How Do Transformers Learn to Represent Symmetries?

Eduardo Santos-Escriche*
Technical University of Munich
Munich Center for Machine Learning

Valerie Engelmayer*
Technical University of Munich
Munich Center for Machine Learning

Ya-Wei Eileen Lin
Technical University of Munich
Munich Center for Machine Learning

Stefanie Jegelka
Technical University of Munich
Munich Center for Machine Learning
Munich Data Science Institute

Abstract

Training Transformer-based architectures with finite data augmentation has become an increasingly popular approach in geometric machine learning. Despite its empirical success, the interplay between the Transformer architecture, invariance to different symmetries, and augmentation budgets remains underexplored. In this paper, we study the ability of a vanilla Transformer to learn various symmetries through finite data augmentation for point cloud datasets. We identify an ordering of increasing *learnability* across the following symmetry groups: *(i)* non-angle-preserving symmetries, *(ii)* angle-preserving symmetries, and *(iii)* base angle-preserving subgroups, such as translation, rotation, and scale. For the base angle-preserving groups, we further investigate the Transformer's extrapolation behavior and conduct a structural analysis of the trained models, allowing us to identify interpretable mechanisms that induce invariance. Finally, we extend our analysis to equivariant functions and show that the detected mechanisms for approximate invariance can also provide a key building block for learned equivariance. Our implementation is available at https://github.com/valerie-engel/transformers_learn_symmetries.

1 Introduction

Data augmentation has been increasingly used as a common technique to induce approximate symmetry in machine learning models, since it provides practitioners with a flexible and model-agnostic approach that can be straightforwardly combined with off-the-shelf architectures. In particular, it has become popular in various scientific fields such as materials science [1], computational chemistry [2], structural biology [3, 4], as well as in computer vision [5], robotics [6] and reinforcement learning [7]. A notable example is the departure of AlphaFold3 [3] from the hard architectural equivariance of AlphaFold2 [8], and many other works have recently followed suit, e.g., MCFs [9], TransIP [2], Zatom-1 [10], or TABASCO [11].

While this technique has been extensively studied in the literature assuming an infinite number of augmentations [12, 13], this assumption strays far from reality: compute budgets and thus the possible number of augmentations are *finite*. A model only sees a limited number of transformed views of each sample, and the resulting symmetry is *approximate*. Recent theoretical work has begun to study this finite data augmentation regime [14, 15], however, a central question still remains open: *which architectures can efficiently learn which symmetries from finite data augmentation?*

*Equal contribution. | Project page.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

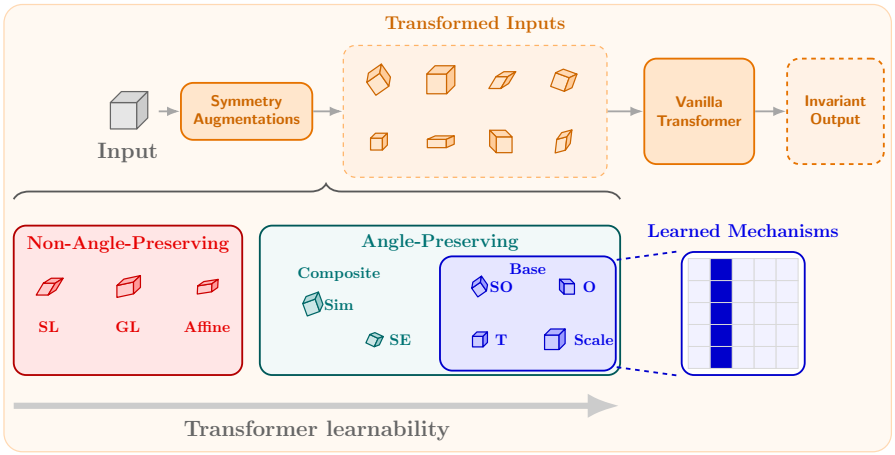


Figure 1: **Overview of our study and main findings.** Under finite data augmentation, vanilla Transformers learn invariance more efficiently for angle-preserving symmetries, especially their base groups, than for non-angle-preserving transformations. For these base symmetries, we identify *interpretable mechanisms* underlying the learned approximate invariance in the trained models.

This question is especially relevant for *Transformers* [16, 17]. The popularity and scalability of Transformers, together with their empirical success on downstream tasks, have been widely accepted as sufficient justification for their consistent adoption as the core architecture for many augmentation-based approaches. However, the interplay between their architecture, data, and augmentation budgets for the learned symmetries has not yet been explored. Their broad applicability does not address whether Transformers are equally capable of learning all types of symmetries from data augmentation, or whether their architecture is better aligned with some transformation groups than others.

In this paper, we investigate this question through a systematic study of a vanilla Transformer's ability to learn different symmetries under limited data augmentation budgets, and examine how these learned symmetries are represented in trained models. Our main contributions are the following:

- We systematically investigate the ability of a vanilla Transformer to learn invariance to different symmetries from limited data augmentation and identify its special affinity for angle-preserving symmetries and, in particular, *base groups*, e.g., translation, rotation, and scale.
- We evaluate the generalization of learned angle-preserving symmetry to transformation ranges that are not observed during training, demonstrating that Transformers can acquire invariant behaviors that extend to unseen transformation ranges.
- We analyze trained Transformers and uncover interpretable mechanisms that implement invariance to each base angle-preserving symmetry, which we further connect to invariant theories and to previously observed empirical phenomena.
- We show that the identified mechanisms underlying approximate invariance in Transformers can also provide key building blocks for learning equivariance.

2 Related work

Incorporating symmetries into machine learning models has been explored for a wide range of symmetry groups, including permutations in sets [18], translations [19], roto-translations [20], or local gauge transformations [21]. Among the multiple equivariant machine learning paradigms, the two most common ones are: *(i) hard equivariance through architectural constraints*, where layers are designed to be equivariant to the desired symmetry (e.g., GCNN [22] or EMLP [23]), and *(ii) approximate equivariance through data augmentation* [24–27, 12, 28], where symmetry-transformed samples are incorporated into the training distribution. Beyond these paradigms, frame averaging [29–31] seeks to achieve equivariance by averaging over a small input-dependent subset of the full group. Another approach, canonicalization [32–35], implements equivariance by defining a map from

$\mathrm{O}(d) = \{\boldsymbol{Q} \in \mathbb{R}^{d\times d} : \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_d\}$	$\mathrm{Scale}(d) = \{s\boldsymbol{I}_d : s \in \mathbb{R}_{>0}\}$
$\mathrm{SO}(d) = \{\boldsymbol{Q} \in \mathbb{R}^{d\times d} : \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_d, \det(\boldsymbol{Q}) = 1\}$	$\mathrm{SE}(d) = \mathrm{T}(d) \rtimes \mathrm{SO}(d)$
$\mathrm{T}(d) = \{\boldsymbol{w} \in \mathbb{R}^d\}$	$\mathrm{Sim}(d) = \mathrm{T}(d) \rtimes (\mathrm{Scale}(d) \times \mathrm{SO}(d))$
$\mathrm{SL}(d) = \{\boldsymbol{A} \in \mathbb{R}^{d\times d} : \det(\boldsymbol{A}) = 1\}$	$\mathrm{GL}(d) = \{\boldsymbol{A} \in \mathbb{R}^{d\times d} : \det(\boldsymbol{A}) \neq 0\}$ $\mathrm{Aff}(d) = \mathrm{T}(d) \rtimes \mathrm{GL}(d)$

Table 1: The symmetry groups studied in this work. Symmetries are separated into angle-preserving (*top*) and non-angle-preserving transformations (*bottom*).

all inputs that are equivalent under the symmetry group to the same canonical form. Symmetry can also be incorporated through multi-task learning [36].

Especially related to our work are the numerous extensions of standard Transformer architectures designed to enforce equivariance with respect to different symmetries, including SE(3) [20, 37–39], Lie groups [40], geometric algebras [41, 42], local gauge transformations [43], and the combination of continuous translations and Platonic symmetries [44]. Such approaches can improve data efficiency and generalization by incorporating known structural priors into the model [45, 22, 46]. However, enforcing exact equivariance also introduces several limitations, including increased implementation complexity, scalability challenges, and an overly restrictive hypothesis space. The latter is specifically relevant when the underlying symmetry is only approximate, for example because of noise, discretization effects, or other imperfections in the data. These limitations have motivated a closer examination of the practical benefits of exact equivariance at scale [47] and have contributed to a recent shift toward alternative formulations of symmetry-aware models in scientific tasks with large amounts of data. Besides the data augmentation methods discussed in §1, one increasingly explored direction is to encourage equivariance through additional loss terms to explicitly bias the models towards equivariance, e.g., Orb-v3 [48], REMUL [2], and LieAugmenter [49].

3 Finite augmentation reveals a symmetry inductive bias in Transformers

Motivated by recent work increasingly combining Transformer-based models with data augmentation to learn representations that remain unchanged under specific transformations, we investigate whether Transformers can learn different types of symmetries in controlled settings. Specifically, we ask whether increasing the number of augmented views during training is sufficient for a Transformer to learn invariance across a broad range of transformation groups (see Fig. 2 for the groups we study).

Architecture. Seeking that our analysis faithfully reflects the ability of Transformers to learn these symmetries, we consider a small vanilla Transformer architecture, without positional encodings or masks, consisting of two blocks, each with a single attention head (see Eq. (3) for a precise formulation). We explore the effect of incorporating additional architectural components in §C.3.

Symmetries. In this analysis, we study multiple symmetry groups that are common across many different domains, which we characterize in Table 1. Invariance to any of those groups G is defined as the output of the Transformer Ψ remaining unchanged under the action of said group on the input domain $\mathcal{X} \subseteq \mathbb{R}^d$, i.e., $\forall g \in G, x \in \mathcal{X}$ we have $\Psi(g \cdot x) = \Psi(x)$.

Datasets. We study two complementary datasets. First, we build a synthetic topology-based dataset, called *3D Topology*. This dataset consists of five classes of point clouds exhibiting distinct topological properties, including genus, connectivity, and ambient entanglement (see bottom-left of Fig. 3 for an illustration). The labels are invariant to all the considered transformations. Thus, it serves to evaluate whether data augmentation can induce invariance to these larger symmetry groups. Second, we consider ModelNet10 [50], a widely used 3D shape classification dataset containing point clouds for 10 object categories. The task is to predict the category from point coordinates, and the label is invariant to global rigid motions and point permutations. Further details on the datasets are in §B.3.

Metrics. To evaluate whether a Transformer learns a given symmetry through data augmentation, we use two complementary criteria. First, we measure the downstream classification accuracy on a test set formed by randomly transforming the prototype samples according to the same symmetry group used during training. Second, we quantify the learned invariance using:

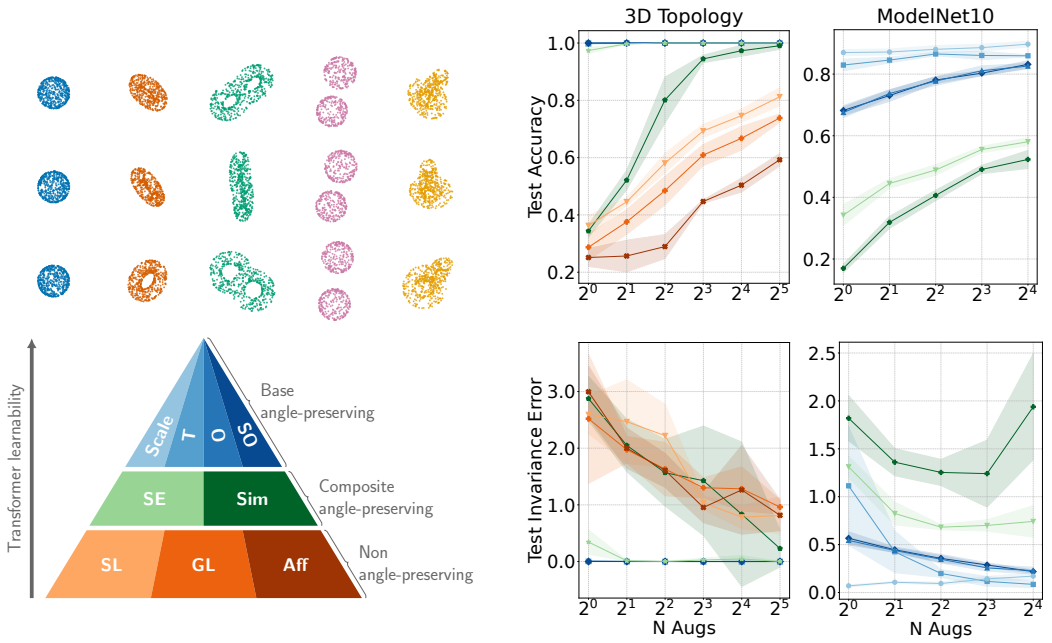


Figure 2: (*Left*) Samples from different classes of the 3D Topology dataset (top), and legend for the plots on the right, summarizing our findings (bottom). (*Right*) Mean $\pm$ std for test accuracies (top) and invariance errors (bottom) across 5 random seeds. Angle-preserving symmetries are *efficiently* learned by the Transformer, while it struggles with non-angle-preserving ones.

$\frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\mathrm{KL}\left(\Psi(x_i)\,||\,\Psi(g_j x_i)\right)$, which contrasts the class-probability distributions for pairs of original and augmented samples. We also analyze alternative invariance errors in §B.2. The model is trained only with the classification objective on augmented samples and the KL-based invariance error is used solely as a diagnostic metric to track invariance and is not part of the training objective. High accuracy together with low invariance loss therefore provides evidence that the model has learned an approximately invariant classifier for the corresponding symmetry.

Results. Fig. 2 reveals a clear separation between two families of symmetries. On the one hand, transformations associated with angle-preserving symmetries and their subgroups, including $\mathrm{SO}(3)$, $\mathrm{O}(3)$, $\mathrm{T}(3)$, $\mathrm{SE}(3)$, $\mathrm{Scale}(3)$, and $\mathrm{Sim}(3)$, are learned *efficiently*. For these groups, only a small number of augmentations is required to achieve high test accuracy and low invariance loss. On the other hand, non-angle-preserving transformation groups such as $\mathrm{Affine}(3)$, $\mathrm{SL}(3)$, and $\mathrm{GL}(3)$ require substantially larger augmentation budgets. Even with many augmentations, their accuracy remains lower and their invariance loss higher than what is observed for the angle-preserving groups.

The ModelNet10 results in Fig. 2 show a more detailed stratification. The base groups $\mathrm{T}(3)$ and $\mathrm{Scale}(3)$, achieve the strongest performance in both test accuracy and invariance loss. Their performance is also stable as the number of augmentations increases, suggesting that these transformations are learned with a small augmentation budget. The other base groups $\mathrm{SO}(3)$ and $\mathrm{O}(3)$, also achieve strong performance, but require a larger augmentation budget. Finally, the Transformer struggles more with compositional groups formed by combining those base transformations, i.e., $\mathrm{SE}(3)$ and $\mathrm{Sim}(3)$. Thus, the fact that the model can efficiently learn individual base symmetries does not automatically imply an ability to learn their compositions with the same efficiency.

Finding 1: A hierarchy in the learnability of symmetries with Transformers

Experimental evidence reveals the following ordering of symmetry families from easiest to hardest to learn for a Transformer:

1. base angle-preserving groups
2. compositional angle-preserving symmetries
3. non-angle-preserving symmetries

Based on these observations, below we focus on the base angle-preserving groups. This allows us to better isolate how each symmetry is learned and which mechanisms may support it.

4 Learning base angle-preserving symmetries

So far, we have observed that a Transformer can learn base angle-preserving symmetries efficiently when trained with augmentations sampled from the same support as test augmentations. To more rigorously investigate its ability to learn to extrapolate within the symmetries, we conduct further experiments testing in- and out-of-distribution generalization on ModelNet10 for each of these groups.

OOD setting. In practice, the augmentation budget may not cover the full range of transformations that the model encounters at test time. Thus, a learned invariance might need to hold on transformation ranges not seen at training time. This is especially critical for non-compact groups, e.g., translation and scale, where the magnitude of the augmentation could change drastically at test time.

To explore this scenario, we define two disjoint augmentation ranges corresponding to transformations of different magnitudes. During training, the model is exposed only to augmentations sampled from the smaller-magnitude range. We then consider two augmented versions of the test set: one using transformations from the same range as training, which we refer to as the *test set*, and one using transformations from a disjoint, larger-magnitude range, which we refer to as the *generalization set*. Further details are provided in §B.4.

Compared models: DeepSets and PointNet. To contextualize the performance of the considered vanilla Transformer architecture, we investigate whether other permutation invariant coordinate-token set architectures, i.e., DeepSets [18] and PointNet [51], can efficiently learn the base angle-preserving groups as well. PointNet is a more specialized point cloud model that extends DeepSets by so-called T-nets, which predict an affine matrix that is multiplied to the current representation matrix. Thus they can learn to canonicalize orientation with respect to the group O. For architectural details, see §B.4. Note that our goal is not to obtain SOTA performance, but to isolate how well each model can learn different invariances.

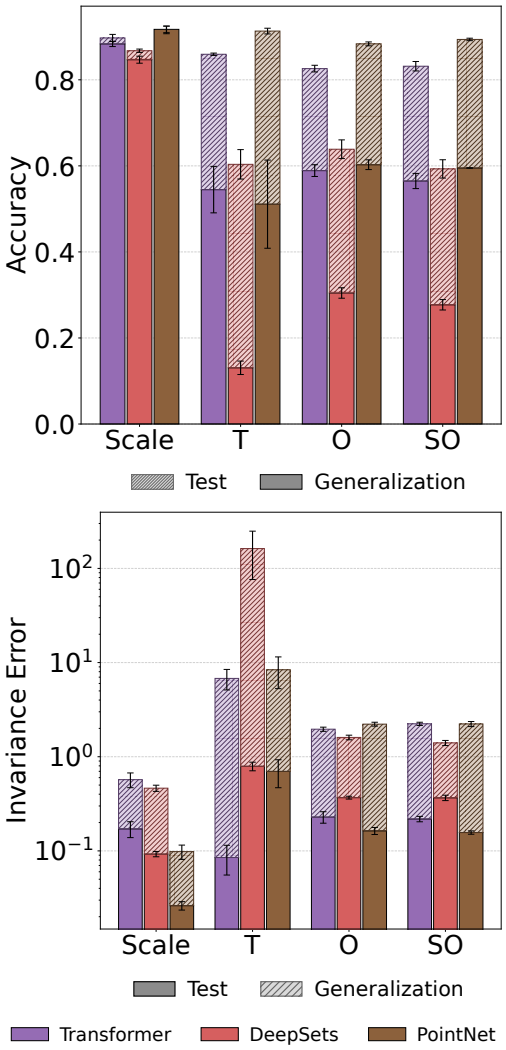


Figure 3: Accuracy (*top*) and invariance error (*bottom*) for different architectures on base angle-preserving symmetries under OOD evaluation. DeepSets exhibits the weakest performance, PointNet the strongest with Transformers as close second. Across architectures, scaling invariance appears easiest to learn, whereas translation invariance appears most difficult.

Results. Fig. 3 shows that scaling poses no major challenge to any of the models in or out of distribution. Differences become more pronounced for the other groups. DeepSets struggles to express any of the other invariances, with by far the lowest test accuracies across groups and poor extrapolation – in the case of translation, prediction is almost reduced to random guessing. Introducing T-nets drastically improves the overall performance of PointNet over DeepSets as expected. Maybe more surprisingly, the Transformer falls only slightly short of the accuracy of PointNet for rotations and reflections. Hence the point-wise interaction of self-attention supports these transformations almost as reliably as PointNet's dedicated canonicalization modules. We observe that, for translations, Trans-

formers exhibit significantly smaller invariance errors and a little higher generalization accuracy than the competing PointNets.

Finding 2: Extrapolation on base angle-preserving symmetries

For base angle-preserving symmetries, the Transformer learns to become invariant to the group structure beyond the limited augmentation ranges seen during training.

5 Mechanisms that can induce invariance to angle-preserving symmetries

Motivated by *Finding 2*, we study how Transformers can learn features that are invariant to rotation and reflection, translation, and scaling. For each group, we train ablations with only the modules involved in the respective mechanism and verify that they can indeed become invariant. In addition, we identify the signature of each mechanism in the activations of full vanilla Transformers trained with augmentations from the corresponding group. We refer to §C.4 for a comprehensive overview.

The main components of the Transformer are central to our analysis. We first define the self-attention module. For an input $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, the self-attention module is defined by

$$\mathrm{SA}(\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})\boldsymbol{V}, \qquad \mathrm{A}(\boldsymbol{X}) = \mathrm{softmax}\left(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d}\right), \tag{1}$$

where $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$, with parameters $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times d}$. We next consider LayerNorm, which normalizes each token independently. For the i-th token, it computes

$$\mathrm{LN}(\boldsymbol{X}_i) = \boldsymbol{w} \odot \frac{\boldsymbol{X}_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \boldsymbol{b}, \qquad \mu_i = \frac{1}{d}\sum\nolimits_{j=1}^{d} \boldsymbol{X}_{ij}, \qquad \sigma_i^2 = \frac{1}{d}\sum\nolimits_{j=1}^{d} (\boldsymbol{X}_{ij} - \mu_i)^2 \tag{2}$$

with learnable parameters $\boldsymbol{w}, \boldsymbol{b} \in \mathbb{R}^d$, where $\odot$ denotes elementwise multiplication and $\epsilon > 0$ for numerical stability. Finally, our analysis also relies on the residual connections. Let MLP denote a two-layer MLP with ReLU activations. A single Transformer block then computes

$$\boldsymbol{Y} = \boldsymbol{X} + \mathrm{SA}(\mathrm{LN}(\boldsymbol{X})), \qquad \boldsymbol{X}' = \boldsymbol{Y} + \mathrm{MLP}(\mathrm{LN}(\boldsymbol{Y})). \tag{3}$$

A full Transformer sequentially forwards through an in-projection, multiple such blocks, and a final out-projection. To isolate the mechanisms identified in our analysis, we use a single attention head and omit masks and positional encodings throughout this work. We present additional experiments with different positional encoding methods and multi-head attention setup in §C.3.

By construction, this architecture is permutation equivariant. We next study how it can learn invariance to rotations, translations, and scaling.

5.1 Rotation and reflection: stable attention over scalar relations

We first consider symmetries that fix the origin. Let $\mathrm{O}(d) = \{\boldsymbol{R} \in \mathbb{R}^{d\times d} : \boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}_d\}$ act on a point cloud $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ by right multiplication, $\boldsymbol{X} \mapsto \boldsymbol{X}\boldsymbol{R}$. This group contains both rotations $\mathrm{SO}(d)$ and reflections. A natural way to build rotation- and reflection-invariant features is to use scalar quantities. By the first fundamental theorem of invariant theory for $\mathrm{O}(d)$, polynomial scalar functions of vector inputs that are invariant under $\mathrm{O}(d)$ can be written as functions of pairwise inner products. Equivalently, for point clouds, the Gram matrix $\boldsymbol{G}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{X}^\top$ contains the canonical scalar information preserved by rotations and reflections. This view is in line with Villar et al. (2021) [52], where the invariant scalar products and scalar contractions provide a universal way to parameterize invariant and equivariant polynomial models for $\mathrm{O}(d)$ and related groups.

A Transformer can expose this type of scalar information through its attention scores. Indeed, the pre-softmax attention scores can be written as

$$\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d} = \boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\top/\sqrt{d}, \qquad \boldsymbol{B} = \boldsymbol{W}_Q\boldsymbol{W}_K^\top. \tag{4}$$

Thus each score is a learned bilinear pairing $\boldsymbol{x}_i^\top\boldsymbol{B}\boldsymbol{x}_j$. In the special case $\boldsymbol{B} = \alpha\boldsymbol{I}_d$, the scores are proportional to the Gram matrix and are invariant to every $\boldsymbol{R} \in \mathrm{O}(d)$. More generally, exact isotropy of $\boldsymbol{B}$ is not necessary on a finite augmentation orbit: because attention is row-wise softmax-normalized, only score differences within each row must be preserved.

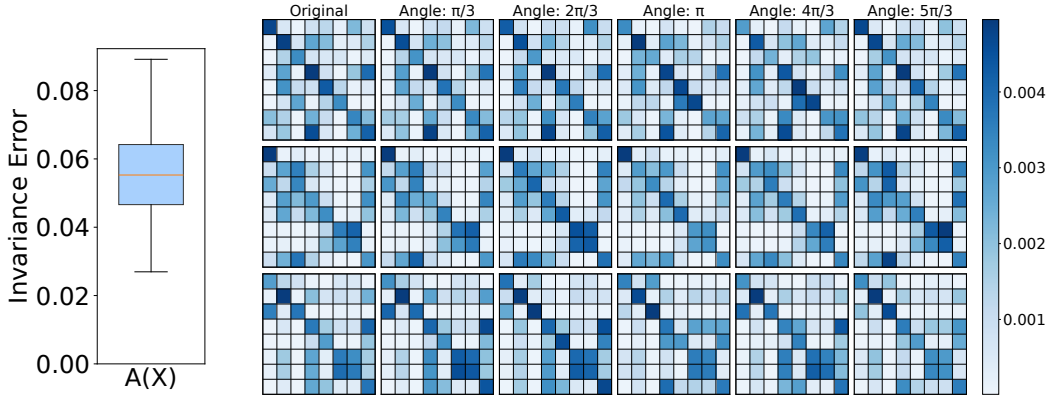


Figure 4: Attention maps of a full Transformer trained on ModelNet10 augmented with 16 rotational augmentations per sample remain highly stable across augmentations. (*Left*) Invariance error of A across the entire test set. (*Right*) Examples of the first 8×8 attention scores. Rows correspond to different samples, and columns correspond to rotations by the indicated angles.

Lemma 1. *Let* $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ *and let* $\boldsymbol{B} = \boldsymbol{W}_Q \boldsymbol{W}_K^\top \in \mathbb{R}^{d\times d}$ *and* $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}) = \frac{1}{\sqrt{d}} \boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\top$. *Let* $\boldsymbol{P}_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ *denote the centering matrix acting on the columns of a score matrix. Then, for any* $\boldsymbol{R} \in \mathrm{O}(d)$, *the following are equivalent:* $\mathrm{A}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X})$, *and* $(\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R}) - \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}))\,\boldsymbol{P}_n = \mathbf{0}$. *Equivalently, we have*

$$\boldsymbol{X}\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{X}^\top \boldsymbol{P}_n = \mathbf{0}. \tag{5}$$

In row-wise notation, this means that for all query indices i *and all key indices* j, k, $\boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)(\boldsymbol{x}_j - \boldsymbol{x}_k) = 0$. *Thus attention-map invariance along an augmentation orbit only requires the row-wise score differences to be invariant.*

We next test whether trained models exhibit this scalar-score mechanism. We train a vanilla Transformer on ModelNet10 with 16 rotated augmentations per sample and evaluate its attention maps across the augmentation orbit of held-out inputs. Fig. 4 shows representative attention maps for several inputs and their rotations. The maps remain nearly unchanged across rotations, and the corresponding invariance errors are small. This behavior matches the activation-level signature predicted by Lemma 1: the learned score differences are approximately preserved under rotations of the input.

5.2 Translation: constructing relative coordinates

Several mechanisms can make a neural network invariant to translation. For example, Villar et al. (2021) [52] show that subtracting one of the points as a landmark to obtain relative coordinates, $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \tilde{f}(\boldsymbol{x}_2 - \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n - \boldsymbol{x}_1)$, is a universal approximator when $\tilde{f}$ is sufficiently expressive. However, they do not test whether Transformers learn this mechanism. Alternatively, Cordonnier et al. (2020) [53] show that when using relative positional encodings, self-attention can become convolution-like by focusing attention on local neighborhoods. Since many forms of positional encoding are used in practice, we remain agnostic to a specific choice and omit positional encodings from our analysis. A third option could be to center the input by setting $\boldsymbol{W}_Q = \boldsymbol{W}_K = \mathbf{0}_{d\times d}$ and $\boldsymbol{W}_V = -\boldsymbol{I}_d$ in a self-attention block, which would deduct the mean from the residual stream.

To identify whether any of the above suggestions matches the actual computation carried out by a Transformer trained with translations, we examine the attention maps $\mathrm{A}(\boldsymbol{X})$ for three different test samples and their augmentations in Fig. 6 (*right*). In each case, we see that all attention

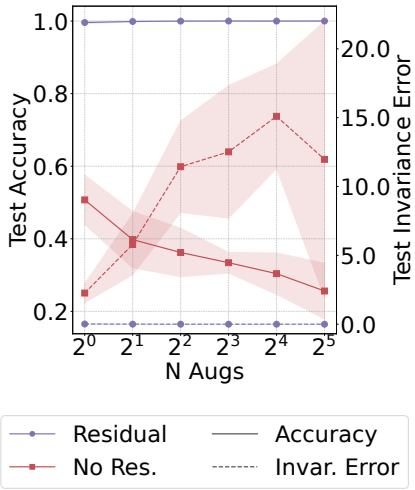


Figure 5: A model with 2 blocks of $\boldsymbol{X} + \mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ (*Residual*) can learn to be translation invariant (dashed) and solve the 3D Topology task (solid lines), but $\mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ (*No Res.*) cannot.

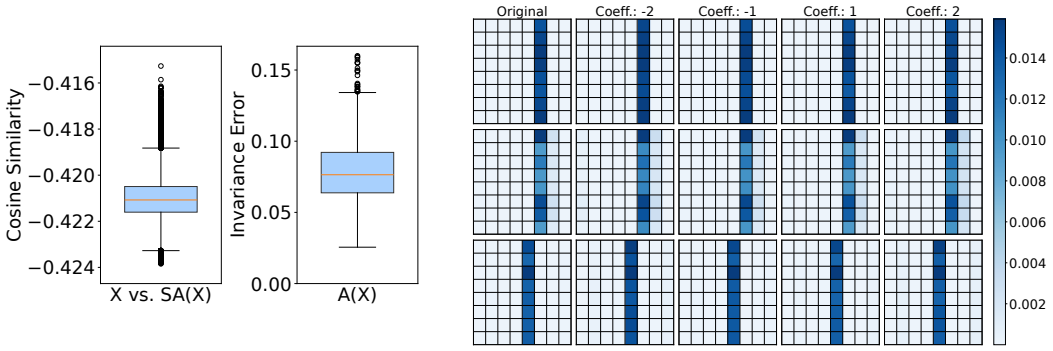


Figure 6: Results for the first block of a full Transformer trained on ModelNet10 with 16 translation augmentations per sample. (*Left*) Cosine similarity between $\boldsymbol{X}$ and $\mathrm{SA}(\boldsymbol{X})$ is strictly negative, indicating that the attention output has signs opposite to those of the residuals. (*Middle*) The attention maps have low invariance error across the full test set. (*Right*) Each row shows the first 8×8 attention scores of a sample and four of its translated augmentations spanning the training range. A translation coefficient of, e.g., 2 means the points are shifted by 2 in the positive direction for each axis. Samples from the same orbit often focus their attention on the same landmark point.

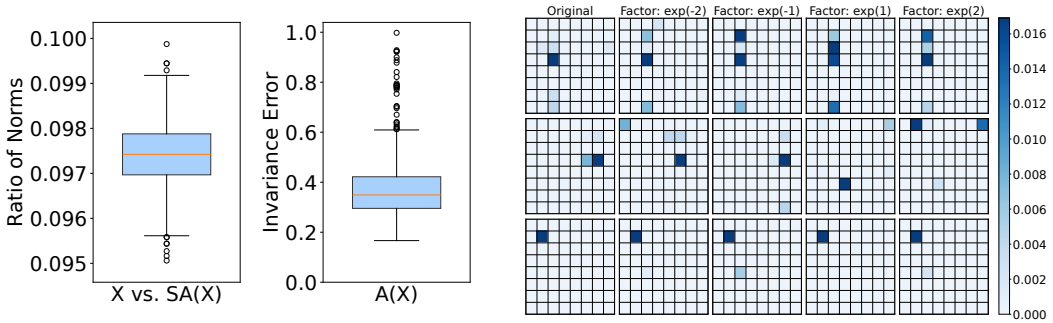


Figure 7: Mechanism indicators for the second block of a Transformer trained on ModelNet10 with 16 scale augmentations. (*Left*) The feature vectors in the attention output have larger norms than those in the residual stream, reducing the residual non-invariance to approximately the noise level. (*Middle and Right*) Attention maps are mostly invariant under scalings. Rows correspond to different samples.

concentrates on a single point. This immediately rules out the dependence on local neighborhoods needed to implement a convolution, as well as the uniform attention for the mean computation. Instead, the pattern matches that of choosing a landmark. We next proceed to investigate how the self-attention output interacts with the residual stream. We compute the cosine similarity between $\boldsymbol{X}$ and $\mathrm{SA}(\boldsymbol{X})$ and show its distribution over the entire test set in Fig. 6 (*left*). The cosine similarities are consistently negative, indicating that the self-attention output tends to oppose the residual stream. This behavior is consistent with the subtraction required by both the landmark- and mean-based mechanisms. In fact, both mechanisms are special cases of the following observation.

Lemma 2. *Let* $\Lambda : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times n}$ *be translation invariant with row-stochastic outputs. Set* $\boldsymbol{\Lambda} = \Lambda(\boldsymbol{X})$. *Then* $f(\boldsymbol{X}) = g(\boldsymbol{X} - \boldsymbol{\Lambda}\boldsymbol{X})$ *is translation invariant for any function* g.

So even when attention does not focus on a landmark, its invariance along with opposite directions of $\boldsymbol{X}$ and $\mathrm{SA}(\boldsymbol{X})$ can ensure stable output under translation.

One may note that the above formulation already requires attention maps to be invariant in the first place. However, it does not require them to be expressive. Meanwhile, the residual connection enables the landmark shift mechanism $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \tilde{f}(\boldsymbol{x}_2 - \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n - \boldsymbol{x}_1)$, which is fully expressive among translation invariant functions [52]. Hence, we probe with ablations whether the residual connections are needed to ensure both high test accuracy and low invariance loss. Fig. 5 shows that, as Lemma 2 suggests, self-attention with residual connections achieves both of these properties simultaneously, while removing them breaks expressive invariance.

5.3 Scale: LayerNorm and residual suppression

The LayerNorm LN is a natural candidate for learning scale invariance. Without additive transformations like residual connections and bias terms, this extends to the entire computation for classification tasks.

Lemma 3. *Let* $\mathrm{SA}(\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})\boldsymbol{X}\boldsymbol{W}_V$ *be a single attention-only layer. If for some* $c > 0$, $\mathrm{A}(c\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})$, *then* $\mathrm{SA}(c\boldsymbol{X}) = c\,\mathrm{SA}(\boldsymbol{X})$. *Hence, an attention-only stack whose attention maps are stable over the scale augmentation range is positively homogeneous* $\Phi(c\boldsymbol{X}) = c\,\Phi(\boldsymbol{X})$. *If the final prediction is made by taking an* argmax *over logits that are scaled by this same positive factor, then the predicted class is invariant to* c.

Fig. 8 shows that a model with two layers of $\mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ easily becomes scale invariant. In this setting, the bias in the input projection is the only source of invariance error. In a full Transformer, the residual stream introduces an additional source of non-invariance. Fig. 7 shows how the model compensates for this effect: the feature vectors in the attention output have larger norms than the corresponding vectors in the residual stream, which reduces the influence of the residual on the block output.

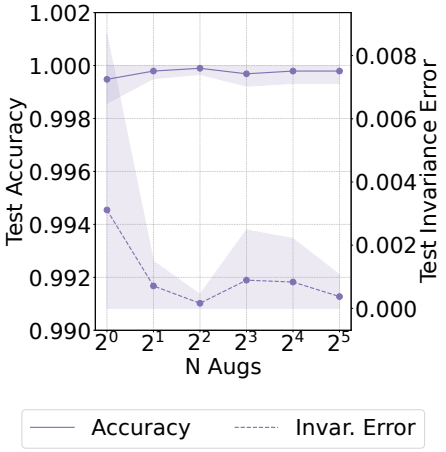


Figure 8: A model consisting only of $\mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ blocks can learn to be scale invariant and solve the 3D Topology task.

6 From invariance to equivariance

Our previous analysis focused on how Transformers can learn approximately invariant predictions from limited data augmentation. However, many real-world tasks require the more general notion of *equivariance*: when the input transforms under a group action, the output should transform in a corresponding way. Formally, for all $g \in G, x \in \mathcal{X}$, we have $\Psi(\rho_{\mathcal{X}}(g)x) = \rho_{\mathcal{Y}}(g)\Psi(x)$, where $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ are linear representations of the group on the input and output spaces, respectively. As a result, we also investigate whether attention-based structures related to the previously identified invariance mechanisms can also emerge in equivariant vector-valued tasks.

Experimental setup. To this end, we conduct additional experiments using ShapeNet [54], another common point cloud dataset that is annotated by the surface-normal vector for each point of the input, as well as segmentation information. Predicting surface normals poses an $\mathrm{O}(3)$- and $\mathrm{SO}(3)$-equivariant task. To construct $\mathrm{T}(3)$ and $\mathrm{Scale}(3)$-equivariant targets, we let each point predict the centroid of the part it belongs to in the segmentation map. Vanilla Transformer models are trained using data augmentation from each respective base angle-preserving group under the same OOD setting as in §4. In Fig. 9, we report task loss and equivariance error (see §B.2 for their definitions). The models achieve low task loss and mostly low equivariance error both in and out of distribution, suggesting that they learn approximately equivariant representations.

Potential mechanism for $\mathrm{O}(3)$-equivariance. We next investigate $\mathrm{O}(3)$-equivariance as a representative case. To identify a possible mechanism, we build on Proposition 4 of Villar et al. (2021) [52], which establishes the following:

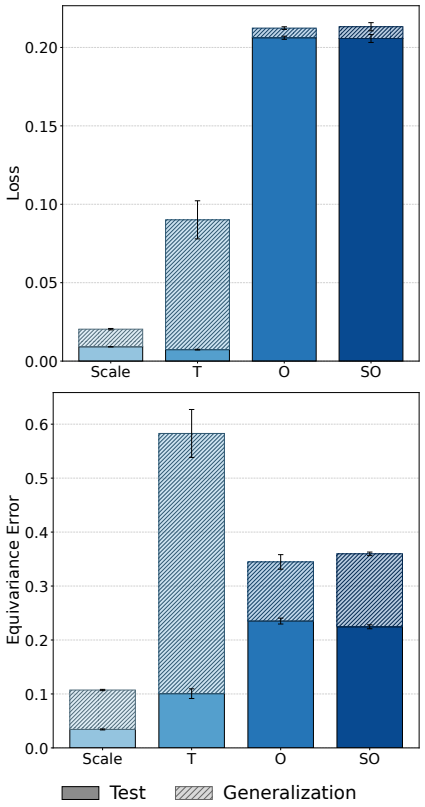


Figure 9: Task losses and equivariance errors of Transformers, trained with 16 augmentations on ShapeNet tasks, are reasonably low. Note that Scale and T consider a different task than O and SO.

Lemma 4 ([52]). *If h is an $\mathrm{O}(d)$-equivariant vector function of n vector inputs $\boldsymbol{v}_1, \boldsymbol{v}_2, .., \boldsymbol{v}_n$, then n $\mathrm{O}(d)$-invariant scalar functions $f_t(\cdot)$ exist such that $h(\boldsymbol{v}_1, \boldsymbol{v}_2, .., \boldsymbol{v}_n) = \sum_{t=1}^{1} f_t(\boldsymbol{v}_1, \boldsymbol{v}_2, .., \boldsymbol{v}_n)\boldsymbol{v}_t$.*

Self-attention has an analogous weighted-sum structure: $\mathrm{SA}(\boldsymbol{X}) = \sum_j \mathrm{A}(\boldsymbol{X})_{ij}(\boldsymbol{X}\boldsymbol{W}_V)_j$. Therefore, we can see that if *(i)* $\mathrm{A}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X})$ and *(ii)* $\boldsymbol{X}\boldsymbol{R}\boldsymbol{W}_V = \boldsymbol{X}\boldsymbol{W}_V\boldsymbol{R}$, then we have $\mathrm{SA}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X}\boldsymbol{R})\boldsymbol{X}\boldsymbol{R}\boldsymbol{W}_V = \mathrm{A}(\boldsymbol{X}\boldsymbol{R})\boldsymbol{X}\boldsymbol{W}_V\boldsymbol{R} = \mathrm{SA}(\boldsymbol{X})\boldsymbol{R}$. That is, self-attention is equivariant whenever its attention map is invariant and its value projection transforms equivariantly. Requiring *(ii)* to hold for every $\boldsymbol{X}$ and every $\boldsymbol{R} \in \mathrm{O}(d)$ is equivalent to enforcing $\boldsymbol{W}_V$ to commute with the standard action of $\mathrm{O}(d)$. This would restrict to an isotropic map, namely a scalar multiple of the identity.

We first test condition *(i)*. Fig. 10 (*right*) reports the invariance error of the attention map A for a trained Transformer. The low error suggests that the attention map is approximately invariant and therefore approximately satisfies condition *(i)*. In contrast, the learned value projection is *not* globally isotropic. The strongest form of condition *(ii)*, i.e., exact commutation with every $\boldsymbol{R} \in \mathrm{O}(d)$ over the full ambient input space, is *not* reflected in the learned parameters. Instead, we hypothesize that the required equivariance may hold only approximately on the data support and along the augmentation orbits. We test this hypothesis directly on the test set. Measuring the equivariance of $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$ requires specifying how $\mathrm{O}(3)$ acts in the higher-dimensional latent space. We instead measure the invariance of the Gram matrix $\boldsymbol{V}\boldsymbol{V}^\top$ as a proxy. If $\boldsymbol{V}$ transforms equivariantly under an orthogonal action, then $\boldsymbol{V}\boldsymbol{V}^\top$ remains invariant. Fig. 10 (*left*) shows that the corresponding invariance error is small, which supports approximate equivariance of $\boldsymbol{V}$ on the data support and along the augmentation orbits. We provide further investigation of the remaining architectural components and symmetry groups in §C.5. Overall, the Transformers trained for equivariant prediction exhibit mechanisms similar to those identified in our invariance analysis. This suggests that invariant components, e.g., those studied in §5, can be used as building blocks for equivariance.

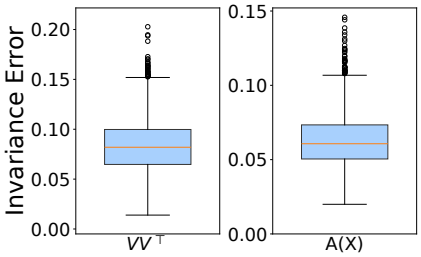


Figure 10: Results for the first layer of a Transformer trained on ShapeNet with 16 $\mathrm{O}(3)$ augmentations per sample. On the test set, the value representations are approximately $\mathrm{O}(3)$-equivariant (*left*) and the attention maps are approximately $\mathrm{O}(3)$-invariant (*right*).

7 Conclusions

In this paper, we study how a vanilla Transformer can learn symmetries from limited data augmentation on point cloud data. Across the transformation groups we consider, Transformers learn angle-preserving symmetries more reliably than non-angle-preserving ones. Among the angle-preserving groups, they learn the base symmetries (i.e., translation, rotation, and scale) more efficiently than compositions of these transformations. Following those observations, we perform structural analyses of the trained Transformer models and identify mechanisms that implement invariance to each base angle-preserving symmetry. We also extend this analysis from invariant to equivariant prediction, suggesting that the invariant mechanism identified in our analysis can also serve as building blocks for equivariant computation. Our observations suggest that data augmentation should not be treated as a uniform substitute for architectural equivariance, and highlight the persisting need to understand when *soft* equivariance is sufficient and when explicit equivariant structure remains necessary. In addition, they also suggest that the recent success of Transformer-based soft-equivariant approaches could be partly attributed to the intrinsic alignment of that architecture with base angle-preserving symmetries, which happen to be among the most prevalent ones in scientific domains.

Limitation discussion. This work focuses on small vanilla Transformers and point cloud dataset to isolate the interplay of the basic architecture with the considered symmetries. An important direction for future work is to test whether similar mechanisms arise in images, meshes, graphs, molecular structures, and other forms of geometric or scientific data. In addition, studying larger datasets and more elaborate scientific tasks would help determine whether the same mechanisms persist when the data contain richer structure, noise, and interacting symmetries. Lastly, the extension of our analysis to equivariance remains preliminary. Future work could identify mechanisms for additional equivariant tasks and symmetry groups, characterize how group actions are represented in latent spaces, and test the causal role of the observed mechanisms through targeted interventions.

Acknowledgments

We thank Leibniz Supercomputing Centre (LRZ) for providing computational resources and the anonymous reviewers for their constructive comments. We also thank Vincent Bürgin for feedback on an earlier version of this manuscript. ESE acknowledges support from a PhD fellowship of the Munich Center for Machine Learning (MCML). YEL acknowledges support from the Schmidt Futures Israeli Women's Postdoctoral Award and the Viterbi Fellowship from the Faculty of Electrical and Computer Engineering at the Technion. This research was supported by an Alexander von Humboldt Professorship.

References

[1] Jason Gibson, Ajinkya Hire, and Richard G Hennig. Data-augmentation for graph neural network learning of the relaxed energies of unrelaxed structures. *npj Computational Materials*, 8(1):211, 2022.

[2] Ahmed A Elhag, Arun Raja, Alex Morehead, Samuel M Blau, Garrett M Morris, and Michael M Bronstein. Learning Inter-Atomic Potentials without Explicit Equivariance. *arXiv preprint arXiv:2510.00027*, 2025.

[3] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[4] Juntao Deng, Miao Gu, Pengyan Zhang, Tao Liu, Guansong Hu, Mingyu Dong, Yabin Zhang, Yizhen Song, Yunfan Zhang, Min Liu, et al. Mutppi+: a multimodal framework for predicting mutation effects on protein–protein interactions via mutation-path-based data augmentation. *Briefings in Bioinformatics*, 27(2):bbag105, 2026.

[5] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[6] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33:19884–19895, 2020.

[7] Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning. *International Journal of Computer Vision*, 133(10):7368–7405, 2025.

[8] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[9] Yuyang Wang, Ahmed AA Elhag, Navdeep Jaitly, Joshua M Susskind, and Miguel Ángel Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. In *International Conference on Machine Learning*, pages 50400–50418. PMLR, 2024.

[10] Alex Morehead, Miruna Cretu, Antonia Panescu, Rishabh Anand, Maurice Weiler, Tynan Perez, Samuel Blau, Steven Farrell, Wahid Bhimji, Anubhav Jain, et al. Zatom-1: A Multimodal Flow Foundation Model for 3D Molecules and Materials. *arXiv preprint arXiv:2602.22251*, 2026.

[11] Carlos Vonessen, Charles Harris, Miruna Cretu, and Pietro Lio. TABASCO: A fast, simplified model for molecular generation with improved physical quality. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025.

[12] Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

[13] Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*, 2020.

[14] Behrooz Tahmasebi and Melanie Weber. Achieving Approximate Symmetry Is Exponentially Easier than Exact Symmetry. In *The Fourteenth International Conference on Learning Representations*, 2026.

[15] Behrooz Tahmasebi, Melanie Weber, and Stefanie Jegelka. Data Augmentation: A Fourier Analysis Perspective. In *NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations*, 2025.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[17] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[18] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[20] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[21] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[22] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[23] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International conference on machine learning*, pages 3318–3328. PMLR, 2021.

[24] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations. *arXiv preprint arXiv:2103.01436*, 2021.

[25] Jan Gerken, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Equivariance versus augmentation for spherical images. In *International Conference on Machine Learning*, pages 7404–7421. PMLR, 2022.

[26] Rui Wang, Robin Walters, and Rose Yu. Data augmentation vs. equivariant networks: A theory of generalization on dynamics forecasting. *arXiv preprint arXiv:2206.09450*, 2022.

[27] Facundo Quiroga, Franco Ronchetti, Laura Lanzarini, and Aurelio F Bariviera. Revisiting data augmentation for rotational invariance in convolutional neural networks. In *Modelling and Simulation in Management Sciences: Proceedings of the International Conference on Modelling and Simulation in Management Sciences (MS-18)*, pages 127–141. Springer, 2020.

[28] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347, 2023.

[29] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame Averaging for Invariant and Equivariant Network Design. In *International Conference on Learning Representations*, 2022.

[30] Alexandre Agm Duval, Victor Schmidt, Alex Hernández-Garcıa, Santiago Miret, Fragkiskos D Malliaros, Yoshua Bengio, and David Rolnick. Faenet: Frame averaging equivariant GNN for materials modeling. In *International Conference on Machine Learning*, pages 9013–9033. PMLR, 2023.

[31] Yuchao Lin, Jacob Helwig, Shurui Gui, and Shuiwang Ji. Equivariance via Minimal Frame Averaging for More Symmetries and Efficiency. In *International Conference on Machine Learning*, pages 30042–30079. PMLR, 2024.

[32] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.

[33] George Ma, Yifei Wang, Derek Lim, Stefanie Jegelka, and Yisen Wang. A canonicalization perspective on invariant and equivariant learning. *Advances in Neural Information Processing Systems*, 37:60936–60979, 2024.

[34] Ya-Wei Eileen Lin, Ronen Talmon, and Ron Levie. Equivariant machine learning on graphs with nonlinear spectral filters. *Advances in Neural Information Processing Systems*, 37:128182–128226, 2024.

[35] Ya-Wei Eileen Lin and Ron Levie. Adaptive canonicalization with application to invariant anisotropic geometric networks. In *The Fourteenth International Conference on Learning Representations*, 2026.

[36] Ahmed A. A. Elhag, T. Konstantin Rusch, Francesco Di Giovanni, and Michael M. Bronstein. Relaxed Equivariance via Multitask Learning. In *The Fourth Learning on Graphs Conference*, 2025.

[37] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Yi-Lun Liao, Brandon M Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In *The Twelfth International Conference on Learning Representations*, 2023.

[39] Yi-Lun Liao, Alexander J Hoffman, Sabrina C Shen, Alexandre Duval, Sam Walton Norwood, and Tess Smidt. EquiformerV3: Scaling Efficient, Expressive, and General SE(3)-Equivariant Graph Attention Transformers. *arXiv preprint arXiv:2604.09130*, 2026.

[40] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pages 4533–4543. PMLR, 2021.

[41] Johann Brehmer, Pim De Haan, Sönke Behrends, and Taco S Cohen. Geometric algebra transformer. *Advances in Neural Information Processing Systems*, 36:35472–35496, 2023.

[42] Jonas Spinner, Victor Bresó, Pim De Haan, Tilman Plehn, Jesse Thaler, and Johann Brehmer. Lorentz-equivariant geometric algebra transformers for high-energy physics. *Advances in Neural Information Processing Systems*, 37:22178–22205, 2024.

[43] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *Advances in Neural Information Processing Systems*, 34:27331–27343, 2021.

[44] Mohammad Mohaiminul Islam, Rishabh Anand, David R Wessels, Friso de Kruiff, Thijs P Kuipers, Rex Ying, Clara I Sánchez, Sharvaree Vadgama, Georg Bökman, and Erik J Bekkers. Platonic transformers: A solid choice for equivariance. *arXiv preprint arXiv:2510.03511*, 2025.

[45] Nima Dehmamy, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu. Automatic Symmetry Discovery with Lie Algebra Convolutional Network. In *Advances in Neural Information Processing Systems*, volume 34, pages 2503–2515. Curran Associates, Inc., 2021.

[46] Srinath Bulusu, Matteo Favoni, Andreas Ipp, David I Müller, and Daniel Schuh. Generalization capabilities of translationally equivariant neural networks. *Physical Review D*, 104(7):074504, 2021.

[47] Johann Brehmer, Sönke Behrends, Pim De Haan, and Taco Cohen. Does equivariance matter at scale? *Transactions on Machine Learning Research*, 2025.

[48] Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

[49] Eduardo Santos-Escriche, Ya-Wei Eileen Lin, and Stefanie Jegelka. LieAugmenter: Equivariant Learning by Discovering Symmetries with Learnable Augmentations. *arXiv preprint arXiv:2506.03914*, 2025.

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[52] Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in Neural Information Processing Systems*, 34:28848–28863, 2021.

[53] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

[54] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[55] Andrei Manolache, Luiz Chamon, and Mathias Niepert. Learning (approximately) equivariant networks via constrained optimization. *Advances in Neural Information Processing Systems*, 38:144352–144381, 2026.

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

[57] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *Ai Open*, 3:71–90, 2022.

[58] Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803, 2023.

[59] Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871*, 2022.

[60] Chi-Heng Lin, Chiraag Kaushik, Eva L Dyer, and Vidya Muthukumar. The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective. *Journal of Machine Learning Research*, 25(91):1–85, 2024.

[61] Ashkan Soleymani, Behrooz Tahmasebi, Stefanie Jegelka, and Patrick Jaillet. A robust kernel statistical test of invariance: Detecting subtle asymmetries. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025.

[62] Ruoqi Shen, Sébastien Bubeck, and Suriya Gunasekar. Data augmentation as feature manipulation. In *International Conference on Machine Learning*, pages 19773–19808. PMLR, 2022.

[63] Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR, 2019.

[64] Pratik Patil and Jin-Hong Du. Generalized equivalences between subsampling and ridge regularization. *Advances in Neural Information Processing Systems*, 36:78926–78963, 2023.

[65] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pages 3351–3418. PMLR, 2021.

[66] Stefanos Pertigkiozoglou, Evangelos Chatzipantazis, Shubhendu Trivedi, and Kostas Daniilidis. Improving equivariant model training via constraint relaxation. *Advances in Neural Information Processing Systems*, 37:83497–83520, 2024.

[67] Ningyuan Huang, Ron Levie, and Soledad Villar. Approximately equivariant graph networks. *Advances in Neural Information Processing Systems*, 36:34627–34660, 2023.

[68] Ahmed A Elhag, T Konstantin Rusch, Francesco Di Giovanni, and Michael Bronstein. Relaxed Equivariance via Multitask Learning. *arXiv preprint arXiv:2410.17878*, 2024.

[69] Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems*, 34:30037–30049, 2021.

[70] Javier Mariño Villadamigo, Rikkert Frederix, Tilman Plehn, Timea Vitos, and Ramon Winterhalder. FASTColor–Full-color Amplitude Surrogate Toolkit for QCD. *arXiv preprint arXiv:2509.07068*, 2025.

[71] Tianxiao He, Alex H Williams, and Sarah E Harvey. How data augmentation shapes neural representations. *arXiv preprint arXiv:2605.15306*, 2026.

[72] Behnam Neyshabur. Towards learning convolutions from scratch. *Advances in Neural Information Processing Systems*, 33:8078–8088, 2020.

[73] Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The Lie Derivative for Measuring Learned Equivariance. In *The Eleventh International Conference on Learning Representations*, 2023.

[74] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

[75] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[76] Diane Bouchacourt, Mark Ibrahim, and Ari Morcos. Grounding inductive biases in natural images: invariance stems from variations in data. *Advances in Neural Information Processing Systems*, 34:19566–19579, 2021.

[77] Andrew Baker. *Matrix groups: An introduction to Lie group theory*. Springer Science & Business Media, 2003.

[78] Karin Erdmann and Mark J Wildon. *Introduction to Lie algebras*, volume 122. Springer, 2006.

[79] Alexander A Kirillov. *An introduction to Lie groups and Lie algebras*, volume 113. Cambridge University Press, 2008.

[80] Brian C Hall. Lie groups, Lie algebras, and representations. In *Quantum Theory for Mathematicians*, pages 333–366. Springer, 2013.

[81] Robert Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, 2003.

[82] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[83] Luca Falorsi, Pim de Haan, Tim R Davidson, and Patrick Forré. Reparameterizing distributions on Lie groups. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3244–3253. PMLR, 2019.

[84] Artem Moskalev, Anna Sepliarskaia, Erik J Bekkers, and Arnold WM Smeulders. On genuine invariance learning without weight-tying. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 218–227. PMLR, 2023.

[85] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[86] Pavlo Melnyk, Michael Felsberg, and Mårten Wadenbäck. Embed me if you can: A geometric perceptron. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1276–1284, 2021.

[87] Pavlo Melnyk, Michael Felsberg, and Mårten Wadenbäck. Steerable 3d spherical neurons. In *International Conference on Machine Learning*, pages 15330–15339. PMLR, 2022.

[88] Robin Winter, Marco Bertolini, Tuan Le, Frank Noe, and Djork-Arné Clevert. Unsupervised learning of group invariant and equivariant representations. *Advances in Neural Information Processing Systems*, 35:31942–31956, 2022.

[89] Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *Journal of the American Chemical Society*, 131(25):8732–8733, 2009.

[90] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[91] Viacheslav Khomenko, Oleg Shyshkov, Olga Radyvonenko, and Kostiantyn Bokhan. Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization. In *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, pages 100–103. IEEE, 2016.

[92] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[93] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, 2018.

[94] David W. Romero and Jean-Baptiste Cordonnier. Group Equivariant Stand-Alone Self-Attention For Vision. In *International Conference on Learning Representations*, 2021.

[95] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

[96] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pages 9992–10002. Ieee, 2021.

[97] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Supplementary Material

A Extended related work **18**

A.1 The shift from hard equivariance towards data augmentation 18

A.2 Approximate equivariance 18

A.3 Learning symmetries with neural networks 18

B Experimental details **19**

B.1 Generating symmetry augmentations 19

B.2 Measuring learned invariance and equivariance 21

B.3 Finite augmentation reveals a symmetry inductive bias in Transformers 22

B.4 Learning base angle-preserving symmetries 23

B.5 From invariance to equivariance 23

C Additional experimental results **25**

C.1 3D Tetris 25

C.2 Molecular property prediction 26

C.3 Additional architectural components 27

C.4 Signature of detected mechanisms for additional models 33

C.5 Further results on equivariance 40

D Proofs **41**

D.1 Proof of Lemma 1 41

D.2 Proof of Lemma 2 41

D.3 Proof of Lemma 3 42

A Extended related work

In this section, we expand on the surrounding literature presented in §1 and §2 by discussing additional related works.

A.1 The shift from hard equivariance towards data augmentation

Recently, a trend has emerged that is moving away from the design and implementation of hard equivariant architectures especially when working with generative tasks in large-scale data regimes. The main motivation behind this development has been the empirical observation that data augmentation is able to provide a similar performance with a much simpler implementation and a reduced computational cost in such scenarios. Most notable is the example of the departure of AlphaFold3 [3] from the constrained architectural equivariance of AlphaFold2 [8], but many other works have recently followed the same path, e.g., MCFs [9], TransIP [2], Zatom-1 [10], or TABASCO [11]. All of these works have two main aspects in common: (1) they propose Transformer-based models, (2) they work with real-world scientific tasks (e.g., molecule and material generation, interatomic potential prediction) which are naturally restricted to a reduced set of possible symmetries, including the angle-preserving ones that we show the Transformer efficiently learns to be invariant to.

A.2 Approximate equivariance

A limitation of equivariant architectures arises in their application to real-world data, which often deviates from strict mathematical symmetries due to, for instance, noisy or biased measurements and symmetry-breaking effects [55, 56]. In such cases, strictly enforcing equivariance can lead to a degradation of the training dynamics and predictive performance of the models. As a result, approximately equivariant networks emerge as a more fitting alternative for such scenarios, where they are still biased towards preserving the symmetry but are not strictly restricted to do so.

Data augmentation has become the most predominant technique to encourage approximate equivariance in many different practical domains: natural language processing [57, 58], reinforcement learning [7], computer vision [5], or graph learning [59]. A theoretical understanding of the role of data augmentation has been explored from multiple directions: its group-theoretic foundations [12], its interpretation as a regularizer [60, 61], or the effect it has on the training dynamics of neural networks [62]. There have also been multiple works that develop and investigate a kernel-based theory of augmentation [63–65].

Other examples of approximate equivariance methods include constrained optimization approaches [55, 66], the relaxed weight-sharing and weight-tying strategy from [56], the graph coarsening method for GNNs described in [67], the multi-task learning approach REMUL [68], or the soft prior approach from [69].

A.3 Learning symmetries with neural networks

Related to our work, there have been some papers in the literature that explore the ability of different machine learning models to learn specific symmetries. For instance, FASTColor [70] empirically shows that a Transformer learns to become invariant to the $\mathrm{SO}(2)$ symmetries in a particle physics task while avoiding invariance to other symmetries like $\mathrm{SL}(4)$, which aligns with our findings in this paper. The effect of data augmentation on the internal representations of neural networks has been explored in [71]. Also related is [72], which shows that fully-connected networks can learn to implement convolution structures when trained with an appropriate regularizer. Different methods and metrics have been proposed to measure the learned equivariance in neural networks [73–76]. However, each of them comes with certain drawbacks, such as requiring various ad-hoc decisions or being designed with only certain transformations and types of data in mind. Most related to our work is [73], which shows that Transformers can learn to be more equivariant than convolutional neural networks after training with data augmentation.

B Experimental details

In this section, we present additional details about the experimental setup and empirical evaluations presented in §3 and §4. All experiments were implemented in Pytorch and executed on NVIDIA A100 and H100 GPUs using CUDA version 12.2. We used the Adam optimizer with a learning rate of 0.001, keeping the rest of its default hyperparameter values. The embedding dimension as well as the hidden dimension of the MLP in all experiments is 64. All models contain two blocks using single-head self-attention and a 2-layer ReLU-MLP.

B.1 Generating symmetry augmentations

We characterize the symmetry groups considered in this work in Table 1. There are multiple ways of generating augmentations based on the action of these groups. In this paper, we follow an approach based on Lie group theory, which we describe in more detail below.

B.1.1 Preliminaries

First, we briefly review some relevant background regarding Matrix Lie groups and Lie algebras. A matrix Lie group G is a subgroup of the general linear group $\mathrm{GL}(d, \mathbb{R})$ that is also a smooth manifold such that multiplication and inversion are both smooth maps [77, 78]. The associated Lie algebra $\mathfrak{g} = T_{\mathrm{Id}}G$ is the tangent space of the group G at the identity element. Each element $A \in \mathfrak{g}$ is an infinitesimal generator of one-parameter subgroups $t \mapsto \exp(tA)$ for $t \in \mathbb{R}$. Let $\{L_i\}_{i=1}^{\dim \mathfrak{g}}$ be a basis of $\mathfrak{g}$. For matrix Lie groups, the exponential map $\exp : \mathfrak{g} \to G$ is given by the matrix exponential, which is a local diffeomorphism in a neighborhood of the identity. If G is connected and compact, $\exp$ is surjective [79, 80], hence every $g \in G$ can be written as $g = \exp(\sum_i w_i L_i)$ with $w_i \in \mathbb{R}$.

Second, we recall that we are working with point cloud data, i.e., $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^{n \times d}$. As a result, we want to generate augmentations that act on the entire point cloud. For computational simplicity and efficiency, we achieve this by sampling point-wise actions and mapping them across the entire corresponding point cloud. Thus, in the following we focus on explaining the generation of such point-wise augmentations.

B.1.2 Parameterizing augmentations using Lie groups

In order to represent the generators of the Lie Group that we want to sample from in a compact, consistent, and computationally efficient way, we adopt a reparameterization approach [81, 82] adapted to Lie groups [83], which samples group elements through the associated Lie algebra.

Let $\mathfrak{g}$ denote the Lie algebra associated with a connected matrix Lie group acting on the representation space of $\mathcal{X}$. We parameterize the sampling of group elements through a standard Lie algebra and randomly sample coefficients $w = (w_1, \ldots, w_C) \in \mathbb{R}^C$ from a distribution $P(\gamma)$ with parameters γ_0, γ_1. Specifically, we consider a uniform distribution $P(\gamma) = \mathcal{U}[-\gamma_0, \gamma_1]$. A group element is obtained by mapping the resulting Lie algebra element to the group via the matrix exponential, given by

$$g = \exp\left[\sum\nolimits_{i=1}^{C} w_i L_i\right], \; w_i \sim P(\gamma). \tag{6}$$

Given an input $\boldsymbol{x}_i$, we draw K independent group elements $g_{i,2}, \ldots, g_{i,K+1}$ according to Eq. 6 and form the corresponding augmentations

$$\boldsymbol{x}_{i,j} := g_{i,j}\, \boldsymbol{x}_i, \qquad j \in [2, \ldots, K+1], \tag{7}$$

by multiplying with the original input $\boldsymbol{x}_{i,1} := \boldsymbol{x}_i$.

B.1.3 Examples

In order to illustrate how such a representation of the group looks like for the symmetries that we consider, we present a standard basis for rotation and translation groups in Table 2.

Note that when working with affine transformations, such as translations, we first embed the point representations into homogeneous coordinates. After computing the augmentations, we project the points back to the original space.

Symmetry group	Lie algebra basis
$\mathrm{SO}(3)$	$\left\{ \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \right\}$
$\mathrm{T}(3)$	$\left\{ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right\}$

Table 2: Example of Lie algebra bases for the rotation $\mathrm{SO}(3)$ and translation groups $\mathrm{T}(3)$.

B.1.4 Additional remarks

We should note that this symmetry augmentation implementation cannot guarantee full coverage of all the considered groups (e.g., $\mathrm{SL}(n), \mathrm{GL}(n)$), since it is restricted to sampling group elements from the connected component of the identity of the Lie group and the map is not always surjective for non-compact groups. However, it is still adequate for the purpose of our analysis due to a number of reasons. First, it allows for a general, uniform and consistent implementation of the augmentation across all symmetry groups, which also improves the ease of their comparability. In addition, this implementation is more computationally efficient than dedicated alternatives, which allows for a more extensive and comprehensive analysis. Moreover, given that the goal of our work is to explore the ability of Transformers to learn different symmetries, focusing on local continuous transformations around the identity is a natural choice. It allows a grounded analysis that corresponds to a realistic setting in which the transformations create meaningful invariance targets while avoiding extreme transformations that could destroy the underlying semantic information of the inputs. Lastly, the limitations of this implementation do not affect the main groups that we focus on in the paper, i.e., angle-preserving symmetries like $\mathrm{SO}(n)$, $\mathrm{T}(n)$, or $\mathrm{Scale}(n)$, all of which are fully covered by this parameterization, since the exponential maps are surjective onto their connected components. In the case of $\mathrm{O}(n)$, we circumvent the restriction to the identity connected component by multiplying elements sampled from $\mathrm{SO}(n)$ by a randomly sampled reflection mask.

B.1.5 Hardness of compositional groups

The mechanistic analysis presented in §5 focuses on the base groups $\mathrm{SO}(3)/\,\mathrm{O}(3)$, $\mathrm{T}(3)$, and $\mathrm{Scale}(3)$. We selected these groups for our analysis because they showed the strongest finite-augmentation performance and clearest OOD generalization, making them the most suitable for searching for stable internal mechanisms.

When considering groups that compose them, there is an important distinction between representational compatibility and learnability from finite joint augmentations. At the level of exact functions, invariance to the generating base groups is sufficient for invariance to their composition. For example, if f is exactly invariant to both translations and rotations, then for any $g = tR \in SE(3)$, $f(gx) = f(t(Rx)) = f(Rx) = f(x)$. The same argument extends to $\mathrm{Sim}(3)$ when scale invariance is also present. The observed difficulty is more plausibly associated with finite sampling and optimization. We currently see three related explanations.

1. *Higher-dimensional augmentation coverage*

   In three dimensions, translations and rotations each have three continuous degrees of freedom and uniform scaling has one, whereas $\mathrm{SE}(3)$ has six and $\mathrm{Sim}(3)$ has seven. For a fixed number of augmented views per sample, random samples provide substantially sparser coverage of the larger joint transformation space.

   Moreover, when augmentations are sampled directly from a compositional group, almost every transformed example combines multiple factors. The model receives relatively little direct supervision corresponding to a pure translation, pure rotation, or pure scaling direction. By contrast, a model trained on a base group receives its entire augmentation budget along that factor.

2. *Simultaneous constraints on shared components*

   The mechanisms identified for the base groups use overlapping architectural components but impose different functional requirements:

   - rotation and reflection favor attention-score differences that are stable under orthogonal transformations
   - translation favors an invariant attention rule followed by subtraction of a selected or weighted reference through the residual stream
   - scale favors stable attention, positive homogeneity, LayerNorm, and suppression of non-homogeneous residual or bias contributions.

   Learning $\mathrm{SE}(3)$ or $\mathrm{Sim}(3)$ requires these conditions to hold simultaneously along joint transformation orbits. Although the mechanisms are compatible in principle, their simultaneous optimization within the same attention and residual pathways may be more difficult than learning each one separately.

3. *Approximation error under composition*

   Exact invariance to each subgroup composes exactly, but finite-data training produces only approximate invariance. Errors associated with the constituent transformations can therefore compound when several transformations are applied jointly. A model that is moderately stable to rotation and moderately stable to translation need not be equally stable to arbitrary roto-translations unless both behaviors are learned with sufficient uniformity.

Our new multi-head experiments in §C.3 provide preliminary evidence relevant to the optimization hypothesis. Increasing the number of heads does not materially improve performance for the compositional groups, and we do not observe a clear specialization in which distinct heads separately implement translation, rotation, and scale invariance. This suggests that simply providing additional parallel attention heads is not sufficient to resolve the joint learning problem.

B.2 Measuring learned invariance and equivariance

We assess equivariance by measuring the consistency of model predictions under transformations sampled from the ground-truth symmetry group. Specifically, we compute

$$\frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\ell\left(\rho_{\mathcal{Y}}(g_j)(\Psi(x_i)),\Psi(\rho_{\mathcal{X}}(g_j)(x_i))\right), \tag{8}$$

which contrasts the transformed predictions of the original samples and those of the augmented samples, averaged over K sampled group elements and N data points. In the case of regression tasks and for attention maps, ℓ is the L1-norm of the difference, while for classification tasks it corresponds to the KL divergence applied to the probability distributions outputted by the model. The representations $\rho_{\mathcal{X}} : G \to \mathrm{GL}(n)$ and $\rho_{\mathcal{Y}} : G \to \mathrm{GL}(m)$ specify linear actions of the symmetry group G on $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$, respectively. When measuring invariance, as is the case in the majority of our experiments, we set $\rho_{\mathcal{Y}} = Id$.

In the classification setting, the invariance condition of direct task relevance is equality of predictive distributions. We used KL divergence since it operates directly on the probability simplex, is zero exactly when the two predictive distributions coincide, and weights discrepancies according to the probability mass assigned by the reference prediction. It also ignores a common additive shift of all logits, which leaves the predictive distribution unchanged. By contrast, a raw-logit L2 distance is a stricter, representation-dependent criterion: it can be nonzero even when the two softmax distributions are identical, and it measures changes in logit parameterization in addition to changes in the model's prediction. This makes KL a natural primary measure of task-level predictive invariance.

Nevertheless, for completeness, we study three complementary metrics: symmetric KL divergence, Jensen-Shannon divergence, and the logit-invariance metric of [84]. The last metric is an L2-type alternative based on the squared distance between the logits.

Across the base symmetries on Topology 3D, these metrics give consistent qualitative conclusions as can be observed in Figure 11. Their correlations with the original KL measure range from moderate to near-perfect: agreement is strongest for Scale, and weakest for Translation. The weaker agreement for Translation is consistent with it being the most difficult OOD case in our experiments. Thus, the

main ordering of the symmetries and the mechanistic conclusions are not artifacts of the choice of invariance metric.

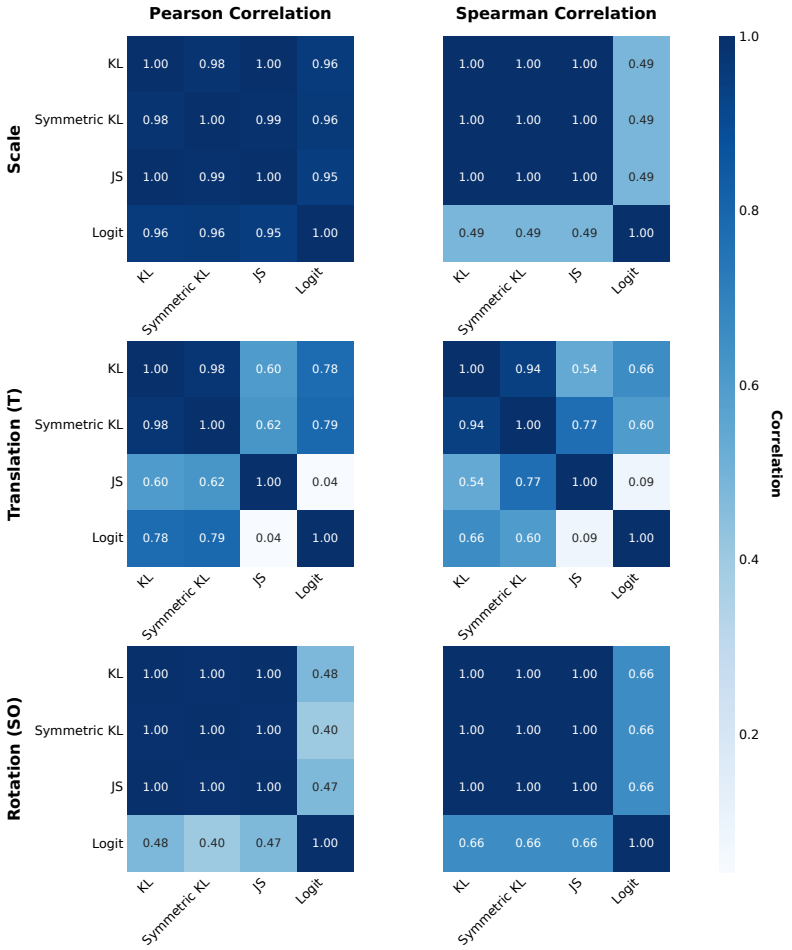


Figure 11: OOD Topology invariance metrics correlation for base angle-preserving symmetries.

B.3 Finite augmentation reveals a symmetry inductive bias in Transformers

In this section, we provide additional details regarding the datasets and experiments presented in §3 of the paper.

3D Topology. In order to demonstrate that our results are due to the alignment of the Transformer architecture with the symmetries instead of it being an artifact of the input-output mapping being lost through the non-rigid transformations, we define a synthetic benchmark. In particular, we construct a very simple dataset consisting of multiple random samplings of point clouds for five different shapes: a sphere, a torus, a double torus, two disjoint spheres, and two linked tori. We randomly generate 100 samples per class for train and 50 for validation and testing. The sphere, torus, and double torus represent increasing values of genus (or first Betti number, or number of holes), while the two disjoint spheres are differentiated through the zeroth Betti number (or number of connected components), and the linked tori distinguish themselves through the linking number (or ambient entanglement). All of those properties remain preserved through not only the angle-preserving transformations, but also the the homeomorphism transformations represented by the non-angle-preserving groups we consider (i.e., $\mathrm{SL}(3), \mathrm{GL}(3), \mathrm{Affine}(3)$).

ModelNet10. We also work with ModelNet10 [50], which following standard practice, we pre-process by centering and normalizing the point coordinates to the range $(-1, 1)$ and sampling 1024 points per cloud. We use the official train-test splits. Note that ModelNet10 objects are provided in an

Table 3: Parameter overview of compared architectures. Both operate on 3D point sets and are parameter-matched to $\approx$51k. The number of classes is denoted by c.

Model	Block	Layer sizes	#Params
DeepSets	Encoder ϕ	$3 \to 156 \to 156$, sum-pool	25,740
	Decoder ρ	$156 \to 156 \to c$ (dropout 0.1)	25,748
PointNet	Input T-Net (k=3)	$3 \to 9 \to 18 \to 140$, maxpool, $140 \to 70 \to 35 \to 9$	16,099
	MLP 1	$3 \to 9 \to 9$	162
	Feature T-Net (k=9)	$9 \to 9 \to 18 \to 140$, maxpool, $140 \to 70 \to 35 \to 81$	18,745
	MLP 2	$9 \to 18 \to 140$, maxpool	3,156
	Final MLP	$140 \to 70 \to 35 \to c$ (dropout 0.1)	12,853

approximately canonical frame. Moreover, all of our comparisons use the same underlying dataset and differ only in the augmentation group. Thus, any invariance induced by the dataset distribution itself is shared across settings, while differences between models reflect the additional symmetry constraints imposed by augmentation.

The effect of data augmentation. We illustrate the effect of the data augmentation process for the different symmetries we consider in our experiments in Figure 12 for some sample point clouds of the 3D Topology dataset.

B.4 Learning base angle-preserving symmetries

In this section, we provide additional details regarding the architectures and experiments presented in §4 of the paper.

DeepSets [18]. Permutation-invariant architecture designed to work with set inputs, defined as $f(X) = \rho(\sum_{\boldsymbol{x} \in \mathbb{X}} \phi(\boldsymbol{x}))$, where $\mathbb{X}$ is a set, and ρ, ϕ are universal approximators. In practice, we implement ϕ as a shared MLP composed of point-wise linear layers, batch normalization, and ReLU activations applied independently to each element of the input. The global aggregation is performed using sum pooling across the input dimension. Finally, ρ is defined as a 2-layer ReLU-MLP with dropout that processes the aggregated feature vector and produces the final predictions.

PointNet [51]. A permutation-invariant architecture for point cloud tasks, e.g., object classification or part segmentation, defined similarly to DeepSets, but relying on max pooling and adding alignment networks known as transformation networks (T-nets) to both raw input coordinates and the intermediate feature space. These T-nets are used to predict transformation matrices and apply said transformations to the coordinates of input points, seeking that the network is invariant to linear transformations, like rotation and and reflection. In practice, both the overarching PointNet architecture and the T-nets are defined using shared MLPs with point-wise linear layers, batch normalization, and ReLU activations applied independently to each element of the input. The aligned per-point features are aggregated via max pooling into a single global feature vector, which is then passed through a 3-layer MLP with dropout to produce the final outputs.

We try to roughly match the parameter count of the used Transformer, while approximately maintaining the relative module sizes, which results in the settings listed in Table 3. For the 10 classes of ModelNet10, the total counts come in at 50,954 (Transformer), 51,802 (DeepSets) and 51,087 (PointNet) trainable parameters.

OOD setting. Seeking to explore whether the learned invariances also hold for transformations not seen at training time, we define the training, validation, test, and generalization ranges of augmentations as shown in Table 4. During training, the model is exposed only to augmentations sampled from the smaller-magnitude range, and it is then tested on augmentations sampled from a disjoint larger-magnitude range.

B.5 From invariance to equivariance

ShapeNet. ShapeNet [54] consists of 16,881 coordinate point clouds representing 16 different shapes like e.g. *Airplane* or *Bag*. It is annotated by surface normal vectors for each point and segmentation maps with two to six segments per category.

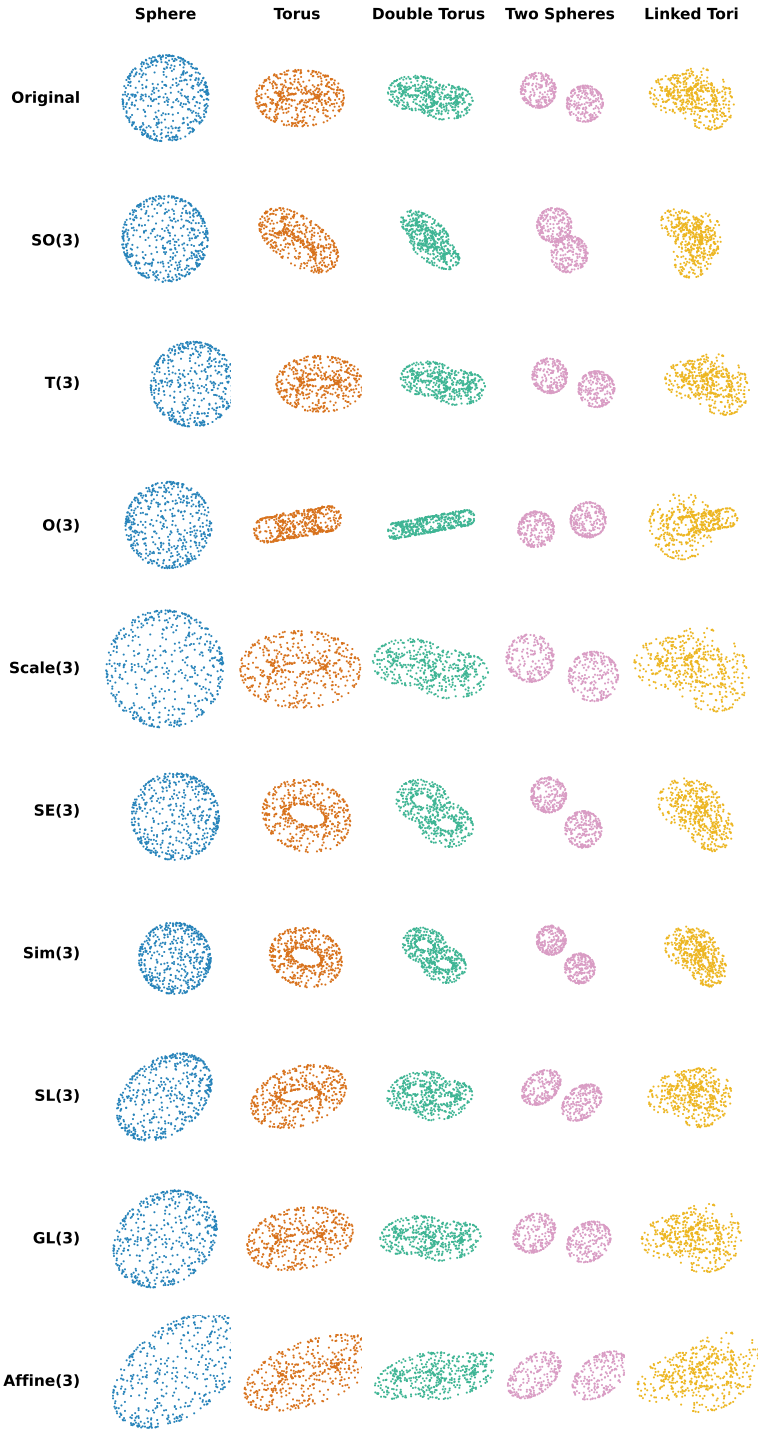


Figure 12: Illustration of augmentations from the different angle-preserving and non-angle-preserving symmetry augmentations that we consider in our experiments for the 3D Topology dataset.

Transformation	Symmetry Group	Train / Validation / Test	Generalization
Rotation	$\mathrm{SO}(3)$	$[0°, 90°)$	$[90°, 180°)$
Rotation and reflection	$\mathrm{O}(3)$	$[0°, 90°)$	$[90°, 180°)$
Translation	$\mathrm{T}(3)$	$[0, 1.5)$	$[1.5, 3)$
Scaling	$\mathrm{Scale}(3)$	$[\times 1, \times 1.5)$	$[\times 1.5, \times 2)$

Table 4: Ranges of augmentations used for training, validation, and testing, as well as for measuring the generalization of the model for the considered based symmetries in §4.

Equivariant targets. For a smooth surface defined implicitly as $F(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \mathbb{R}^3$, the unit normal at a point $\boldsymbol{x}$ is

$$\mathrm{N}(\boldsymbol{x}) = \frac{\nabla F(\boldsymbol{x})}{\|\nabla F(\boldsymbol{x})\|}. \tag{9}$$

Normal vectors for each point of the input are the $\mathrm{O}(3)$- and $\mathrm{SO}(3)$-equivariant targets in §6.

Given a segmentation map $\mathrm{S} : \mathbb{X} \to \mathbb{P}$ for a point cloud $\mathbb{X}$ and part identities $\mathbb{P} = \{1, \ldots, p\}$, the centroid of part k can be defined as

$$\mathrm{C}(k) = \frac{1}{|\,\mathrm{S}^{-1}(k)|} \sum_{\boldsymbol{x} \in \mathrm{S}^{-1}(k)} \boldsymbol{x}. \tag{10}$$

Mapping each point to the centroid of the part it belongs to requires a model to first determine the correct segment subset and then compute the mean over it. This poses the $\mathrm{T}(3)$- and $\mathrm{Scale}(3)$-equivariant task used in §6.

Training. We use the same small vanilla Transformer architecture as specified at the beginning of this section and §5, but drop the final mean aggregation to obtain per-point outputs. The training objective is a mean squared error loss.

C Additional experimental results

C.1 3D Tetris

3D Tetris [85] is a common synthetic benchmark for the study of point cloud invariance [86–88], and contains eight shapes, each consisting of four three-dimensional points, representing the center of each Tetris block shape (see bottom-left of Fig. 13). We extend this dataset by generating 50 samples per shape by adding small amounts of jitter and noise to the points of each original shape in order to increase the complexity of the task. Since the topology of Tetris shapes is identical, their only distinguishing features are geometric angles and distances, which are distorted under non-angle-preserving transformations ($\mathrm{SL}(3), \mathrm{GL}(3), \mathrm{Affine}(3)$), therefore the classification problem becomes ill-posed in such scenarios, and those symmetries are not expected to be learned too well. Similarly, the reflections introduced by the orthogonal group $\mathrm{O}(3)$ would also result in a restricted upper-bound classification performance for the models, since they will collapse the representations of the two classes with opposite chirality together.

As previously observed in §3, Figure 13 shows that there is again a stratification in the performance provided by the Transformer according to the type of symmetry groups. In particular, the results reinforce the previously identified increase in learnability of the symmetry when going from non-angle-preserving groups to compositional angle-preserving ones and then to base angle-preserving groups. We should note, however, that the labels of this standard dataset are not necessarily exactly invariant to all non-angle-preserving transformations regardless of their strength, which may account for their low observed performance.

In addition, when analyzing the models trained with this dataset (see Figs. 14–17, we find further evidence of the invariance mechanisms identified in §5.

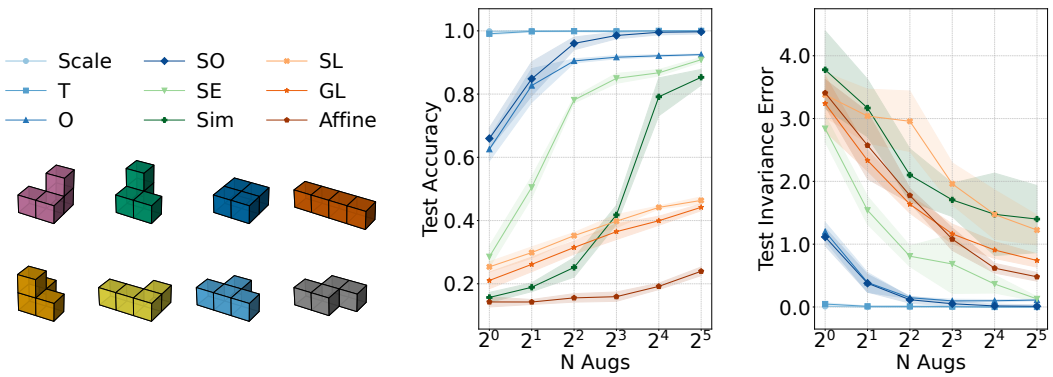


Figure 13: Illustration of samples of 3D Tetris (bottom-left) and results measuring the mean and standard deviation for test accuracies (middle) and invariance errors (right) across 5 runs with different random seeds. Angle-preserving symmetries (blue and green tones) are efficiently learned by the Transformer, while it struggles with non-angle-preserving ones (orange tones).

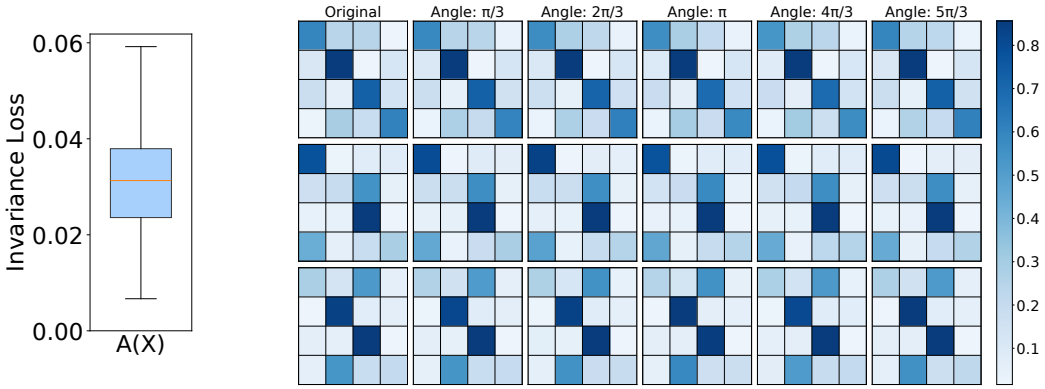


Figure 14: Attention maps of a full Transformer trained on Tetris augmented by 32 rotations per sample are very stable across augmentations, as also indicated by their low invariance loss.

C.2 Molecular property prediction

In order to complement the datasets obtained with the previous datasets, we also consider the QM9 dataset [89, 90], which differs in that it does not consist of point clouds with the same number of points. In order to overcome this difference, we implement a sequence length bucketing approach [91], which allows for more efficient training than padding all molecules to the same length. We split the dataset into 80% of training samples, 10% validation samples, and 10% test samples. Just like for ModelNet10, we carry out an out-of-distribution analysis to measure the robustness with which base angle-preserving symmetries are learned in this dataset. In that sense, we again define the training, validation, test, and generalization ranges of augmentations as shown in Table 4.

Fig. 18 shows that the Transformer largely preserves its task performance and invariance behavior on the OOD generalization set. Thus, although the model is trained using only a limited number of augmentations sampled from a narrow range of transformations, it is still able to extrapolate approximately invariant behavior to a disjoint range of unseen transformations at test time. Moreover, the trends in both task and invariance losses are similar on both test and OOD generalization. Increasing the number of training augmentations within the restricted range also improves performance under larger, unseen transformations. This observation provides a strong indication that the model is learning an underlying mechanism that encodes invariance to the symmetry in a robust and generalizable manner. The main exception is translation, for which OOD generalization performance lags behind rotations and scaling. This indicates that the robustness and generality of the learned mechanism for translation invariance is more sensitive to the number of augmentations observed during training.

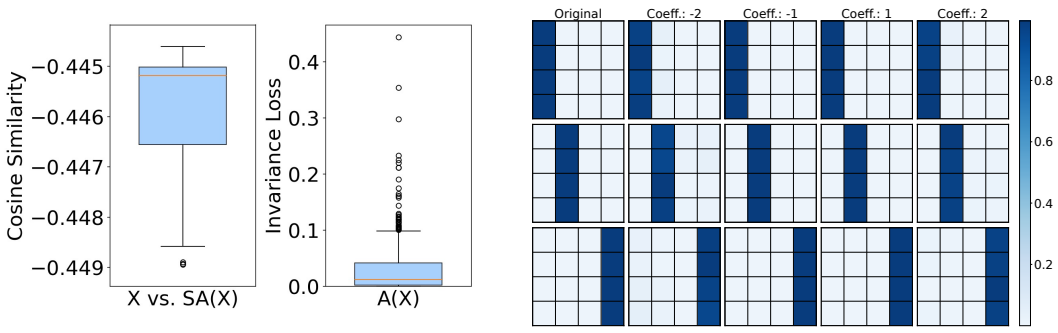


Figure 15: Results for the first block a full Transformer trained on Tetris with 32 translations per sample. Right: Each row shows the attention maps of a sample (Original) and four of its augmentations over the full training range. A coefficient of, e.g., 2 means the points are shifted by 2 in the positive direction for each axis. Samples from the same orbit focus their attention on the same landmark point. Middle: Invariance loss of the attention maps is low for the entire test set. Left: Cosine similarity between $\boldsymbol{X}$ and $\mathrm{SA}(\boldsymbol{X})$ is strictly negative, i.e. the attention output has signs opposite to those of the residuals.

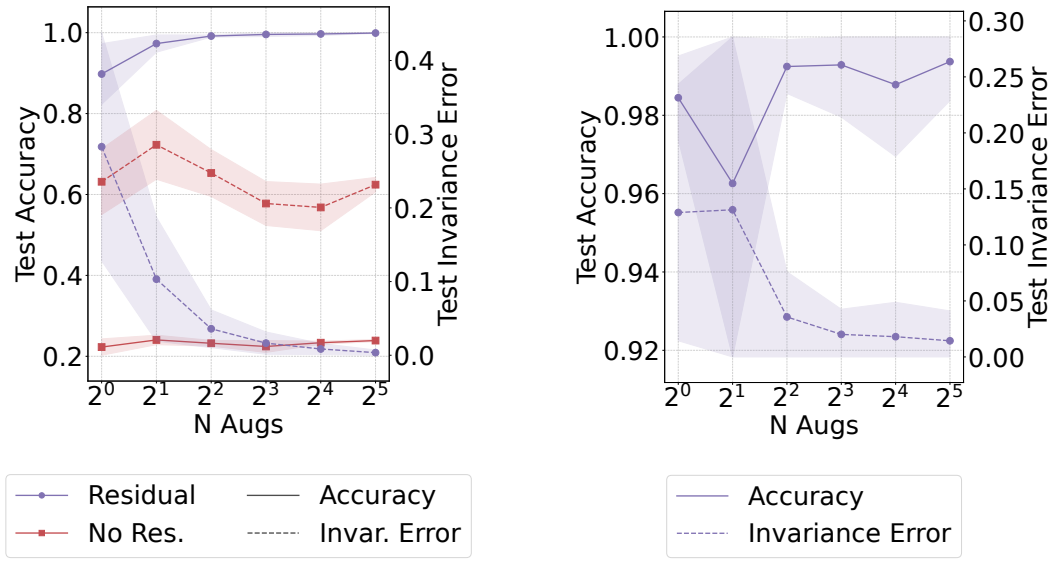


Figure 16: Mechanism ablations for Tetris dataset. Left: A model with 2 blocks of $\boldsymbol{X} + \mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ (*Residual*) can learn to be translation invariant and solve the Tetris task, but $\mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ (*No Res.*) cannot. Right: A model consisting only of $\mathrm{SA}(\mathrm{LN}(\boldsymbol{X}))$ blocks can learn to be scale invariant and solve the Tetris task.

C.3 Additional architectural components

In order to extend our analysis from the vanilla Transformer architecture towards more realistic models, we explore the effect of additional architectural components, namely positional encodings and multi-head attention. These experiments provide preliminary evidence that the principal trends are not solely consequences of single-head attention or the complete absence of positional information.

C.3.1 Positional encodings

Positional encodings are a common component of modern Transformer-based architectures. Self-attention treats the inputs as an unordered collection of elements and is equivariant to their permutations. Positional encodings break or constrain this symmetry by associating tokens with information about, e.g., their absolute coordinates, their relative displacements, or other geometrical relationships [92, 93]. For spatial data, encodings can similarly be designed to provide information about distances

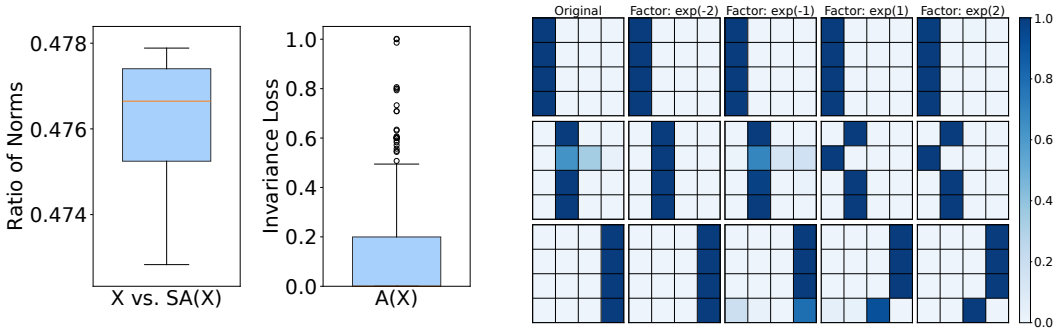


Figure 17: Mechanism indicators for the second block of a Transformer trained on Tetris with 32 scale augmentations. Left: points in the attention output have larger norms than points in the residual. The non-invariance of the residuals is therefore reduced to the noise level. Middle and Right: Attention maps are mostly invariant under scalings. Different rows correspond to different samples.

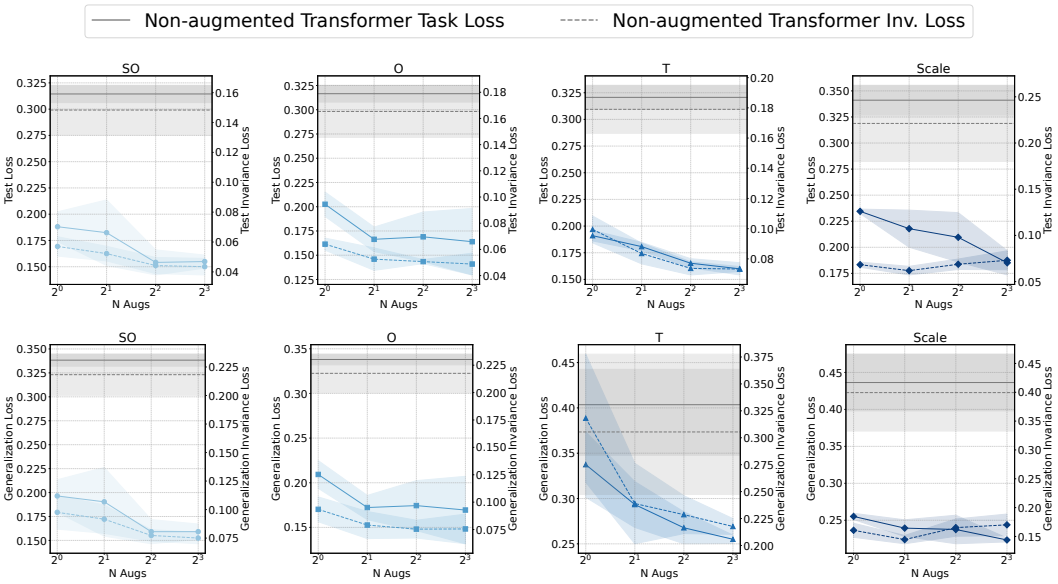


Figure 18: Test (in-distribution) and OOD generalization (disjoint distribution) task (solid line) and invariance losses (dashed line) for base angle-preserving symmetries in the QM9 dataset across 3 random seeds reported as MAE in eVs (lower is better). The gray lines represent the performance of a Transformer trained without data augmentation and tested in each augmented testing set as a reference to evaluate the improvement derived from data augmentation. All symmetries are learned robustly with enough augmentations, with translations being the most sensitive to distribution shifts.

or directions, thereby introducing an inductive bias adapted to the geometry of the domain. There are also methods that have explored the definition of equivariant self-attention layers through the design of positional encodings that are invariant under selected group transformations [94].

In the following experiments, we explore how incorporating geometrical information into the model through positional encodings affects the performance observed for the synthetic 3D Tetris dataset, which contains inputs represented as point clouds $\mathbb{X} = \{\boldsymbol{x}_i\}_{i=1}^{T}$, $\boldsymbol{x}_i \in \mathbb{R}^d$.

Geometric sinusoidal absolute positional encoding. We adapt the sinusoidal absolute positional encoding of **(author?)** [92] to spatial coordinates by applying sinusoidal features independently to each coordinate axis. In particular, for each coordinate axis $a \in \{1, \dots, d\}$ and frequency index $m \in \{1, \dots, F\}$, define $\omega_m = B^{-m/F}$, where B is usually $10,000$. The geometric sinusoidal features are

$$\boldsymbol{\gamma}(\boldsymbol{x}_i) = \Big[\sin(\omega_m x_{i,a}), \cos(\omega_m x_{i,a})\Big]_{a=1,\dots,d;\, m=1,\dots,F}.$$

These features are then projected to the model dimension and added to the token embedding $\boldsymbol{h}_i$: $\tilde{\boldsymbol{h}}_i = \boldsymbol{h}_i + \boldsymbol{W}_\gamma \boldsymbol{\gamma}(\boldsymbol{x}_i)$. This encoding provides the model with information about the absolute coordinates of each point.

Geometric ALiBi. We adapt Attention with Linear Biases (ALiBi) [95] to geometric distances. Standard ALiBi applies a linear penalty based on separation in the sequence. Here, we instead penalize attention between points according to their geometric distance:

$$\tilde{a}_{ij} = a_{ij} - m\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_p}{s},$$

where a_{ij} is the original attention logit, $m > 0$ is the slope associated with the attention head, p specifies the distance metric, and $s > 0$ is a scale factor. The bias therefore favors interactions between points that are close under the chosen metric.

Geometric learned relative position bias. Following learned relative-position representations and spatial relative-position bias tables [93, 96], we discretize the pairwise displacement between points and associate each displacement bucket with a learned scalar bias.

We define the integer displacement as $\boldsymbol{\delta}_{ij} = \mathrm{clip}\left(\mathrm{round}(\boldsymbol{x}_j - \boldsymbol{x}_i), -K, K\right) \in \{-K, \dots, K\}^d$. A learned lookup table maps each displacement bucket to a scalar: $b : \{-K, \dots, K\}^d \to \mathbb{R}$. The attention logit is then modified as $\tilde{a}_{ij} = a_{ij} + b(\boldsymbol{\delta}_{ij})$.

Geometric RoPE. We also consider a multidimensional, coordinate-based version of Rotary Position Embedding (RoPE) [97]. In standard RoPE, query and key vectors are rotated by phases determined by their sequence positions. In the geometric setting, these phases can instead be determined by the associated spatial coordinates.

For each rotary pair r, we define a frequency vector $\boldsymbol{w}_r \in \mathbb{R}^d$, and the phase associated with point $\boldsymbol{x}_i$ as $\theta_r(\boldsymbol{x}_i) = \langle \boldsymbol{x}_i, \boldsymbol{w}_r \rangle$. Within each two-dimensional rotary subspace, we apply the rotation matrix

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

As a result, we obtain the modified queries and keys $\tilde{\boldsymbol{q}}_i = \boldsymbol{R}(\theta(\boldsymbol{x}_i))\boldsymbol{q}_i$, $\tilde{\boldsymbol{k}}_j = \boldsymbol{R}(\theta(\boldsymbol{x}_j))\boldsymbol{k}_j$, where $\boldsymbol{R}(\theta(\boldsymbol{x}_i))$ denotes the block-diagonal operator formed from the pairwise rotations, and the attention logits become $\tilde{a}_{ij} = \frac{\langle \tilde{\boldsymbol{q}}_i, \tilde{\boldsymbol{k}}_j \rangle}{\sqrt{d}}$. Because planar rotations compose through differences of their angles, each rotary pair depends on the relative phase $\theta_r(\boldsymbol{x}_j) - \theta_r(\boldsymbol{x}_i) = \langle \boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{w}_r \rangle$. Thus, although the rotations themselves are defined using absolute coordinates, the contribution to the query-key inner product depends on the relative displacement between the pair of points.

Results. We carry out experiments using the synthetic 3D Tetris dataset discussed in §C.1. As we can see in Figures 19 and 20, the test performance of the Transformer remains stable across each of the previously defined geometric positional encodings, and is itself similar to the original results without positional encodings (`no_pe`). In addition, repeating the mechanistic diagnostics for the base groups (Fig. 21) also shows that orbit-stable attention maps and the principal activation signatures remain qualitatively present. As a result, these experiments suggest that, within the coordinate-token setting, adding auxiliary geometric positional information does not eliminate the tendency to learn the reported invariant computations.

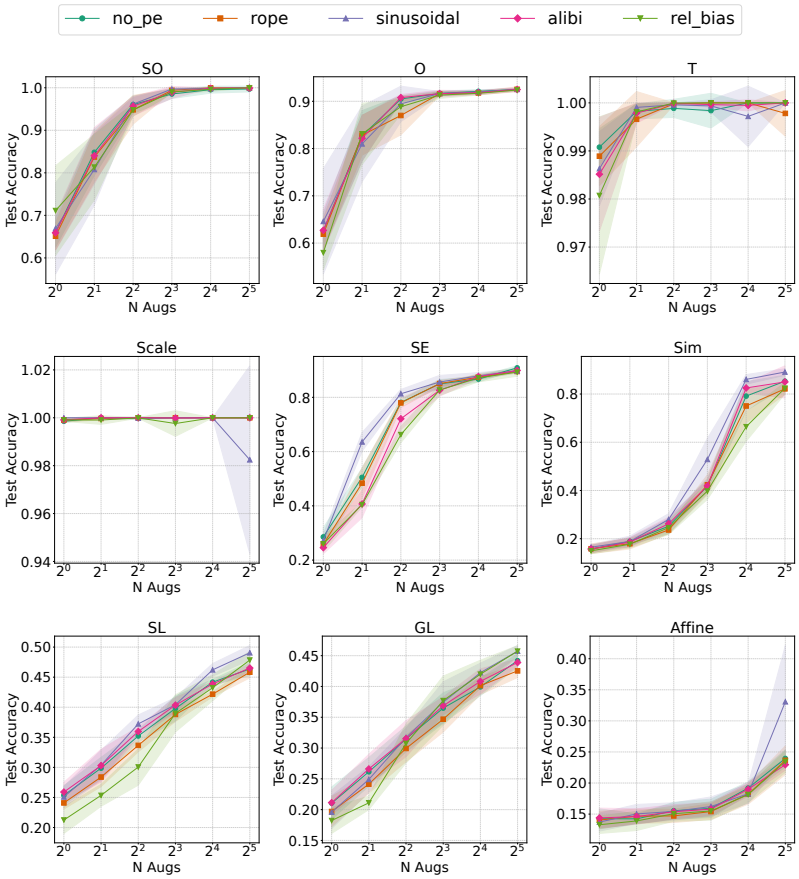


Figure 19: Test accuracy for different positional encoding approaches as measured for each symmetry group considered in the 3D Tetris dataset.

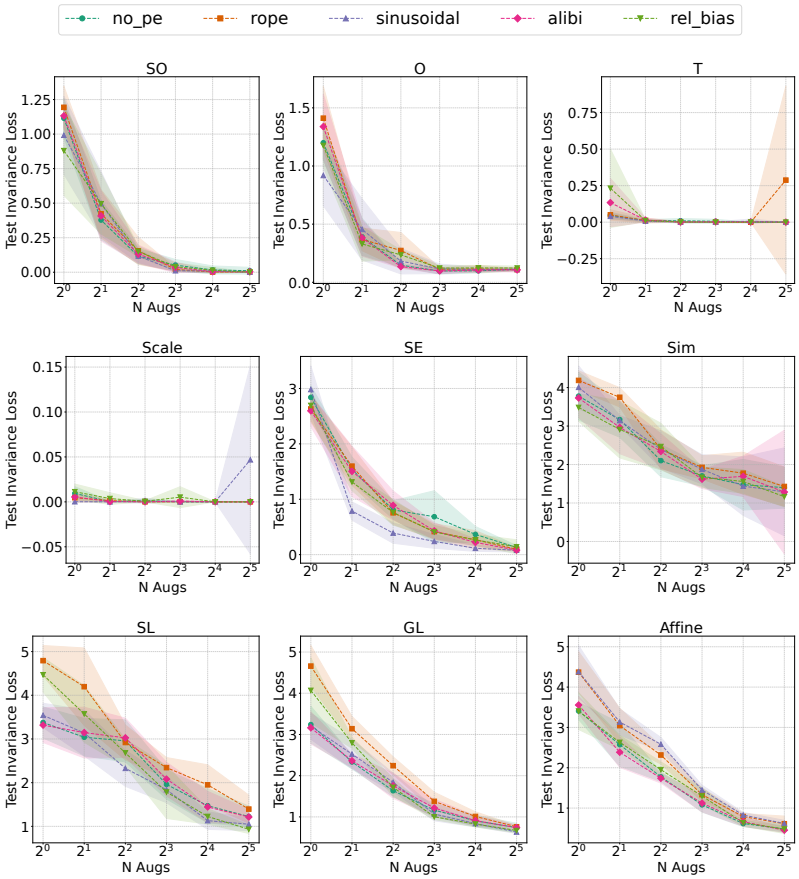


Figure 20: Test invariant error for different positional encoding approaches as measured for each symmetry group considered in the 3D Tetris dataset.

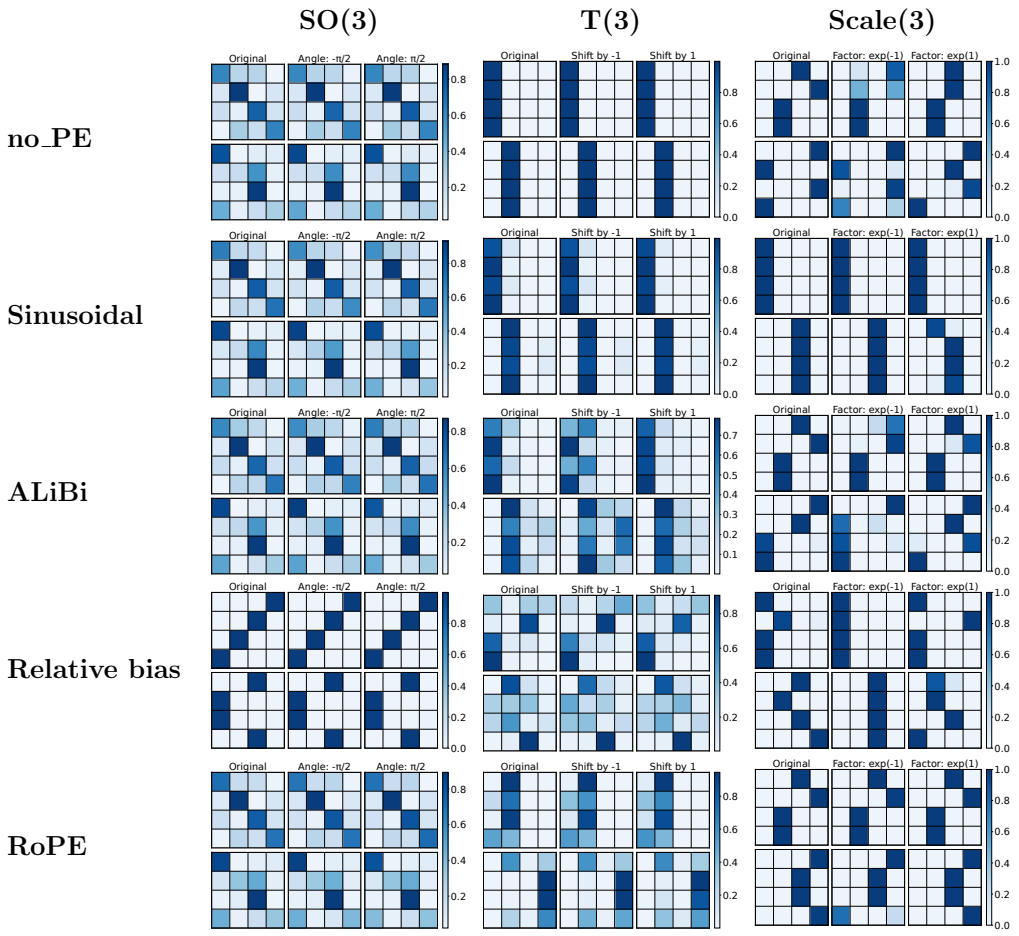


Figure 21: Attention softmax for first layer of a Transformer trained on 32 augmentations using each considered geometric positional encoding and for each base angle-preserving symmetry. The attention maps remain approximately invariant across augmentations for all positional encodings.

C.3.2 Multi-head attention

We carry out further experiments on the 3D Tetris dataset using 2 and 4 attention heads, respectively. To observe extrapolation behavior, we use the OOD Setting defined in §4 and §B.4 for base angle-preserving groups. As can be seen in Figure 23, the qualitative behavior is very consistent across head counts. The only visible exception is extrapolation on $\mathrm{T}(3)$, which seems sensitive to hyperparameter settings in general with no clear trend regarding number of attention heads. Quantitatively, the performance on compositional groups both in terms of accuracy and invariance shows small improvements when using more heads. This suggests that heads might be specializing in base subgroups, easing the pressure to become invariant to multiple transformation types at once. However, we see no empirical evidence for this hypothesis. As shown in Figure 22 by way of example, it is often one head that is more invariant than the other on all present base subgroups. Either the attention heads work together by a different mechanism or since performance gains are small in the first place, they might be attributable to a more flexible adaptation to the data distribution rather than a generalizable change of mechanism. Overall, the number of attention heads seems to have no major impact on the results presented in this paper.

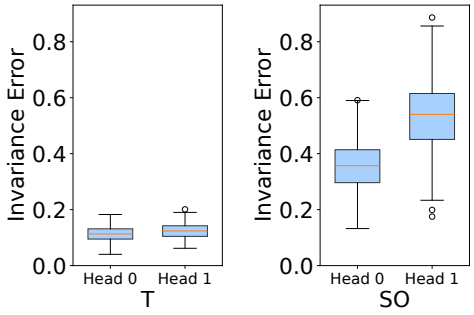


Figure 22: Invariance of attention heads to $\mathrm{T}(3)$ (left) and $\mathrm{SO}(3)$ (right) for first layer of a Transformer trained with 32 $\mathrm{SE}(3)$-augmentations per sample. Head 0 is more invariant than head 1 to both base subgroups, so they do not seem to specialize in either group.

C.4 Signature of detected mechanisms for additional models

C.4.1 Translation

As we can see in Figure 24, the detected mechanism prevails across seeds and number of augmentations for the first blocks of Transformers trained on ModelNet10 with translations. That is, in the first block, self-attention subtracts approximately equivariant features from the residuals.

If strict invariance were introduced in the first block, invariance of the second block would follow immediately. In the approximate scenario, the only way to destroy invariance after the first block would be to amplify small invariance errors. But as shown in Figure 25, the second block also displays highly invariant attention maps. However there is no need for the invariant features to be subtracted again, as reflected by the positive cosine similarities.

C.4.2 Rotation

Figure 26 shows that the attention maps of both blocks of Transformers trained on ModelNet10 with rotation augmentations exhibit strong rotation invariance. In both cases, increasing the number of augmentations induces higher invariance, as can be expected.

C.4.3 Scale

When trained with sufficiently many scale augmentations, Transformers exhibit the signature of scaling up self-attention outputs with respect to the residuals in both blocks (Figures 27 and 28).

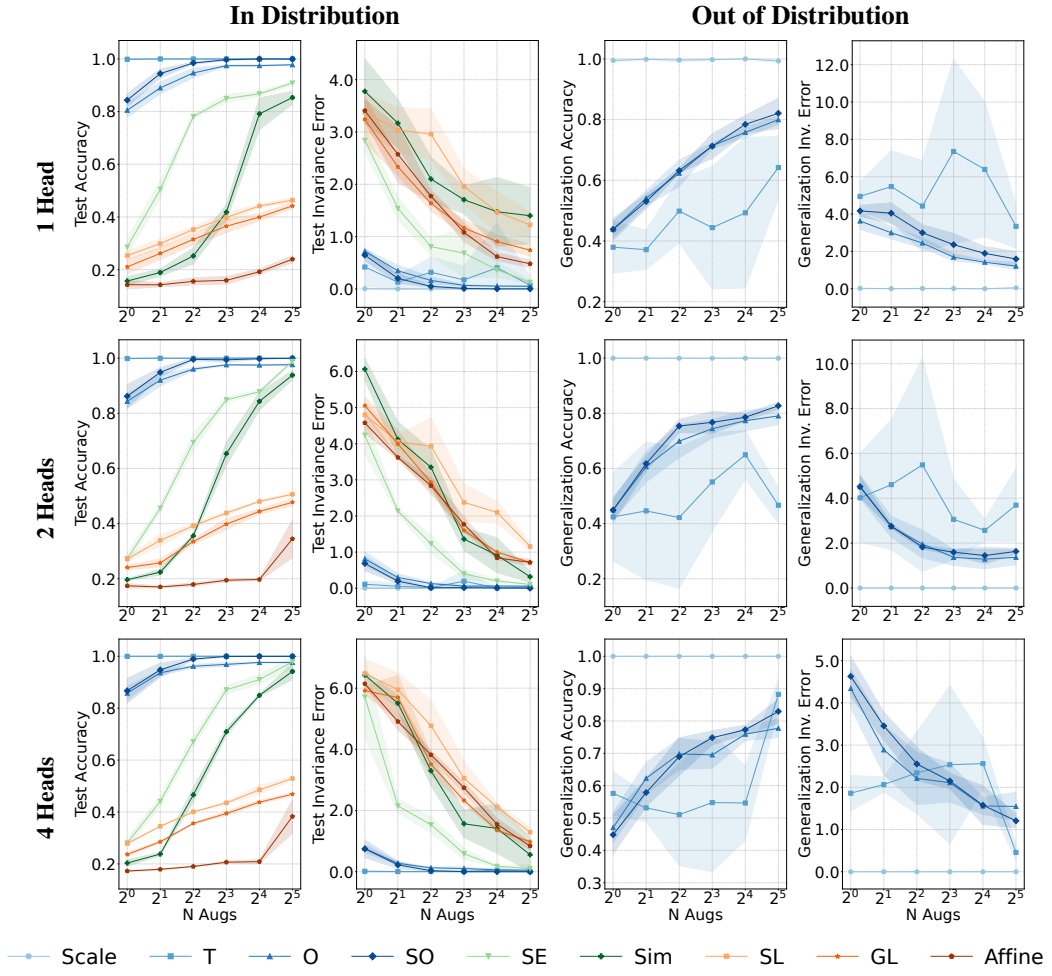


Figure 23: In- and out-of-distribution results for 3D Tetris comparing 1, 2, and 4 attention heads. Increasing the number of heads slightly increases performance on compositional groups. The only visible qualitative difference seems to be extrapolation on $\mathrm{T}(3)$, which seems sensitive to hyperparameter settings with no clear trend.

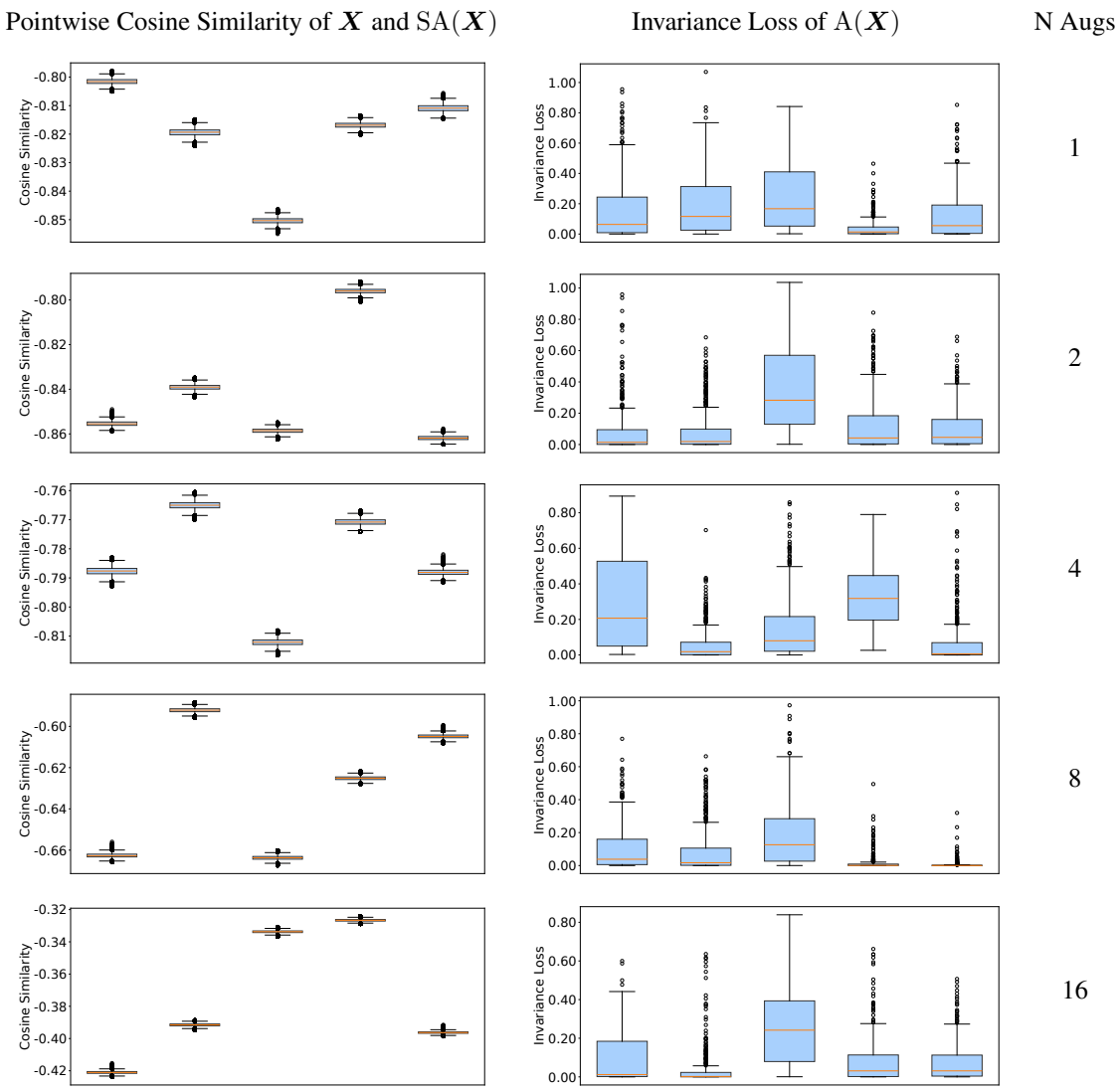


Figure 24: Mechanism indicators for first block of all models trained on ModelNet10 with translations.

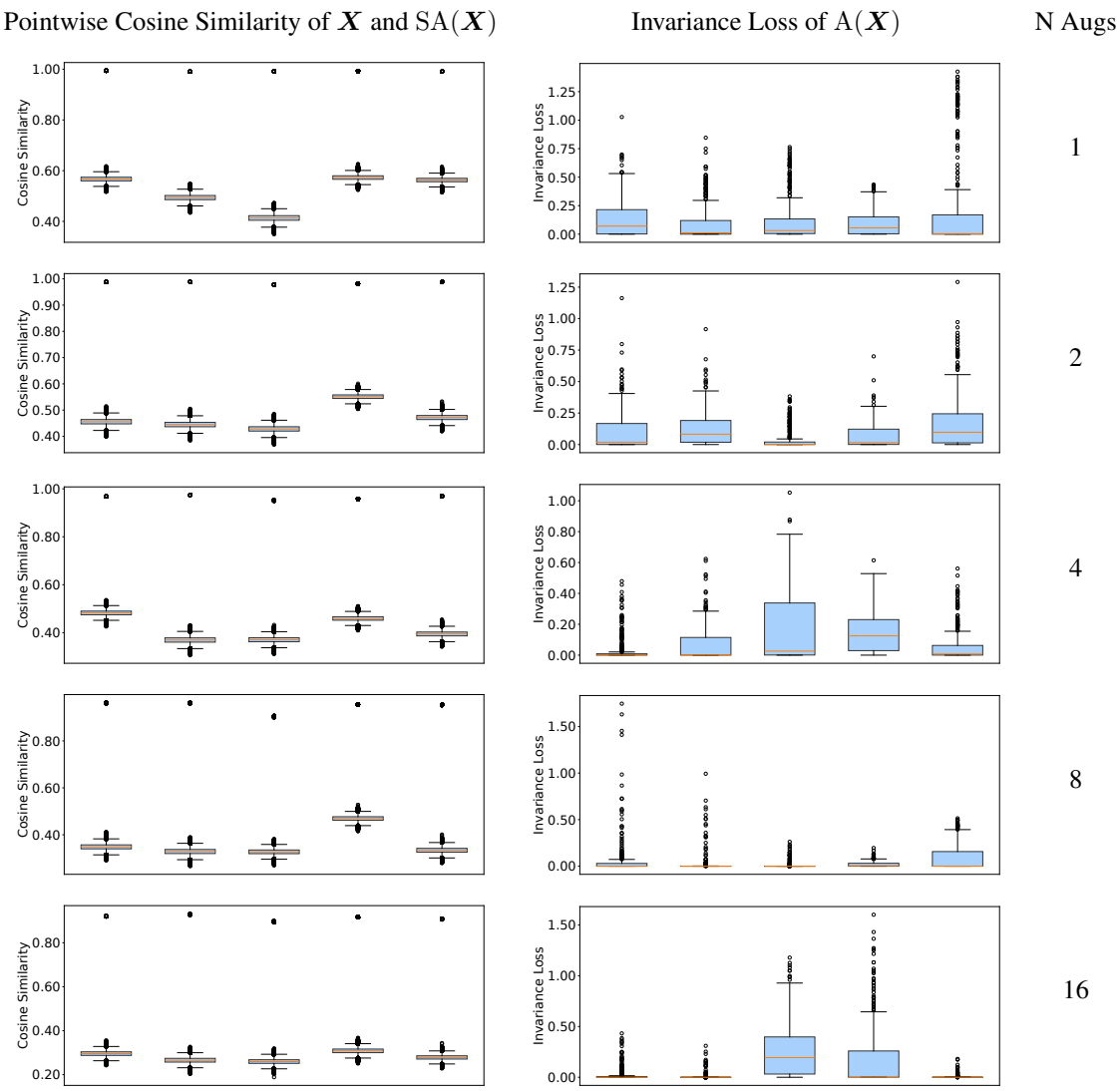


Figure 25: Mechanism indicators for second block of all models trained on ModelNet10 with translations.

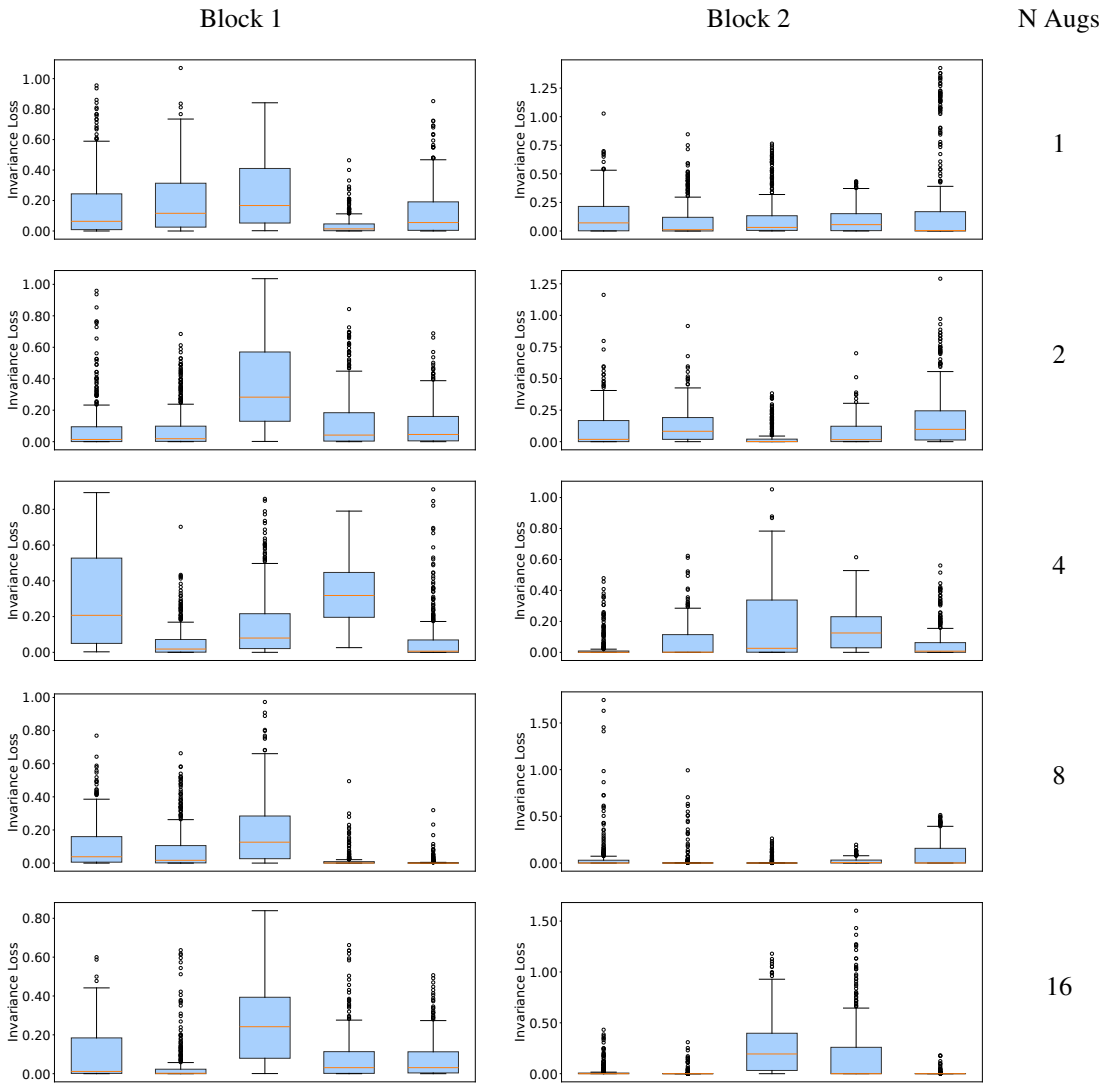


Figure 26: Invariance loss per block of the attention maps of all models trained with rotations on ModelNet10.

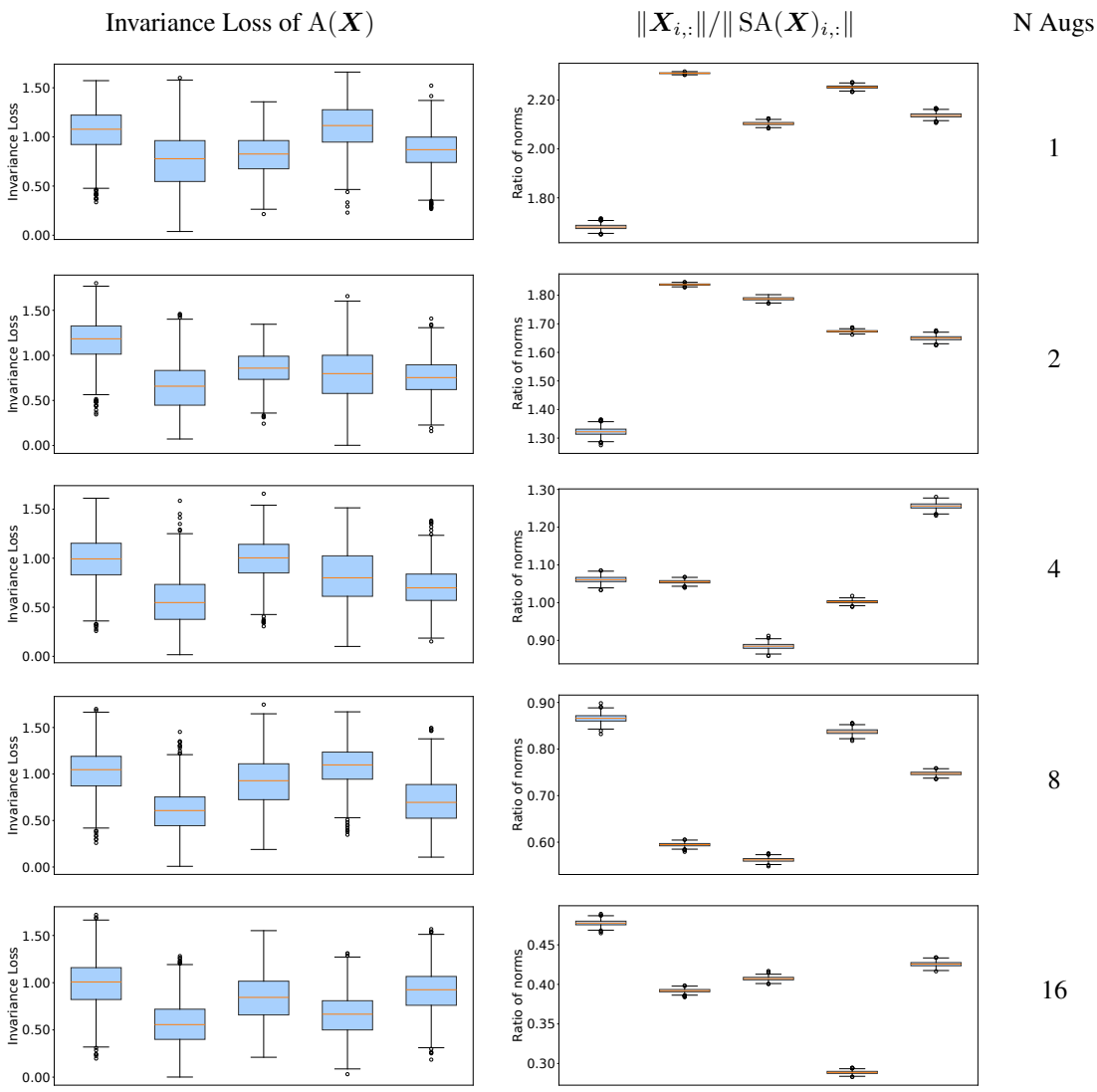


Figure 27: Mechanism indicators of first block of all models trained on ModelNet10 with scale augmentations.

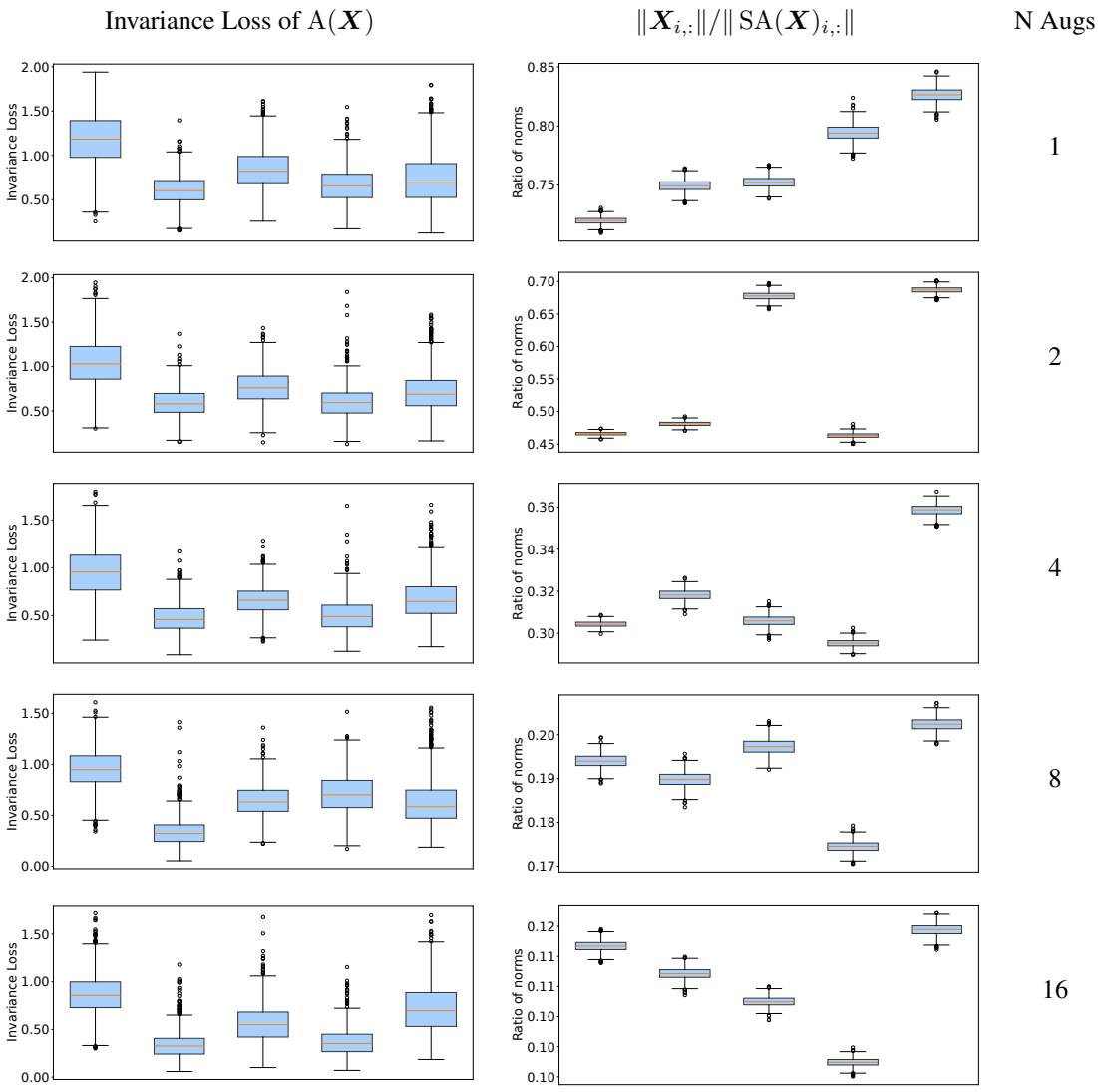


Figure 28: Mechanism indicators of second block of all models trained on ModelNet10 with scale augmentations.

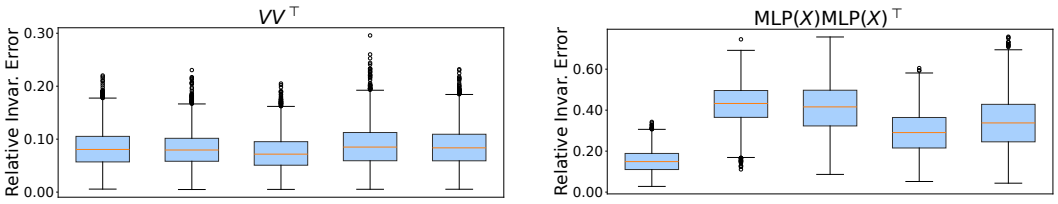


Figure 29: Comparison of equivariance of $\boldsymbol{V}$ and $\mathrm{MLP}(\boldsymbol{X})$ in the first block of a Transformer trained on ShapeNet normals prediction with 16 $\mathrm{O}(3)$ augmentations. As a proxy, we measure the invariance of inner products, divided by a random baseline. Latent representations after the MLP appear less equivariant than those after the value projection, but more equivariant than the random baseline.

C.5 Further results on equivariance

We provide further insight into how approximate $\mathrm{O}(3)$-equivariance is maintained throughout a forward pass and isolate first components that may be relevant to the learned equivariant mechanisms for other base angle preserving groups.

C.5.1 $\mathrm{O}(3)$-equivariance of remaining architecture

Once invariance is induced in a model, it cannot be destroyed by subsequent modules. The same does not hold true for equivariance: Once established, it needs to be maintained by the entire following computation to hold for the output.

So while we have proposed a mechanism to ensure O-equivariance in the self-attention module SA in §6, this property needs to be supported by the remaining architectural components as well.

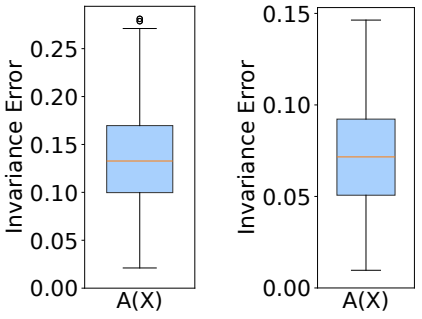
The residual connections do not pose a problem here, since for any module M and $\boldsymbol{R} \in \mathrm{O}(d)$, if $\mathrm{M}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{M}(\boldsymbol{X})\boldsymbol{R}$, then $\mathrm{M}(\boldsymbol{X}\boldsymbol{R}) + \boldsymbol{X}\boldsymbol{R} = \big(\mathrm{M}(\boldsymbol{X}) + \boldsymbol{X}\big)\boldsymbol{R}$.

The MLP on the other hand is more problematic in this context. Strict equivariance to $\mathrm{O}(d)$ would require zero bias terms and isotropic projection matrices, which again we do not observe in trained models. So as for the value projection, we resort to measuring a proxy for the empirical equivariance error of the MLP's output by computing the invariance error of its inner product. To make the metric more comparable between modules, we normalize by a random baseline. Concretely, we divide by the error obtained when matching original samples to augmentations of random other samples. This should compensate for varying overall spread in the latent space at different layers. As we can see in Figure 29, forwarding through MLP increases the relative invariance error of inner products, while still remaining significantly below one.

Overall, our findings suggest that approximate equivariance is maintained by each submodule individually, instead of being introduced towards the end of the forward pass, potentially by a shared mechanism spanning all components. Wherever equivariance on the full ambient space of the input would be too restrictive on the parameter space, models rely on data-dependent approximations instead.

C.5.2 Other base groups: attention maps are invariant

For other base angle preserving groups, namely translation and scale, we also observe approximately invariant attention maps, see Figure 30. This further supports our claim that learning invariant building blocks can be helpful in learning equivariance. However as of now, the actual mechanisms and the role of the invariant attention maps remain to be identified in future work.



(a) Scale (b) Translation

Figure 30: Results for first block of Transformers trained on ShapeNet per-part centroid prediction with 16 $\mathrm{Scale}(3)$ (left) or $\mathrm{T}(3)$ augmentations (right) per sample. Both attention maps are approximately invariant to the respective group.

D Proofs

We include the proof of our theoretical result in §5. The numbering of the statements follows the numbering used in the paper.

D.1 Proof of Lemma 1

Lemma 1. *Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and let $\boldsymbol{B} = \boldsymbol{W}_Q\boldsymbol{W}_K^\top \in \mathbb{R}^{d\times d}$ and $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}) = \frac{1}{\sqrt{d}}\boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\top$. Let $\boldsymbol{P}_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ denote the centering matrix acting on the columns of a score matrix. Then, for any $\boldsymbol{R} \in \mathrm{O}(d)$, the following are equivalent: $\mathrm{A}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X})$, and $(\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R}) - \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}))\,\boldsymbol{P}_n = \mathbf{0}$. Equivalently, we have*

$$\boldsymbol{X}\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{X}^\top\boldsymbol{P}_n = \mathbf{0}.$$

In row-wise notation, this means that for all query indices i and all key indices j, k, $\boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)(\boldsymbol{x}_j - \boldsymbol{x}_k) = 0$. Thus attention-map invariance along an augmentation orbit only requires the row-wise score differences to be invariant.

Proof. Let $\boldsymbol{S} = \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X})$, $\widetilde{\boldsymbol{S}} = \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R})$, and $\boldsymbol{D} = \widetilde{\boldsymbol{S}} - \boldsymbol{S}$. Note that for two row vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$,

$$\mathrm{softmax}(\boldsymbol{u}) = \mathrm{softmax}(\boldsymbol{v})$$

if and only if there exists a scalar $c \in \mathbb{R}$ such that $\boldsymbol{u} - \boldsymbol{v} = c\mathbf{1}_n^\top$. If $\boldsymbol{u} = \boldsymbol{v} + c\mathbf{1}_n^\top$, then the factor e^c cancels between the numerator and denominator of the softmax. Conversely, if the softmaxes are equal, then for every coordinate j, $\frac{e^{u_j}}{\sum_{\ell=1}^n e^{u_\ell}} = \frac{e^{v_j}}{\sum_{\ell=1}^n e^{v_\ell}}$.

Taking logarithms gives $u_j - v_j = \log\sum_{\ell=1}^n e^{u_\ell} - \log\sum_{\ell=1}^n e^{v_\ell}$, which is independent of j. Hence $\boldsymbol{u} - \boldsymbol{v}$ is constant across coordinates.

Since $\mathrm{A}(\boldsymbol{X})$ is obtained by applying softmax row-wise to $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X})$, we therefore have $\mathrm{A}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X})$ if and only if, for every row index i, there exists a scalar c_i such that $\boldsymbol{D}_{i,:} = c_i\mathbf{1}_n^\top$. Equivalently, every row of $\boldsymbol{D}$ is constant across columns.

Now recall that $\boldsymbol{P}_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix. For any row vector $\boldsymbol{z} \in \mathbb{R}^n$, $\boldsymbol{z}\boldsymbol{P}_n = \boldsymbol{z} - \frac{1}{n}\left(\sum_{j=1}^n z_j\right)\mathbf{1}_n^\top$. Thus $\boldsymbol{z}\boldsymbol{P}_n = \mathbf{0}^\top$ if and only if $\boldsymbol{z}$ is constant across coordinates. Applying this row by row, we obtain

$$\mathrm{A}(\boldsymbol{X}\boldsymbol{R}) = \mathrm{A}(\boldsymbol{X}) \quad\Longleftrightarrow\quad \boldsymbol{D}\boldsymbol{P}_n = \mathbf{0}.$$

By substituting the definition of $\boldsymbol{D}$, we have $(\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R}) - \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}))\,\boldsymbol{P}_n = \mathbf{0}$.

Since $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}) = \frac{1}{\sqrt{d}}\boldsymbol{X}\boldsymbol{B}\boldsymbol{X}^\top$ and $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R}) = \frac{1}{\sqrt{d}}(\boldsymbol{X}\boldsymbol{R})\boldsymbol{B}(\boldsymbol{X}\boldsymbol{R})^\top = \frac{1}{\sqrt{d}}\boldsymbol{X}\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top\boldsymbol{X}^\top$, we have $\boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}\boldsymbol{R}) - \boldsymbol{S}_{\boldsymbol{B}}(\boldsymbol{X}) = \frac{1}{\sqrt{d}}\boldsymbol{X}\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{X}^\top$.

Since the scalar factor $1/\sqrt{d}$ is nonzero, the condition above is equivalent to $\boldsymbol{X}\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{X}^\top\boldsymbol{P}_n = \mathbf{0}$.

Finally, the row-wise form follows by looking at differences between two columns j and k of the same row. The (i,j) entry of $\boldsymbol{X}\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{X}^\top$ is $\boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{x}_j$. Henece, each row is constant across columns if and only if, for all i, j, k, $\boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{x}_j - \boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)\boldsymbol{x}_k = 0$, which is $\boldsymbol{x}_i^\top\left(\boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^\top - \boldsymbol{B}\right)(\boldsymbol{x}_j - \boldsymbol{x}_k) = 0$.

□

D.2 Proof of Lemma 2

Lemma 2. *Let $\Lambda : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times n}$ be translation invariant with row-stochastic outputs. Set $\boldsymbol{\Lambda} = \Lambda(\boldsymbol{X})$. Then $f(\boldsymbol{X}) = g(\boldsymbol{X} - \boldsymbol{\Lambda}\boldsymbol{X})$ is translation invariant for any function g.*

Proof. Let $\boldsymbol{t} \in \mathbb{R}^d$ and $\boldsymbol{T} = \mathbf{1}_n\boldsymbol{t}^\top$ the matrix with $\boldsymbol{t}$ in each row. Since $\boldsymbol{\Lambda}$ is row-stochastic, $\boldsymbol{\Lambda}\mathbf{1}_n = \mathbf{1}_n$, so $\boldsymbol{\Lambda}\boldsymbol{T} = \boldsymbol{T}$. By translation invariance of Λ, $\Lambda(\boldsymbol{X} + \boldsymbol{T}) = \boldsymbol{\Lambda}$. Hence

$$(\boldsymbol{X} + \boldsymbol{T}) - \boldsymbol{\Lambda}(\boldsymbol{X} + \boldsymbol{T}) = \boldsymbol{X} - \boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{T} - \boldsymbol{\Lambda}\boldsymbol{T} = \boldsymbol{X} - \boldsymbol{\Lambda}\boldsymbol{X},$$

and so $f(\boldsymbol{X}+\boldsymbol{T}) = f(\boldsymbol{X})$. □

D.3 Proof of Lemma 3

Lemma 3. *Let* $\mathrm{SA}(\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})\boldsymbol{X}\boldsymbol{W}_V$ *be a single attention-only layer. If for some* $c > 0$, $\mathrm{A}(c\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})$, *then* $\mathrm{SA}(c\boldsymbol{X}) = c\,\mathrm{SA}(\boldsymbol{X})$. *Hence, an attention-only stack whose attention maps are stable over the scale augmentation range is positively homogeneous* $\Phi(c\boldsymbol{X}) = c\,\Phi(\boldsymbol{X})$. *If the final prediction is made by taking an* argmax *over logits that are scaled by this same positive factor, then the predicted class is invariant to* c.

Proof. For a single attention-only layer, we compute directly. Since $\mathrm{SA}(\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})\boldsymbol{X}\boldsymbol{W}_V$, we have $\mathrm{SA}(c\boldsymbol{X}) = \mathrm{A}(c\boldsymbol{X})(c\boldsymbol{X})\boldsymbol{W}_V$. By assumption, $\mathrm{A}(c\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})$. Therefore $\mathrm{SA}(c\boldsymbol{X}) = \mathrm{A}(\boldsymbol{X})(c\boldsymbol{X})\boldsymbol{W}_V = c\,\mathrm{A}(\boldsymbol{X})\boldsymbol{X}\boldsymbol{W}_V = c\,\mathrm{SA}(\boldsymbol{X})$. Thus the layer is positively homogeneous of degree one on any scale orbit on which its attention map is unchanged.

We now extend this argument to a stack of attention-only layers. Let $\boldsymbol{H}^{(0)}(\boldsymbol{X}) = \boldsymbol{X}$, and $\boldsymbol{H}^{(\ell+1)}(\boldsymbol{X}) = \mathrm{SA}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X}))$, where $\mathrm{SA}_\ell(\boldsymbol{H}) = \mathrm{A}_\ell(\boldsymbol{H})\boldsymbol{H}\boldsymbol{W}_{V,\ell}$. Assume that, over the scale augmentation range under consideration, the attention maps are stable at every layer, meaning $\mathrm{A}_\ell(\boldsymbol{H}^{(\ell)}(c\boldsymbol{X})) = \mathrm{A}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X}))$ for every layer ℓ.

We prove by induction that $\boldsymbol{H}^{(\ell)}(c\boldsymbol{X}) = c\,\boldsymbol{H}^{(\ell)}(\boldsymbol{X})$ for all layers ℓ. Then, we have $\boldsymbol{H}^{(0)}(c\boldsymbol{X}) = c\boldsymbol{X} = c\,\boldsymbol{H}^{(0)}(\boldsymbol{X})$. Now suppose that $\boldsymbol{H}^{(\ell)}(c\boldsymbol{X}) = c\,\boldsymbol{H}^{(\ell)}(\boldsymbol{X})$. Then,

$$\begin{aligned}
\boldsymbol{H}^{(\ell+1)}(c\boldsymbol{X}) &= \mathrm{SA}_\ell(\boldsymbol{H}^{(\ell)}(c\boldsymbol{X})) \\
&= \mathrm{A}_\ell(\boldsymbol{H}^{(\ell)}(c\boldsymbol{X}))\boldsymbol{H}^{(\ell)}(c\boldsymbol{X})\boldsymbol{W}_{V,\ell} \\
&= \mathrm{A}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X}))\left(c\,\boldsymbol{H}^{(\ell)}(\boldsymbol{X})\right)\boldsymbol{W}_{V,\ell} \\
&= c\;\mathrm{A}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X}))\boldsymbol{H}^{(\ell)}(\boldsymbol{X})\boldsymbol{W}_{V,\ell} \\
&= c\,\boldsymbol{H}^{(\ell+1)}(\boldsymbol{X}).
\end{aligned}$$

Hence, by induction, the whole attention-only stack is positively homogeneous $\Phi(c\boldsymbol{X}) = c\,\Phi(\boldsymbol{X})$.

The same conclusion holds if the attention-only stack includes residual connections of the form $\boldsymbol{H}^{(\ell+1)}(\boldsymbol{X}) = \boldsymbol{H}^{(\ell)}(\boldsymbol{X}) + \mathrm{SA}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X}))$.

We see that using the same induction hypothesis and the single-layer homogeneity above,

$$\begin{aligned}
\boldsymbol{H}^{(\ell+1)}(c\boldsymbol{X}) &= \boldsymbol{H}^{(\ell)}(c\boldsymbol{X}) + \mathrm{SA}_\ell(\boldsymbol{H}^{(\ell)}(c\boldsymbol{X})) \\
&= c\,\boldsymbol{H}^{(\ell)}(\boldsymbol{X}) + c\;\mathrm{SA}_\ell(\boldsymbol{H}^{(\ell)}(\boldsymbol{X})) \\
&= c\,\boldsymbol{H}^{(\ell+1)}(\boldsymbol{X}).
\end{aligned}$$

Thus residual addition preserves positive homogeneity.

Finally, suppose the final logits satisfy $\boldsymbol{z}(c\boldsymbol{X}) = c\,\boldsymbol{z}(\boldsymbol{X})$ with $c > 0$. Then, we have $\arg\max_k z_k(c\boldsymbol{X}) = \arg\max_k c\,z_k(\boldsymbol{X})$. Multiplication by a positive scalar preserves the ordering of the coordinates, thus, $\arg\max_k c\,z_k(\boldsymbol{X}) = \arg\max_k z_k(\boldsymbol{X})$. Therefore the predicted class is invariant to the scale factor c. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims presented in the abstract and introduction are justified by our experimental analyses and the associated explanations presented in Sections 3–6. In addition, we complement the corresponding explanations where appropriate through additional information and results Appendices B, C, and D.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the main limitations of our work in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result, we describe the corresponding assumptions in the paper and provide proofs in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our description of the considered Vanilla Transformer in Section 5 and of the experiments and results in Sections 3–6, together with additional details presented in Appendix B, covers all information for the reproduction of the results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide detailed descriptions of our method's implementation and hyperparameters in Section B. We will publicly release the code of our implementation upon acceptance of the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details needed to understand the results throughout Sections 3–6, together with Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The numerical results reported in Sections 3–6, as well as Appendix C are presented as the mean and standard deviation of multiple runs with different random seeds in order to provide information about their statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details about the compute resources used in our experiments in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper fully conforms with the criteria specified in the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: Our work focuses on an empirical analysis of the behavior and learned structure of Transformer models, The paper has no foreseeable societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: When considering code, data, or models defined by others, we make sure to properly cite the original work throughout the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Upon acceptance, we plan to publicly release the code corresponding to our implementation of the proposed approach and the reported experiments, which is documented in detail.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our research does not involve the usage of LLMs as an important or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.