. The annotations *, **, *** indicate medium, large and very large effect sizes respectively.

Model	Acc. ↑	Effect
NIso Mitchel et al. (2024)	90.26 (1.27)	0.1
NFT Koyama et al. (2024)	83.24 (2.03)	3.5***
AE with aug	69.36 (2.81)	8.5***
MC Mitchel et al. (2022)	86.5	–
Ours	**90.45** (2.1)	

J 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 10, 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 10: Mean accuracy and standard deviation (over 5 runs) for different values of λ on the DDMNIST D_4 task. $\lambda = 0$ is the baseline.


Effect of λ on Model Performance
Mean ± Std Dev
0.800±0.001
0.831±0.009
0.836±0.009
0.868±0.002
0.834±0.010
0.826±0.022
Test Accuracy
Hyperparameter λ