Harmformer: Harmonic Networks Meet Transformers for Continuous Roto-Translation Equivariance

Anonymous Author(s)
Affiliation
Address
email

Abstract

Convolutional Neural Networks exhibit inherent equivariance to image translation, leading to efficient parameter and data usage, faster learning, and improved robustness. The concept of translation equivariant networks has been successfully extended to rotation transformation using group convolution for discrete rotation groups and harmonic functions for the continuous rotation group encompassing 360°. We explore the compatibility of the Self-Attention mechanism with full rotation equivariance, in contrast to previous studies that focused on discrete rotation. We introduce the Harmformer, a harmonic transformer with a convolutional stem that achieves equivariance for both translation and continuous rotation. Accompanied by an end-to-end equivariance proof, the Harmformer not only outperforms previous equivariant transformers, but also demonstrates inherent stability under any continuous rotation, even without seeing rotated samples during training.

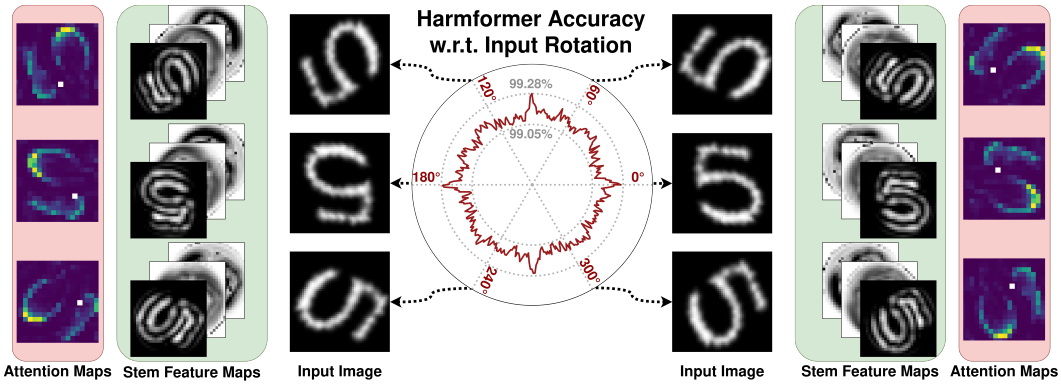

Figure 1: Equivariance of the Harmformer feature and attention maps in response to rotation of the input image: While the maps themselves are rotated, the magnitudes in the maps remain the same.

1 Introduction

A key strength that positions Convolutional Neural Networks (CNNs) [1] as a superior architecture for computer vision tasks is the weight sharing across the spatial domain. This design ensures that CNN feature maps retain their values as the input is translated, only being shifted according to the input. Formally known as translation equivariance, this property provides CNNs with inherent robustness and efficiency in managing translations. Equivariance can be extended to other groups of transformations, such as rotation, scaling, or mirroring. The advantage of equivariant models is

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

that they ensure a tractable response of the model to the transformation of the input. As a result, the model can eliminate the effects of the transformations and produce predictions that are invariant to them. For instance, to achieve translation invariance in conventional CNNs, the feature maps are commonly aggregated by global average pooling before the classification layer.

Group Equivariant Convolutional Neural Networks (G-CNNs) [2] show that CNNs can be modified to become equivariant to any discrete transformation group, such as rotation by a discrete set of angles. An extension to continuous rotation and translation group was introduced by Worrall et al. [3]. The authors proposed Harmonic Networks (H-Nets), which restrict the convolution filters to a family of harmonic functions ideal for expressing full rotation equivariance. Both approaches improve the generalization and efficiency of training for the chosen group, similarly to how CNNs benefit from translation equivariance. For example, rotation equivariant networks are well suited for object detection in aerial imagery because such images lack natural orientation and equivariant networks inherently accommodate all rotations. Beyond aerial imagery [4, 5], equivariant CNNs are effective in many other applications, such as microscopy [6, 7], histology [8], and remote sensing [9].

With the adoption of transformer architectures in computer vision, the Self-Attention mechanism has also been integrated into equivariant networks [10, 11, 12]. Equivariant transformers are gaining importance especially in domains such as graph-based structures (e.g. molecules) [13, 14, 15, 16], vector fields [17], manifolds [18], and generic geometric data [19, 20]. In the 2D domain, Romero and Cordonnier [21] proposed a transformer equivariant to discrete rotation and translation groups by using the principle of G-CNNs in the positional encoding of the Self-Attention (SA). The formulation was further improved by Xu et al. [22]. In both cases, the computational complexity of the equivariant SA increases quadratically with the number of angles in the considered rotation group, which limits the model angular resolution. Equivariance to continuous rotation presents a versatile solution.

In this paper, we introduce Harmformer, the first vision transformer capable of achieving continuous 2D roto-translation equivariance. The name is derived from circular harmonics [23] which provide the equivariance property preserved throughout the architecture. To ensure computational efficiency, our network starts with an equivariant convolutional stem based on Harmonic networks [3], where we redesign the key components, such as activations, normalization layers, and introduce equivariant residual connections. The stem output is divided into equivariant patches, which are then passed to the transformer. Alongside a novel self-attention SA mechanism, we introduce layer normalization and linear layers to guarantee end-to-end equivariance. The equivariance property allows Harmformer to remove the effect of roto-translation just before classification, preserving all relevant information at earlier stages (see Fig. 1).

Through experimental validation, we show that Harmformer surpasses all previous discrete equivariant transformers [21, 22] on established benchmarks [24, 25, 26]. It also outperforms earlier invariant models [27, 28, 29] on classification tasks where the model is trained solely on non-rotated data.

2 Related Work

We review the three foundational concepts from prior research that Harmformer builds upon: the SA mechanism, equivariant convolution networks, and transformers with a convolutional stem stage. Additionally, we discuss other equivariant transformer architectures.

Visual Self-Attention The well-known SA mechanism originates from natural language processing [30] and is widely used in computer vision since the publication of the Visual Transformer (ViT) [31]. Transformers, unlike CNNs, exhibit larger model capacities but require substantial amounts of data and have quadratic complexity with respect to input size. Transformers closely related to Harmformer include CoAtNet [32] and, more specifically, ViT$_p$ [33]. These architectures begin with a convolution stem to downscale the input and thereby reduce the computational complexity of the subsequent application of SA. However, these architectures are not equivariant to roto-translation.

Equivariant Convolutions Since the publishing of the G-CNNs [2], the concept of equivariant convolutional networks has expanded across various modalities and transformation groups. In 2D, these transformations include rotation [3, 34], scaling [35, 36, 37], and general $E(2)$ transformations [38]. In 3D, applications cover $SO(3)$ transformations in volumetric data [39, 40] and point clouds [41], as well as spherical CNNs [42]. Equivariant networks are also applied to graphs [43] and non-Euclidean manifolds [44]. Harmformer builds on and extends the H-Nets published by Worrall

et al. [3], which are purely convolutional networks equivariant to continuous 360° rotation. In our implementation of H-Nets, we incorporate the improvements introduced in H-NeXt [27].

Equivariant Transformers As previously mentioned, equivariant networks have integrated the SA mechanism in various domains, including 3D graphs and point clouds using irreducible representations [10, 15, 14], operations on Lie algebras [12], and general geometric data using geometric algebras [19, 20]. Particularly relevant to our work are the planar roto-translation equivariant transformers, such as Group Equivariant Self-Attention Networks (GSA-Nets) [21] and $E(2)$-Equivariant Vision Transformer (GE-ViT), which reformulate relative positional encoding to construct equivariant transformers. However, GSA-Nets and GE-ViT operate only on discrete rotation groups such as $\left\{k\frac{\pi}{2} \mid k \in \mathbb{Z}\right\}$, where finer angular sampling substantially increases the computational complexity.

3 On Equivariance in Vision Transformers

We analyze the roto-translation equivariance of the ViT architecture, a well-known representative of vision transformers. First, we formalize the notion of equivariance. Intuitively, a function f is equivariant to a transformation a_g if the transformation and the function commute, $f(a_g(x)) = a_g(f(x))$. For example, processing a rotated input image has the same effect as directly rotating the features of the unrotated image. In practice, such a definition would be too restrictive. The function f (layer or network) typically has a different domain and codomain, so the transformation may act differently on each. The core idea remains the same: the model response to the input transformation is predictable. To formally define equivariance, we draw upon the seminal work of Cohen and Welling [2] or the more recent one formulated by Weiler et al. [45].

Definition 3.1 (Equivariance)**.** A function $f : X \to Y$ (a whole network or a single layer) is called group equivariant with respect to a group G if for every element g in G, represented by a linear map $a_g : X \to X$, there exists a corresponding linear map $b_g : Y \to Y$ such that the following holds:

$$f(a_g(x)) = b_g(f(x)) \quad \text{for all } x \in X \text{ and } g \in G. \tag{1}$$

A composition $f_2(f_1(x))$ of two equivariant functions $f_1 : X \to Y$ and $f_2 : Y \to Z$ is equivariant. We call invariance a special case of equivariance when b_g is the identity for all g in G.

Self-Attention A key mechanism that distinguishes transformers from previous architectures is the SA layer [30]. Before discussing the properties of SA, let us formally define it.

Definition 3.2 (Self-Attention)**.** Given an input matrix $Y \in \mathbb{R}^{n\times d}$, where each row of Y represents a feature vector of dimension d, usually called a patch. The matrices Q (queries), K (keys), and V (values) are computed as linear projections of Y:

$$[Q, K, V] = [YW_q, YW_k, YW_v] \qquad W_{q,k,v} \in \mathbb{R}^{d\times d_h}, \tag{2}$$

where d_h is the dimension within the SA layer. The output of the self-attention layer, $\mathrm{SA}(Y)$, is a weighted sum of the vectors in V, where the weights are defined as the softmax-normalized pairwise similarity scores between the vectors in Q and K:

$$A = \mathrm{softmax}(QK^T/\sqrt{d_h}) \qquad A \in \mathbb{R}^{n\times n}, \tag{3}$$

$$\mathrm{SA}(Y) = AV. \tag{4}$$

In practice, SA is typically extended to Multi-Head Self-Attention (MSA), in which multiple SA layers with different embedding matrices $W_{(k,v,q)}$ are computed in parallel and then combined.

The construction of SA implies its well-known property, in the literature often referred to as permutation invariance [46]. According to Def. 3.2, it is more accurate to call the SA layer permutation equivariant rather than permutation invariant. For if we change the order of the rows in Y, the $\mathrm{SA}(Y)$ remains the same except for the same change in the order of its output rows.

What makes ViT non-equivariant? As rotation and translation are special cases of permutation, the permutation equivariance of SA might suggest the roto-translation equivariance of the whole ViT. However, the permutation-equivariance of SA holds at the patch level and not at the pixel level, where translation or rotation takes place. In the initial stage of ViT, before the first SA layer, the image is split into n non-overlapping patches of fixed size, typically 16×16 pixels. These are linearly transformed and flattened to form the rows of the input matrix Y of the first SA layer. This patch-wise

operation breaks the direct rotation (or translation) equivariance of ViT at the image level, because for an image rotation a_g in Eq. (1) corresponding to an angle g from the rotation group G, there is no b acting on the patches that can be expressed as a rotation b_g by g; the same holds for translation.

A solution typically used by previous equivariant approaches [21, 22] is to consider pixel-level "patches" of size 1×1 pixel. Then the image rotation is equivalent to patch-level permutation, and the corresponding transformer model remains equivariant, assuming that interpolation errors and boundary effects are minimal. This approach, however, has two major drawbacks. As seen from Eq. (3), the SA has a quadratic complexity with respect to the number of patches n, so operating on a pixel grid (as opposed to 16×16) incurs an almost 10^5 penalty factor in memory requirements and correspondingly increases the processing time. To mitigate this, GSA-Nets and GE-ViT reduce complexity by using local SA [47] that restricts the attention field to the 7×7 neighborhood of the patch. The second drawback is that the local self-attention in the first layers is not very informative, because nearby pixels are usually highly correlated.

Position Encoding During construction of the input matrix Y for the first SA layer, the patches are also given absolute position encoding that provides information about their locations. This breaks equivariance as the patches of a transformed image will receive different encoding compared to their counterparts in the original image. Equivariant transformers [21, 22] replace the absolute encoding with circular relative encoding, similar to iRPE introduced by Wu et al. [48].

In Harmformer, we address these challenges with a convolutional stem stage that initially reduces spatial dimensions and extracts high-level features. Subsequently, we create 1×1 patches from these high-level features and process them by the SA layers. To maintain spatial correspondence among the patches while ensuring equivariance, Harmformer also uses circular relative position encoding.

4 Harmonic Convolutions and Equivariance to Continuous Roto-Translation

To understand the equivariance property of Harmformer, it is essential to understand the concept of harmonic convolutions introduced in H-Nets [3], as they are employed in the stem and affect the subsequent transformer layers. The main difference from the traditional CNNs is that the convolution filters based on circular harmonic functions are specifically designed to encode rotational symmetries. The filters are defined as follows.

Definition 4.1 (Harmonic Filter)**.** A harmonic filter $W_m : \mathbb{R}^2 \to \mathbb{C}$ parameterized by a rotation order m is given by:

$$W_m(r, \theta) = R(r)e^{i(m\theta+\beta)}, \tag{5}$$

where (r, θ) are polar coordinates. Here, $R : \mathbb{R} \to \mathbb{R}$ is a learnable radial function and $\beta \in \mathbb{R}$ is a learnable phase shift. The rotation order m is a parameter that determines the filter symmetry.

As translation equivariance is inherently provided by convolution, we will focus solely on rotation in the following discussion and denote the rotation operator by $[\cdot]^\alpha$, where α is the angle of rotation. Let us look in detail at how the rotation applied to the input affects feature maps generated by harmonic convolution. The H-Nets features are represented as complex values in polar form.

Lemma 4.1 (Harmonic Convolution Property)**.** Let I be an input image and W_{m_1} a harmonic filter. Under image rotation by angle α, convolution of I with W_{m_1} is given by:

$$[I]^\alpha \circledast W_{m_1} = e^{im_1\alpha} \, [I \circledast W_{m_1}]^\alpha \ . \tag{6}$$

This equation shows that rotating the image only results in a phase shift of the feature values, while the spatial coordinates are rotated accordingly. This property also holds for subsequent convolution layers. If the first feature map is denoted as $F_{m_1}([I]^\alpha)$, then convolution with another harmonic filter W_{m_2} is given by:

$$F_{m_1}([I]^\alpha) \circledast W_{m_2} = e^{i(m_1+m_2)\alpha} \, [F_{m_1}(I) \circledast W_{m_2}]^\alpha \ . \tag{7}$$

The authors of H-Nets also construct activation, batch normalization, and pooling layers that preserve this property. As a result, their classifier can be independent of input rotation and translation. To remove the influence of rotation, they extract only the magnitude from the last feature map and discard the phase. To aggregate spatial information, they use global average pooling. Note that for

tasks where the orientation of the object is relevant, phase can be used as no information is lost due to equivariance.

To unify the equivariance property within the Harmformer architecture, we define Harmonic Equivariance (HE), which is motivated by Lemma 4.1 and satisfies the general definition of equivariance (Def. 3.1). HE describes how features transform with respect to the rotation of an input image. By showing that each Harmformer layer satisfies HE, we establish the relationship between the features and the rotation of an input throughout the model.

Definition 4.2 (Harmonic Equivariance – HE)**.** A layer $F_m(\cdot)$ associated with a rotation order m is said to be HE, if for any rotation by angle α and admissible input I, it is transformed as follows:

$$F_m([I]^\alpha) = e^{im\alpha}\,[F_m(I)]^\alpha\,. \tag{8}$$

Here $[F_m(I)]^\alpha$ are features obtained from an unrotated input I and then rotated. The phase is shifted by a multiple of the rotation angle, where the factor is given by the rotation order of the layer. The process is illustrated in Fig. 3a.

5 Harmformer Architecture

The architecture of Harmformer is shown in Figure 2 and its layers will be discussed one by one. HE (Def. 4.2) of each layer is proved in Appendix A, demonstrating the end-to-end continuous rotation and translation equivariance. The architecture begins with a stem stage based on H-Nets, which we have further improved by refining activation and normalization layers and incorporating residual connections. The stem is followed by an equivariant encoder tailored to maintain HE, and the last component is a classifier, which takes the HE output of the encoder and computes an invariant representation for classification.

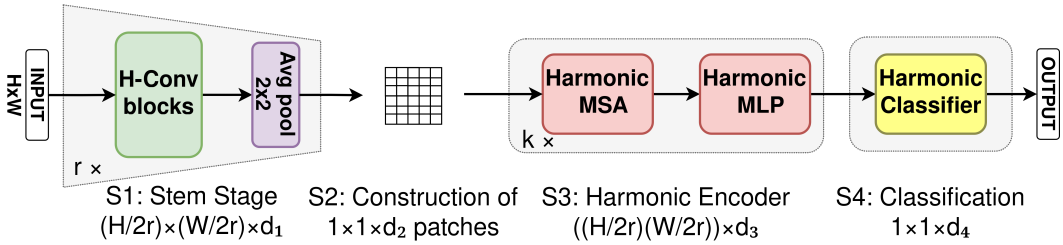

Figure 2: Overview of the Harmformer architecture, divided into four stages: S1 - downscaling the input, S2 - constructing patches from feature maps, S3 - Harmonic Encoder, and S4 - Classifier.

5.1 Harmformer: S1 Stem Stage

The main role is to prepare features for the Harmonic Encoder (S3) so that they are HE and have lower spatial resolution to keep the computational complexity of SA manageable, as discussed in Sec. 3. To this end, we design the stage to comprise r iterations of H-Conv blocks, followed by average pooling, as shown in Fig. 2. Each iteration increases the number of channels while decreasing the spatial dimension.

The stage starts with an input that formally satisfies HE for the rotation order $m = 0$, expressed as $[I]^\alpha = e^{i0\alpha}\,[I]^\alpha$, followed by the first H-Conv block shown in Figure 3b.

Rotation Order Streams The HE and the definition of Harmonic Convolution have already been detailed in Lemma 4.1 and Def. 4.1. An important aspect that remains to be addressed is the selection of rotation orders for the harmonic filters. In our initial convolution with the input image (often called lifting convolution), we employ harmonic filters of rotation orders -1, 0, and 1. This setup produces three streams of feature maps, each corresponding to one of these rotation orders.

Our experiments, along with the results reported in [27, 3], indicate that generating feature maps of higher rotation orders does not significantly improve performance but increases the computational complexity. Based on this evidence, we limit rotation orders to -1, 0, and 1.

Most layers process these streams independently and those that interact across streams are indicated in the diagrams by a "spoon" symbol, as in the case of Harmonic Convolution in Figure 3b. Streams

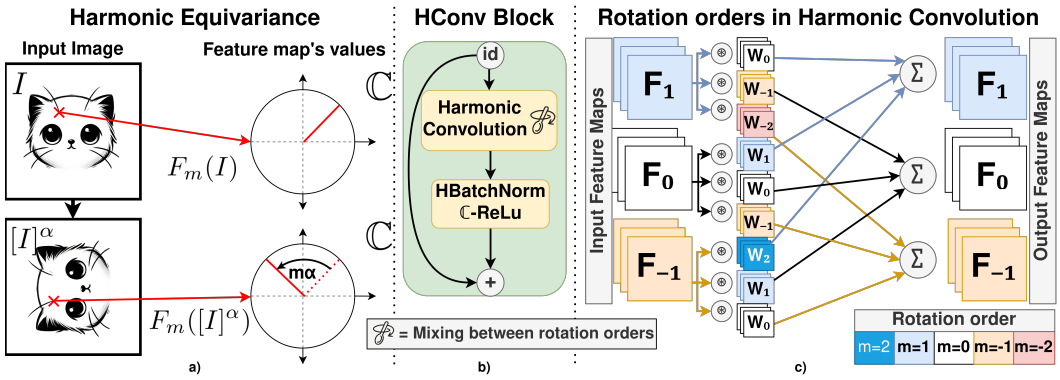

Figure 3: (a) Phase shift of HE feature values when the input is rotated; (b) Harmonic Convolution (H-Conv) Block of the stem stage; (c) Interaction of harmonic filters W_m with feature maps F_m within the Harmonic Convolution layer of the H-Conv Block, where m is the rotation order.

in the Harmonic Convolution block are mixed similarly as in H-Nets. The proposed mixing strategy is shown in Figure 3c and follows from the Harmonic Convolution property in (7), which states that

$$F_m^{out} = \sum_{m=m_1+m_2} F_{m_1}^{in} \circledast W_{m_2}, \tag{9}$$

where m, m_1, and m_2 are the rotation orders of the output, input, and harmonic filter, respectively.

Layers Operating on Magnitude Because rotation affects only the phase of the features leaving the magnitude untouched, element-wise functions, such as normalization or activation, operating only on magnitudes preserve the HE property. In contrast with previous H-Nets [27, 3], we restrict the codomain of every element-wise function f transforming magnitudes to non-negative numbers, $f : \mathbb{R} \to \mathbb{R}_0^+$, since negative magnitudes inadvertently flip the phase, thus violating the HE property. This consideration leads us to propose a novel normalization fused together with activation (HBatchNorm and $\mathbb{C}$-ReLu), detailed in Appendix A.4. Restricting the codomain and fusing the normalization with the activation has a positive impact on performance, as shown in Ablation B.1.

Residual Connection The final stem element is the residual connection, previously unused in H-Nets. Residual connections are also used within our encoder blocks. As in standard CNNs, they improve gradient flow and reduce training time. With respect to rotation orders, they process streams independently, thus preserving HE according to the following lemma:

Lemma 5.1 (HE of Residual Connections)**.** A residual connection between feature maps of the same rotation order, $F_m(I)$ and $F'_m(I)$, preserves HE property:

$$F'_m([I]^\alpha) + F_m([I]^\alpha) = e^{im\alpha} \left[(F'_m(I) + F_m(I))\right]^\alpha . \tag{10}$$

5.2 Harmformer: S2 Construction of the Patches

To integrate the stem output with the encoder, the final stem feature maps are divided into 1×1-sized patches, as illustrated in Figure 4a. The patches are constructed separately for all three streams of rotation orders. The resulting stack of patches then comprises three matrices $F_{-1}, F_0, F_1 \in \mathbb{C}^{(h\cdot w)\times d}$, each representing a single rotation order, where h, w, and d denote the height, width, and number of channels of the last feature maps, respectively. We keep this notation for encoder feature maps (patches), as they correspond to the stem feature maps, just reshuffled.

Neglecting small interpolation errors, the spatial transformation of the input translates only into a permutation of the stack of patches F_m as discussed in Sec. 3. For clarity, we use a discrete representation but it should be noted that the encoder can be modeled using a functional framework, as shown by Romero and Cordonnier [21].

Before feeding the SA with patches, transformer networks typically apply a linear projection to adjust the dimension d. We use a linear layer that processes the patches independently with respect to their order of rotation to preserve HE:

Lemma 5.2 (HE of Linear Layer)**.** A linear layer applied to a HE feature map $F_m([I]^\alpha) \in \mathbb{C}^{(hw)\times d_{in}}$ preserves the rotation order m. Formally, we have:

$$F_m([I]^\alpha)W = e^{mi\alpha}\,[F_m(I)W]^\alpha\,, \tag{11}$$

where $W \in \mathbb{C}^{d_{in}\times d_{out}}$ represents a shared weight matrix applied independently over all spatial positions of the input feature map.

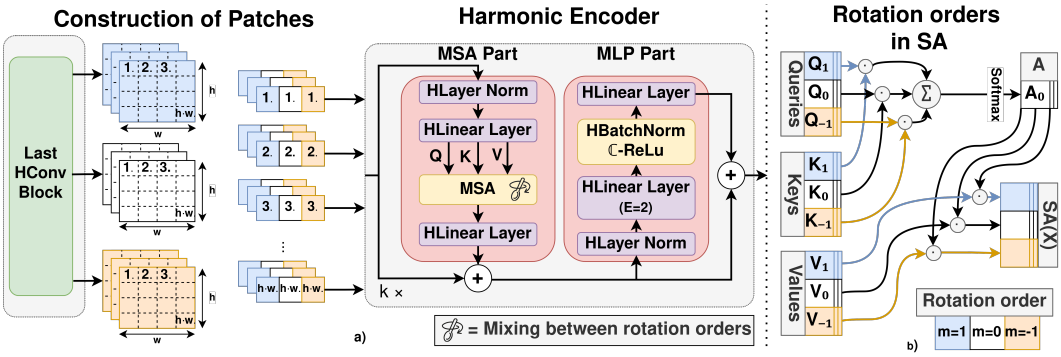

Figure 4: a) Construction of the patches (colors represent rotation orders) and the Harmonic Encoder structure. b) Diagram depicting the interaction of SA mechanisms across different rotation orders.

5.3 Harmformer: S3 Harmonic Encoder

This section outlines our encoder, which is designed to preserve the HE property. The encoder is organized into several k blocks, each containing Multi-Head Self-Attention (MSA) and Multi-Layer Perceptron (MLP) components, as shown in Figure4a. Along with the layers presented in the previous sections, we propose a SA mechanism and a layer normalization, both of which preserve the HE.

As the following lemma shows, the layer normalization can be adapted to satisfy HE by operating independently on the streams of rotation orders.

Lemma 5.3 (HE of Layer Norm)**.** A feature map $F_m([I]^\alpha) \in \mathbb{C}^{(hw)\times d}$ with a rotation order m preserves HE when normalized by its mean and standard deviation:

$$\frac{F_m([I]^\alpha) - \mu}{\sigma + \epsilon} = e^{im\alpha}\frac{[F_m(I)]^\alpha - \mu}{\sigma + \epsilon}, \tag{12}$$

where μ, σ are the sample means and standard deviations of the original feature maps computed over their spatial dimensions, respectively, and ϵ is a small constant added for numerical stability.

Self-Attention The essential components of the encoder are MSA layers. The proposed MSA mixes features with different rotation orders. In the first step, queries, keys and values are generated for each rotation order -1, 0, and 1 independently, which preserves HE as follows from Lemma 5.2. We split the SA calculation (Eq. (3),(4)) into two operations: dot product and matrix multiplication, and demonstrate their properties by the following lemmas.

Lemma 5.4 (Dot product subtracts rotation orders)**.** Consider two HE feature maps $Q_{m_1}([I]^\alpha) \in \mathbb{C}^{(hw)\times d}$ and $K_{m_2}([I]^\alpha) \in \mathbb{C}^{(hw)\times d}$ that represent queries and keys, respectively. The dot product of these feature maps is HE and has the rotation order $m_1 - m_2$. Formally, we have:

$$Q_{m_1}([I]^\alpha)\overline{K_{m_2}([I]^\alpha)^T} = e^{i(m_1-m_2)\alpha}\left[Q_{m_1}(I)\overline{K_{m_2}(I)^T}\right]^\alpha, \tag{13}$$

where $\overline{K_{m_2}([I]^\alpha)^T}$ denotes the complex conjugate transpose of $K_{m_2}([I]^\alpha)$.

Lemma 5.5 (Matrix multiplication sums rotation orders)**.** Consider a HE feature map $A_{m_1}([I]^\alpha) \in \mathbb{C}^{(hw)\times(hw)}$ representing an attention matrix and HE feature map $V_{m_2}([I]^\alpha) \in \mathbb{C}^{(hw)\times d}$ representing values. The result of their matrix multiplication is HE with a rotation order $m = m_1 + m_2$:

$$A_{m_1}([I]^\alpha)V_{m_2}([I]^\alpha) = e^{i(m_1+m_2)\alpha}\,[A_{m_1}(I)V_{m_2}(I)]^\alpha\,. \tag{14}$$

where $[A_{m_1}(I)]^\alpha$ and $[V_{m_2}(I)]^\alpha$ are feature maps created from unrotated I and rotated afterwards.

After these operations, the relative circular encodings [48] are added to the result of the dot product before it undergoes the softmax activation. The Harmformer softmax operates only on magnitudes and its codomain is within $\mathbb{R}_0^+$ to avoid breaking HE.

We have shown that the dot product between queries Q_{m_1} and keys K_{m_2} results in a rotation order of $m = m_1 - m_2$. Similarly, matrix multiplication between the attention matrix A_{m_1} and values V_{m_2} yields a feature map with a rotation order of $m = m_1 + m_2$. The final task is to combine these rotation orders to produce output feature maps with the same number of rotation orders as the input feature maps, i.e. -1, 0, and 1.

Mixing Orders in MSA Since there are multiple strategies for combining rotation orders, we explored several of them and provide details on other configurations in Ablation B.2. The optimal approach, according to our experiments, is shown in Figure 4 and involves:

1. **Dot Product Calculation:** The dot product is computed only between the same rotation orders, separately. According to Lemma 5.4, this results in three feature maps with the rotation order 0 and dimension $\mathbb{C}^{(hw)\times(hw)}$.
2. **Attention Matrix Formation:** These results are summed to form a single matrix of rotation order 0. A softmax function is then applied to form a single attention matrix $A_0 \in \mathbb{C}^{(hw)\times(hw)}$, preserving the rotation order 0, because softmax function operates only on magnitudes and outputs non-negative numbers.
3. **Self-Attention Output:** Finally, the self-attention output is produced by matrix multiplication of the attention matrix (rotation order zero) with the values of each rotation order. This process results in a triplet of outputs with the target rotation orders $(-1, 0, 1)$.

Other layers, such as linear layers and residual connections, have been introduced in previous sections. By stacking HE layers, the last feature map coming from our encoder will maintain the HE property.

5.4 Harmformer: S4 Classification

Spatial position and orientation are generally redundant for classification tasks, except when classifying directional objects such as arrows. In the final stage, we remove this redundant information from the feature maps and produce an invariant feature vector. The feature maps entering the classification stage form a matrix of the shape $\mathbb{C}^{3\times n\times d}$. To aggregate over different rotation orders, we keep only the magnitude, resulting in $\mathbb{R}^{n\times 3d}$. The spatial information is then eliminated by applying global average pooling over the dimension n (patches), reducing the shape to $\mathbb{R}^{3d}$. The final feature vector, which is roto-translation invariant, is processed by a single linear layer for classification.

6 Experiments

To validate the properties of Harmformer, we conducted experiments on four benchmarks listed in Table 1. For detailed experimental configurations and an ablation study of architectural modifications, see Appendices C and B, respectively. Addition segmentation experiment is included in Appendix D.

Table 1: Dataset overview, detailing sizes and whether training and test sets contain rotated images.

Dataset Name	Sample Size	Train/Test/Val. Size	Rot. Train/Test	Ref.	Scenario
mnist-rot-test	$28 \times 28 \times 1$	50k / 10k / 10k	✗/✓	[27]	1
cifar-rot-test	$32 \times 32 \times 3$	42k / 10k / 8k	✗/✓	[27]	1
rotated MNIST	$28 \times 28 \times 1$	10k / 2k / 50k	✓/✓	[24]	2
PCam	$96 \times 96 \times 3$	262k / 32k / 32k	✗/✗	[25, 26]	2

Model Architecture The models are designed to match the number of parameters of the previous state-of-the-art models while maintaining the same overall architecture. Depending on the benchmark, the stem stage consists of 2-4 blocks to reduce resolution, followed by 3-4 harmonic encoder blocks. To ensure that the equivariant properties emerge from the architecture, we avoid any data augmentation. Consistent with H-NeXt [27], the inputs are initially upscaled by a factor of two to mitigate interpolation errors.

Table 2: Error on mnist-rot-test

Model	Error	Param.
ResNets-50 [28]	57.6%	
SWN-GCN [28]	8.20%	2.7M
H-Nets [3]	7.11%	33k
H-NeXt [27]	1.30%	28k
Harmformer	**0.82%**	29.7k

Table 3: Error on cifar10-rot-test

Model	Error	Param.
ResNets-50 [28]	63.90%	
SWN-GCN [28]	49.50%	2.7M
H-NeXt[27]	38.54%	118k
Harmformer	31.41%	118k
Harmformer (Large)	**29.29%**	217k

Table 4: Error on rotated MNIST

SA model	Error	Param.
GSA-Nets[21]	2.03%	44.7k
GE-ViT[22]	1.99%	45k
Harmformer	**1.18%**	30k
CNNs model		
G-CNNs[2]	1.69%	73.1k

Table 5: PCam

Error	Param.
15.24%	206k
16.18%	
12.47%	146k
10.88%	141k

Table 6: Average Error

Dataset	Avg %±Std
rotMNIST	1.26±0.055
PCam	14.2±0.986
mnist-rot-test	0.91±0.142
cifar-rot-test	31.9±0.636
cifar-rot-test (Large)	29.7±0.203

Invariance Benchmarks In the first scenario, we verify the equivariance of Harmformer by training the model exclusively on upright (non-rotated) data and testing it on randomly rotated data; the first two datasets in Tab. 1. Since the model is trained only on upright images, any equivariant properties must arise purely from the model design, not from the training data.

We outperform previous methods on both datasets, as shown in Tables 2 and 3. Harmformer improves the robustness to rotation, as we discuss further in Section B.4, which partially explains the performance gain on mnist-rot-test. Stability under rotation is less enhanced on cifar-rot-test (Sec. B.4), and the superior results there are likely due to the higher model capacity of the transformer architecture, which improves the overall detection performance.

Equivariance Benchmarks In the second scenario, we compare the performance of the Harmformer on established equivariance benchmarks for roto-translation where there is no significant distribution shift between the training and test sets, either containing rotated samples or not. This evaluation assesses how our method stacks up against previous equivariant transformers that are equivariant to discrete rotation and translation. Tables 4 and 5 show the top results on the rotated MNIST and PCam datasets, respectively. Harmformer outperforms previous equivariant transformers and narrows the performance gap between equivariant transformers and convolution-based models.

For completeness, we include the average performance in each benchmark listed in Table 6. The accuracy on the PCam dataset was slightly unstable, probably due to the characteristics of the dataset. Methods specifically designed for PCam, such as [8], use extensive augmentation techniques, which were avoided in our case to ensure unbiased results.

7 Conclusion and Future Work

The proposed Harmformer is the first transformer model to achieve end-to-end equivariance to continuous rotation and translation in 2D. This was accomplished by designing an equivariant self-attention inspired by harmonic convolution. Along with the novel SA, we introduced several layers specifically tailored for equivariance, including linear layers, layer normalization, batch normalization, activations, and residual connections. Our model outperforms previous equivariant transformers, narrowing the performance gap with convolution-based equivariant networks.

We hypothesize that the full potential of transformers may not be realized due to the nature of traditional benchmarks. The 2D equivariant transformers have so far been tested on datasets containing relatively small images that lack global dependencies. Therefore, future research should explore the application of equivariant transformers on larger datasets where, similar to ViT, they could demonstrate their potential. In addition, the proposed model can be extended to other modalities while maintaining its equivariance properties. For example, the harmonic networks that form the basis of our approach can also be adapted for 3D applications.

References

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://ieeexplore.ieee.org/abstract/document/726791.

[2] Taco Cohen and Max Welling. Group Equivariant Convolutional Networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2990–2999. PMLR, 2016. doi: 10.48550/arXiv.1602.07576. URL https://arxiv.org/pdf/1602.07576.

[3] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic Networks: Deep Translation and Rotation Equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5028–5037, 2017. doi: 10.48550/arXiv.1612.04642. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Worrall_Harmonic_Networks_Deep_CVPR_2017_paper.html.

[4] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2786–2795, June 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Han_ReDet_A_Rotation-Equivariant_Detector_for_Aerial_Object_Detection_CVPR_2021_paper.html.

[5] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning RoI transformer for oriented object detection in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Ding_Learning_RoI_Transformer_for_Oriented_Object_Detection_in_Aerial_Images_CVPR_2019_paper.html.

[6] Benjamin Chidester, Tianming Zhou, Minh N Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz353. URL https://doi.org/10.1093/bioinformatics/btz353.

[7] Benjamin Chidester, That-Vinh Ton, Minh-Triet Tran, Jian Ma, and Minh N. Do. Enhanced rotation-equivariant U-Net for nuclear segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. URL https://openaccess.thecvf.com/content_CVPRW_2019/html/CVMI/Chidester_Enhanced_Rotation-Equivariant_U-Net_for_Nuclear_Segmentation_CVPRW_2019_paper.html.

[8] Simon Graham, David Epstein, and Nasir Rajpoot. Dense steerable filter cnns for exploiting rotational symmetry in histology images. *IEEE Transactions on Medical Imaging*, 39(12):4124–4136, 2020. doi: 10.1109/TMI.2020.3013246. URL https://ieeexplore.ieee.org/abstract/document/9153847.

[9] Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2019. doi: 10.1109/TIP.2018.2867198. URL https://ieeexplore.ieee.org/abstract/document/8445665/.

[10] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1970–1981. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/15231a7ce4ba789d13b722cc5c955834-Paper.pdf.

[11] David W. Romero and Mark Hoogendoorn. Co-attentive equivariant neural networks: Focusing equivariance on transformations co-occurring in data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1g6ogrtDr.

[12] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4533–4543. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hutchinson21a.html.

[13] Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 6(4):417–427, Apr 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00815-9. URL https://doi.org/10.1038/s42256-024-00815-9.

[14] 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*, 2024. URL https://openreview.net/forum?id=mCOBKZmrzD.

[15] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

[16] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB.

[17] Serge Assaad, Carlton Downey, Rami Al-Rfou', Nigamaa Nayakanti, and Benjamin Sapp. VN-transformer: Rotation-equivariant attention for vector neurons. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EiX2L4sDPG.

[18] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27331–27343. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf.

[19] Pim de Haan, Taco Cohen, and Johann Brehmer. Euclidean, projective, conformal: Choosing a geometric algebra for equivariant transformers. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3088–3096. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/haan24a.html.

[20] Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco S Cohen. Geometric algebra transformer. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35472–35496. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6f6dd92b03ff9be7468a6104611c9187-Paper-Conference.pdf.

[21] David W. Romero and Jean-Baptiste Cordonnier. Group Equivariant Stand-Alone Self-Attention For Vision. In *International Conference on Learning Representations*, 2021. doi: 10.48550/arXiv.2010.00977. URL https://openreview.net/forum?id=JkfYjnOEo6M.

[22] Renjun Xu, Kaifan Yang, Ke Liu, and Fengxiang He. $E(2)$-Equivariant Vision Transformer. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2356–2366. PMLR, 31 Jul–04 Aug 2023. doi: 10.48550/arXiv.2306.06722. URL https://proceedings.mlr.press/v216/xu23b.html.

[23] William T Freeman et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[24] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 473–480. Association for Computing Machinery, 2007. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273556. URL https://dl.acm.org/doi/abs/10.1145/1273496.1273556.

[25] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen A. W. M. van der Laak, , and the CAMELYON16 Consortium. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, 318(22):2199–2210, 12 2017. ISSN 0098-7484. doi: 10.1001/jama.2017.14585. URL https://doi.org/10.1001/jama.2017.14585.

[26] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II*, page 210–218, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-00933-5. doi: 10.1007/978-3-030-00934-2_24. URL https://doi.org/10.1007/978-3-030-00934-2_24.

[27] Tomáš Karella, Filip Šroubek, Jan Blažek, Jan Flusser, and Václav Košík. H-NeXt: The next step towards roto-translation invariant networks. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. URL https://papers.bmvc2023.org/0578.pdf.

[28] Sungwon Hwang, Hyungtae Lim, and Hyun Myung. Equivariance-bridged SO (2)-invariant representation learning using graph convolutional network. In *The 32nd British Machine Vision Conference (BMVC 2021)*. The British Machine Vision Association, 2021. doi: 10.48550/arXiv.2106.09996. URL https://www.bmvc2021-virtualconference.com/assets/papers/0218.pdf.

[29] Renata Khasanova and Pascal Frossard. Graph-based isometry invariant representation learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1847–1856. PMLR, 2017. doi: 10.48550/arXiv.1703.00356. URL http://proceedings.mlr.press/v70/khasanova17a.html.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[32] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3965–3977. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/20568692db622456cc42a2e853ca21f8-Paper.pdf.

[33] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollar, and Ross Girshick. Early convolutions help transformers see better. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30392–30400. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ff1418e8cc993fe8abcfe3ce2003e5c5-Paper.pdf.

[34] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Weiler_Learning_Steerable_Filters_CVPR_2018_paper.html.

[35] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f04cd7399b2b0128970efb6d20b5c551-Paper.pdf.

[36] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgpugrKPS.

[37] Md Ashiqur Rahman and Raymond A. Yeh. Truly scale-equivariant deep nets with fourier layers. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 6092–6104. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1343edb2739a61a6e20bd8764e814b50-Paper-Conference.pdf.

[38] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf.

[39] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D Steerable CNNs: Learning rotationally equivariant features in volumetric data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/488e4104520c6aab692863cc1dba45af-Paper.pdf.

[40] Daniel Worrall and Gabriel Brostow. CubeNet: Equivariance to 3D rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Daniel_Worrall_CubeNet_Equivariance_to_ECCV_2018_paper.html.

[41] Erik J Bekkers, Sharvaree Vadgama, Rob Hesselink, Putri A Van der Linden, and David W. Romero. Fast, expressive $\mathrm{SE}(n)$ equivariant networks through weight-sharing in position-orientation space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dPHLbUqGbr.

[42] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hkbd5xZRb.

[43] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9323–9332. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/satorras21a.html.

[44] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks – isometry and gauge equivariant convolutions on Riemannian manifolds, 2021.

[45] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. *Equivariant and Coordinate Independent Convolutional Networks*. 2023. URL https://maurice-weiler.gitlab.io/cnn_book/EquivariantAndCoordinateIndependentCNNs.pdf.

[46] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2023. doi: 10.1109/TNNLS.2022.3227717. URL https://ieeexplore.ieee.org/abstract/document/10088164.

[47] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3416a75f4cea9109507cacd8e2f2aefc-Paper.pdf.

[48] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10033–10041, 2021. doi: 10.48550/arXiv.2107.14222. URL https://openaccess.thecvf.com/content/ICCV2021/html/Wu_Rethinking_and_Improving_Relative_Position_Encoding_for_Vision_Transformer_ICCV_2021_paper.html.

[49] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

[50] J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004. doi: 10.1109/TMI.2004.825627.

[51] Erik J. Bekkers, Maxime W. Lafarge, Mitko Veta, Koen A. J. Eppenhof, Josien P. W. Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, page 440–448, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-00927-4. doi: 10.1007/978-3-030-00928-1_50. URL https://doi.org/10.1007/978-3-030-00928-1_50.

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[53] Wentao Liu, Huihua Yang, Tong Tian, Zhiwei Cao, Xipeng Pan, Weijin Xu, Yang Jin, and Feng Gao. Full-resolution network and dual-threshold iteration for retinal vessel and coronary angiograph segmentation. *IEEE Journal of Biomedical and Health Informatics*, 26(9):4623–4634, 2022. doi: 10.1109/JBHI.2022.3188710.

[54] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31, 2018.

A Proofs of Harmformer Equivariance

In this section, we systematically formulate the proofs of the HE property (Definition 4.2) for each layer of the Harmformer. The central concept of the architecture is the handling of three streams corresponding to different rotation orders. By oversimplifying the interactions of rotation orders, two main properties of the harmonic function should be highlighted:

1. Feature maps with the same rotation order can be summed:
$$e^{im\alpha}F_1 + e^{im\alpha}F_2 = e^{im\alpha}(F_1 + F_2)$$
2. Multiplication of feature maps results in the sum of their rotation orders:
$$e^{im_1\alpha}F_1 \cdot e^{im_2\alpha}F_2 = e^{i(m_1+m_2)\alpha}(F_1 \cdot F_2)$$

Interactions between these streams occur in layers harmonic convolution or Multi-Head Attention (MSA). Other layers, such as layer normalization, process feature maps of the different rotation order independently. Additionally, some operations, such as batch normalization and activation functions, operate solely on the magnitudes of complex numbers leaving the phase untouched.

A.1 Equivariance of Harmonic Convolutions

Note that the proof of H-Conv equivariance was originally formulated by Worrall et al. [3]. For the sake of completeness, we have provided a highly simplified version of these proofs. However, we encourage readers to read the more comprehensive work on G-steerable convolution kernels and the theory of steerable equivariant convolution networks in [45] (Chapters 4-5), which provides a broader perspective and demonstrates the equivalence with G-CNNs.

Lemma A.1 (Rotation of a Harmonic Filter)**.** When the coordinates of a harmonic filter are rotated by an angle α, it only changes by a factor $e^{im\alpha}$, where m is the rotation order of the harmonic filter and $\mathcal{R}_\alpha$ is a corresponding 2D rotation matrix.

Proof.

$$\begin{aligned} W_m(\mathcal{R}_\alpha^{-1}\mathbf{x}) \equiv \tilde{W}_m(r, \theta - \alpha) &= R(r) \cdot e^{-im(\theta-\alpha)} \\ &= e^{im\alpha}\tilde{W}_m(r,\theta) \equiv e^{im\alpha}W_m(\mathbf{x}), \end{aligned} \tag{15}$$

where x is the spatial coordinates. □

Let us denote an input image I that is rotated by the angle α and translated by vector $\mathbf{t}$ as

$$[I]_t^\alpha(\mathbf{x}) \equiv I(\mathcal{R}_\alpha^{-1}\mathbf{x} + \mathbf{t}). \tag{16}$$

Theorem A.1 (Harmonic convolution sums the rotation orders)**.** When an input image I is rotated by α and translated by $\mathbf{t}$, the output of a multiple successive harmonic convolution is given by:

$$[W_{m_1} \circledast W_{m_2} \circledast \cdots \circledast [I]_t^\alpha](\mathbf{x}) = e^{i(m_1+m_2+\cdots)\alpha}\,[W_{m_1} \circledast W_{m_2} \circledast \cdots \circledast I](\mathcal{R}_\alpha\mathbf{x} + \mathbf{t}) \tag{17}$$

Proof. We start with the very first harmonic convolution.

$$[W_{m_1} \circledast [I]_t^\alpha](\mathbf{x}) = \int_{\mathbb{R}^2} W_{m_1}(\mathbf{z})\,[I]_t^\alpha(\mathbf{x} - \mathbf{z})d\mathbf{z} \tag{18}$$

$$= \int_{\mathbb{R}^2} W_{m_1}(\mathbf{z})I(\mathcal{R}_\alpha(\mathbf{x} - \mathbf{z}) + \mathbf{t})d\mathbf{z} \quad (Eq.16) \tag{19}$$

$$= \int_{\mathbb{R}^2} W_{m_1}(\mathcal{R}_\alpha^{-1}\mathbf{z}')I(\mathcal{R}_\alpha\mathbf{x} - \mathbf{z}' + \mathbf{t})\,d\mathbf{z}' \quad (\mathbf{z}' = \mathcal{R}_\alpha\mathbf{z})^1 \tag{20}$$

$$= e^{im_1\alpha}\int_{\mathbb{R}^2} W_{m_1}(\mathbf{z}')I((\mathcal{R}_\alpha\mathbf{x} + \mathbf{t}) - \mathbf{z}')d\mathbf{z}' \quad \text{(Lemma A.1)} \tag{21}$$

$$= e^{im_1\alpha}\,[W_{m_1} \circledast I](\mathcal{R}_\alpha\mathbf{x} + \mathbf{t}) \tag{22}$$

Denote the first feature map as F, if we roto-translate the input image:

$$[F]_t^\alpha(\mathbf{x}) \equiv e^{im_1\alpha} F(\mathcal{R}_\alpha \mathbf{x} + \mathbf{t}) \tag{23}$$

The following harmonic convolution is given by a similar equation.

$$[W_{m_2} \circledast [F]_t^\alpha](\mathbf{x}) = \int_{\mathbb{R}^2} W_{m_2}(\mathbf{z}) [F]_t^\alpha(\mathbf{x} - \mathbf{z}) d\mathbf{z} \tag{24}$$

$$= e^{im_1\alpha} \int_{\mathbb{R}^2} W_{m_2}(\mathbf{z}) F(\mathcal{R}_\alpha \mathbf{x} - \mathcal{R}_\alpha \mathbf{z} + \mathbf{t}) d\mathbf{z} \tag{25}$$

$$= e^{i(m_1+m_2)\alpha} \int_{\mathbb{R}^2} W_{m_2}(\mathbf{z}') F((\mathcal{R}_\alpha \mathbf{x} + \mathbf{t}) - \mathbf{z}') d\mathbf{z}' \quad (\mathbf{z}' = \mathcal{R}_\alpha \mathbf{z}) \tag{26}$$

$$= e^{i(m_1+m_2)\alpha} [W_{m_2} \circledast F](\mathcal{R}_\alpha \mathbf{x} + \mathbf{t}) \tag{27}$$

Accordingly for all following harmonic convolution layers. □

A.2 Layers Operating on Magnitudes

This section describes the original H-Nets layers that operate on magnitudes as formulated by Worrall et al. [3], and introduces our proposed enhancements.

Definition A.1 ($\mathbb{C}$-ReLU)**.**

$$\mathbb{C}\text{-ReLU}_b(Xe^{i\theta}) = \text{ReLU}(X + b)e^{i\theta}, \tag{28}$$

where $Xe^{i\theta}$ represents a complex number in exponential form, and $b \in \mathbb{R}$ is a learnable bias parameter of the activation function.

Definition A.2 (Harmformer $\mathbb{C}$-ReLU)**.**

$$\mathbb{C}\text{-ReLU}_{a,b}(Xe^{i\theta}) = \text{ReLU}(a \cdot X + b)e^{i\theta}, \tag{29}$$

where $Xe^{i\theta}$ is a complex number in exponential form, and $a, b \in \mathbb{R}$ are learnable parameters of the activation function.

Definition A.1 uses only the bias parameter b. Such an activation function cannot zero out higher values while leaving lower values unaffected. To allow this, our enhanced $\mathbb{C}$-ReLU also incorporates a multiplication by the parameter a.

A.3 Complex Batch Normalization in Harmonic Networks

In H-Nets, batch normalization is adapted from its traditional definition. The layer standardizes only the magnitudes of the complex numbers, leaving the phase components unaffected. The $\mathbb{C}$-BN can be formally defined as follows:

Definition A.3 ($\mathbb{C}$-BN)**.**

$$\mathbb{C}\text{-BN}_{\gamma,\beta}(Xe^{i\theta}) = \left(\gamma\left(\frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}}\right) + \beta\right) e^{i\theta} = \text{BN}_{\gamma,\beta}(X)e^{i\theta}, \tag{30}$$

where $Xe^{i\theta}$ represents a complex number in exponential form, and $\gamma, \beta \in \mathbb{R}$ are learnable scaling and shifting parameters, respectively. Here, μ and σ denote the running sample mean and variance, which are estimated during the training phase and fixed during inference.

However, this formulation can produce negative magnitudes, thus inverting the phase and violating HE. Therefore in Harmformer we instead use a batch normalization integrated with an activation function, that can be defined as:

Definition A.4 (Harmformer HBatchNorm + $\mathbb{C}$-ReLU)**.**

$$\mathbb{C}\text{-BN-ReLU}_{a,b}(Xe^{i\theta}) = \text{ReLU}\left(a\left(\frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}}\right) + b\right) e^{i\theta}, \tag{31}$$

where $Xe^{i\theta}$ represents a complex number in exponential form, and $a, b \in \mathbb{R}$ are learnable scaling and shifting parameters, respectively. Here, μ and σ denote the running sample mean and variance, which are estimated during the training phase and fixed during inference.

Our formulation uses the ReLU function, which maps $\mathbb{R}$ to $\mathbb{R}^+$, to ensure that changes in magnitudes are always positive. Additionally, by integrating the scaling and shifting parameters a and b into batch normalization, the number of learnable parameters is reduced. See section B.1 for a comparison of different normalization layers.

A.4 Discrete Representation

The normalization layers are the last from the stem stage, as can be seen in Figure 2. For the sake of clarity, we will make the transition to discrete space and focus only on rotation for the following layers, as mentioned in Section 5.2. Suppose that the feature maps (stack of patches) $F_m(I) \in \mathbb{C}^{n\times d}$, extracted from the input image I, transforms under a rotation of the input as follows

$$F_m([I]^\alpha) = e^{im\alpha}\,[F_m(I)]^\alpha\,, \tag{32}$$

where m is the rotation order and α is the rotation angle of the image I. Here n is the number of patches and d is the dimension of each patch.

This property implies that $[\cdot]^\alpha$ is a linear operator, thus for the feature maps the following applies:

$$[F_{m_1}(I)]^\alpha + [F_{m_2}(I)]^\alpha = [F_{m_1}(I) + F_{m_2}(I)]^\alpha \tag{33}$$

$$[F_{m_1}(I)]^\alpha \cdot [F_{m_2}(I)]^\alpha = [(F_{m_1}(I) \cdot F_{m_2}(I)]^\alpha \tag{34}$$

A.5 Residual Connection

Lemma A.2 (HE of Residual Connections (Lemma 5.1))**.** A residual connection between feature maps of the same rotation order, $F_m(I)$ and $F'_m(I)$, preserves HE property:

$$F'_m([I]^\alpha) + F_m([I]^\alpha) = e^{im\alpha}\,[(F'_m(I) + F_m(I))]^\alpha\,. \tag{35}$$

Proof. By the properties of harmonic equivariance, we have:

$$F'_m([I]^\alpha) + F_m([I]^\alpha) = (e^{im\alpha}\,[F'_m(I)]^\alpha) + (e^{im\alpha}\,[F_m(I)]^\alpha) \tag{36}$$

$$= e^{im\alpha}([F'_m(I)]^\alpha + [F_m(I)]^\alpha) \tag{37}$$

Since $[\cdot]^\alpha$ is a linear operator, we can combine the rotated feature maps:

$$= e^{im\alpha}\,[(F'_m(I) + F_m(I))]^\alpha \tag{38}$$

□

A.6 Linear Layers

Lemma A.3 (HE of Linear Layer (Lemma 5.2))**.** A linear layer applied to a HE feature map $F_m([I]^\alpha) \in \mathbb{C}^{(hw)\times d_{in}}$ preserves the rotation order m. Formally, we have:

$$F_m([I]^\alpha)W = e^{mi\alpha}\,[F_m(I)W]^\alpha\,, \tag{39}$$

where $W \in \mathbb{C}^{d_{in}\times d_{out}}$ represents a shared weight matrix applied independently over all spatial positions of the input feature map.

Proof. As the matrix W has a rotation order $m = 0$ because it doesn't change under an input rotation. Then the property comes trivially from Eq (34). □

A.7 Multi-Head Self-Attention

Lemma A.4 (HE of Layer Norm (Lemma 5.3))**.** A feature map $F_m([I]^\alpha) \in \mathbb{C}^{(hw)\times d}$ with a rotation order m preserves HE when normalized by its mean and standard deviation:

$$\frac{F_m([I]^\alpha) - \mu}{\sigma + \epsilon} = e^{im\alpha}\frac{[F_m(I)]^\alpha - \mu}{\sigma + \epsilon}, \tag{40}$$

where μ, σ are the sample means and standard deviations of the original feature maps computed over their spatial dimensions, respectively, and ϵ is a small constant added for numerical stability.

Proof.

$$\frac{F_m([I]^\alpha) - \hat{\mu}}{\hat{\sigma} + \epsilon} = \frac{F_m([I]^\alpha) - \frac{\sum F_m([I]^\alpha)}{h \cdot w}}{\hat{\sigma} + \epsilon} \tag{41}$$

By the properties of HE, we express $F_m([I]^\alpha) = e^{im\alpha} F_m(I)$. Sigma of does not change under rotation, because its equal to standard deviation of magnitude in complex numbers.

$$= \frac{e^{im\alpha} \left[F_m(I)\right]^\alpha - \frac{\sum e^{im\alpha} [F_m(I)]^\alpha}{h \cdot w}}{\sigma + \epsilon} \tag{42}$$

$$= e^{im\alpha} \frac{\left[F_m(I)\right]^\alpha - \frac{\sum [F_m(I)]^\alpha}{h \cdot w}}{\sigma + \epsilon} \tag{43}$$

The sum of the feature map is invariant to its rotation.

$$= e^{im\alpha} \frac{\left[F_m(I)\right]^\alpha - \mu}{\sigma + \epsilon} \tag{44}$$

□

Lemma A.5 (Dot product subtracts rotation orders (Lemma 5.4))**.** Consider two HE feature maps $Q_{m_1}([I]^\alpha) \in \mathbb{C}^{(hw) \times d}$ and $K_{m_2}([I]^\alpha) \in \mathbb{C}^{(hw) \times d}$ that represent queries and keys, respectively. The dot product of these feature maps is HE and has the rotation order $m_1 - m_2$. Formally, we have:

$$Q_{m_1}([I]^\alpha) \overline{K_{m_2}([I]^\alpha)^T} = e^{i(m_1 - m_2)\alpha} \left[Q_{m_1}(I) \overline{K_{m_2}(I)^T} \right]^\alpha, \tag{45}$$

where $\overline{K_{m_2}([I]^\alpha)}^T$ denotes the complex conjugate transpose of $K_{m_2}([I]^\alpha)$.

Proof. By the properties of harmonic equivariance, we express:

$$Q_{m_1}([I]^\alpha) = e^{im_1\alpha} \left[Q_{m_1}(I)\right]^\alpha,$$
$$K_{m_2}([I]^\alpha) = e^{im_2\alpha} \left[K_{m_2}(I)\right]^\alpha.$$

Taking the complex conjugate transpose of $K_{m_2}([I]^\alpha)$, we obtain:

$$\overline{K_{m_2}([I]^\alpha)^T} = \overline{e^{im_2\alpha} \left[K_{m_2}(I)^T\right]^\alpha} = e^{-im_2\alpha} \left[\overline{K_{m_2}(I)^T}\right]^\alpha.$$

As is derived from the commutativity of the scalar multiplication with the matrix multiplication.

$$\begin{aligned} Q_{m_1}([I]^\alpha) \overline{K_{m_2}([I]^\alpha)^T} &= e^{im_1\alpha} \left[Q_{m_1}(I)\right]^\alpha e^{-im_2\alpha} \left[\overline{K_{m_2}(I)^T}\right]^\alpha \\ &= e^{i(m_1 - m_2)\alpha} \left[Q_{m_1}(I)\right]^\alpha \left[\overline{K_{m_2}(I)^T}\right]^\alpha \\ &= e^{i(m_1 - m_2)\alpha} \left[Q_{m_1}(I) \overline{K_{m_2}(I)^T}\right]^\alpha. \end{aligned}$$

This shows that the dot product result is also HE with a rotation order of $m_1 - m_2$. □

Lemma A.6 (Matrix multiplication sums rotation orders (Lemma 5.5))**.** Consider a HE feature map $A_{m_1}([I]^\alpha) \in \mathbb{C}^{(hw) \times (hw)}$ representing an attention matrix, and HE feature map $V_{m_2}([I]^\alpha) \in \mathbb{C}^{(hw) \times d}$ representing values. The result of their matrix multiplication is HE with a rotation order $m = m_1 + m_2$:

$$A_{m_1}([I]^\alpha) V_{m_2}([I]^\alpha) = e^{i(m_1 + m_2)\alpha} \left[A_{m_1}(I) V_{m_2}(I)\right]^\alpha. \tag{46}$$

where $[A_{m_1}(I)]^\alpha$, $[V_{m_2}(I)]^\alpha$ are feature maps created from unrotated I and rotated afterwards.

Proof. Proof is analogical to Lemma 5.4 By the properties of harmonic equivariance, the HE feature maps $A_{m_1}([I]^\alpha)$ and $V_{m_2}([I]^\alpha)$ can be represented as:

$$A_{m_1}([I]^\alpha) = e^{im_1\alpha} \left[A_{m_1}(I)\right]^\alpha,$$
$$V_{m_2}([I]^\alpha) = e^{im_2\alpha} \left[V_{m_2}(I)\right]^\alpha.$$

Multiplying these matrices, we find:

$$\begin{aligned} A_{m_1}([I]^\alpha)V_{m_2}([I]^\alpha) &= (e^{im_1\alpha}\,[A_{m_1}(I)]^\alpha)(e^{im_2\alpha}\,[V_{m_2}(I)]^\alpha) \\ &= e^{im_1\alpha}e^{im_2\alpha}\,[A_{m_1}(I)]^\alpha\,[V_{m_2}(I)]^\alpha \\ &= e^{i(m_1+m_2)\alpha}\,[A_{m_1}(I)]^\alpha\,[V_{m_2}(I)]^\alpha\,. \\ &= e^{i(m_1+m_2)\alpha}\,[A_{m_1}(I)V_{m_2}(I)]^\alpha\,. \end{aligned}$$

This confirms that the product is HE and preserves the combined rotation order of $m_1 + m_2$. □

B Ablation Study and Additional Experiments

In addition to the experiments in the main text that compare the Harmformer to other methods, we include an ablation study that demonstrates its rotational robustness and explores other architectural choices. We have omitted the PCam benchmark due to computational constraints, as its training time was extremely long.

B.1 Ablation: Normalization Layers in Stem Stage (S1)

In Section A.4, we propose a modification of batch normalization by integrating it with an activation function. Specifically, we first apply batch normalization to the feature magnitudes, followed by a ReLU activation on these normalized values. This approach is more consistent with the original purpose of batch normalization as formulated by Ioffe and Szegedy [49], which is to standardize the distribution of activations across layers.

To test this novel normalization approach, we replaced our normalization layers in the Harmformer H-Conv block (see Figure 3b) with the original H-Nets batch normalization [3] followed by a $\mathbb{C}$-ReLU. We also evaluated how our proposed layer normalization used in the encoder block would perform in the H-Conv block.

The results depicted in Figure 5 show that our proposed normalization (blue bar) outperforms the original H-Nets normalization layer (red bar) across all three benchmarks, significantly reducing variance across different runs. It also exceeds the performance of layer normalization (yellow bar) in the rotated MNIST and mnist-rot-test, although it slightly underperforms in the cifar-rot-test. In addition, the layer normalization was more computationally expensive according to our experiments.

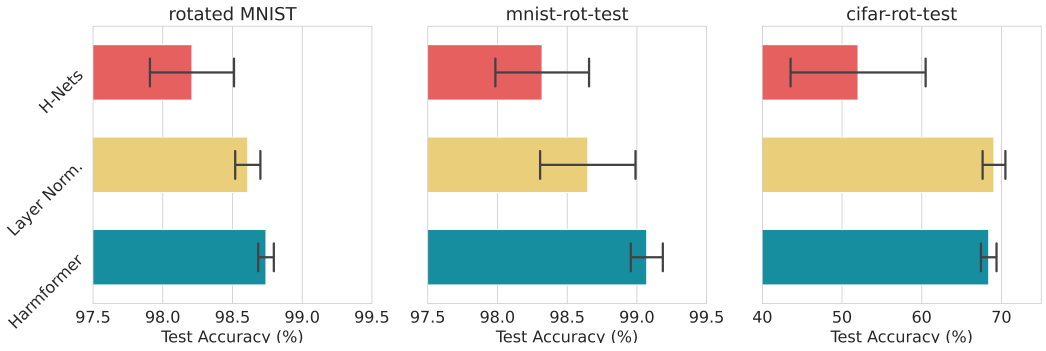

Figure 5: Ablation study on different normalization layers. The rows represent different normalization layers in the H-Conv block. Each plot is aggregated from 5 different runs. The error bars represent the standard deviation.

B.2 Ablation: Mixing Rotation Orders in Self-Attention Mechanism

As mentioned in the main text (Section 5), determining how queries, keys, and values should interact based on their rotation order is not intuitive. Therefore, we have extensively tested various configurations and listed the most promising ones in this section. In choosing the final solution, in addition to performance, we focused on the principles that the number of streams should not increase and that the method should not require extensive computation.

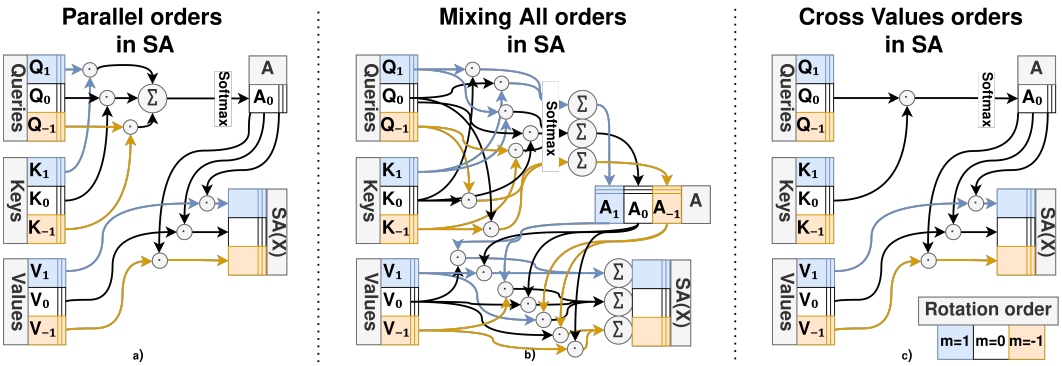

Figure 6: Ablation study on mixing rotation orders within the SA mechanism. a) The principle used in Harmformer b) Mixing all possible combinations of queries, keys and values c) Cross Values a method of mixing only values of different rotation orders.

In Lemmas 5.4 and 5.5, we demonstrate that the dot product subtracts the rotation orders and matrix multiplication sums the rotation orders. Based on this, we propose several configurations, illustrated in Figure 6. Apart from those mentioned here, we investigated learnable weights for each rotation order combination and different placements of softmax or combinations of these configurations together, but none yielded significant improvements. The final configuration used in Harmformer is shown in Figure 6a. The configuration in Figure 6b allows all possible combinations to produce the three streams (1, 0, -1). The last configuration, Cross Values, illustrated in Figure 6c, uses higher rotation orders only for values. Similarly, we tested Cross Keys and Cross Queries only for keys and queries, respectively.

Figure 7 presents the performance of these configurations on our benchmarks. The only configuration surpassing Harmformer (Figure 6a) was Mixing All (Figure 6b) in the case of rotated MNIST and mnist-rot-test. Since the performance difference was minor and the computational demands were significantly higher, we did not use Mixing All in our final architecture.

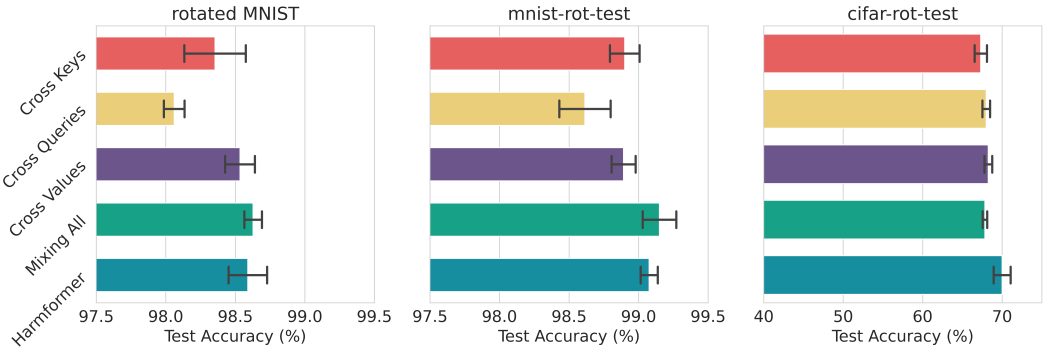

Figure 7: Ablation study on different SA mixing configurations. Each plot is aggregated from 5 different runs. The error bars represent the standard deviation.

B.3 Ablation: Relative Positional Encoding (RPE)

In Harmformer, we use relative circular encoding similar to those published in iRPE [48]. The encoding is added immediately after calculating the dot product between keys and queries. RPE significantly improves performance, as demonstrated in Figure 8, which shows Harmformer performance with and without RPE.

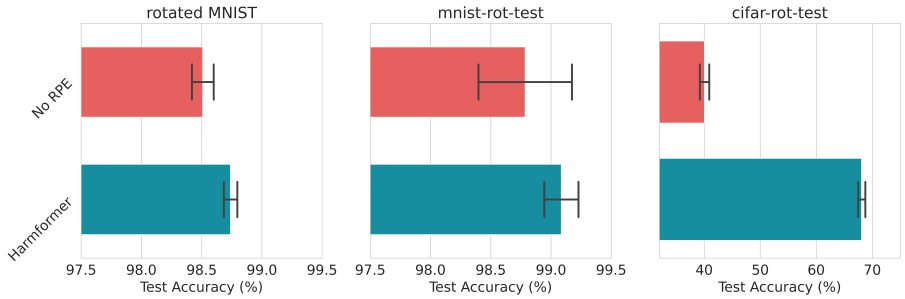

Figure 8: Ablation study on the use of relative circular position encoding (RPE) in the Harmformer. The error bars represent the standard deviation.

B.4 Experiments: Stability of Classification w.r.t. Input Rotation

In these experiments, we investigate the influence of input interpolation errors on performance, following previous invariant models [29, 28, 27]. Stability is primarily examined using the invariance benchmarks mnist-rot-test and cifar-rot-test, where the training data consists of non-rotated images, while the test data consists of randomly rotated images. Note that this implies that the training images consist of original sharp images, but the test images contain images with interpolation errors. In contrast, the rotated MNIST dataset contains rotated images in both the training and test sets, resulting in interpolation errors in both sets.

The test accuracy with respect to the input rotation is shown in Figure 9. Since the rotated MNIST dataset does not contain all images rotated by all angles, we use the original MNIST dataset [1] for this experiment. As a result, the test set, and therefore its accuracy, is different from that described in Section 6 of the main text.

For the mnist-rot-test, we observe very small oscillations, almost the same as for rotated MNIST. The accuracy reaches maxima at $0°, 90°, 180°$, and $270°$, where there is no interpolation effect. For the cifar-rot-test, the oscillations are more significant due to the low resolution of the dataset relative to the complexity of the objects, with minima at $45°, 135°, 225°$, and $315°$, where the interpolation errors are greatest.

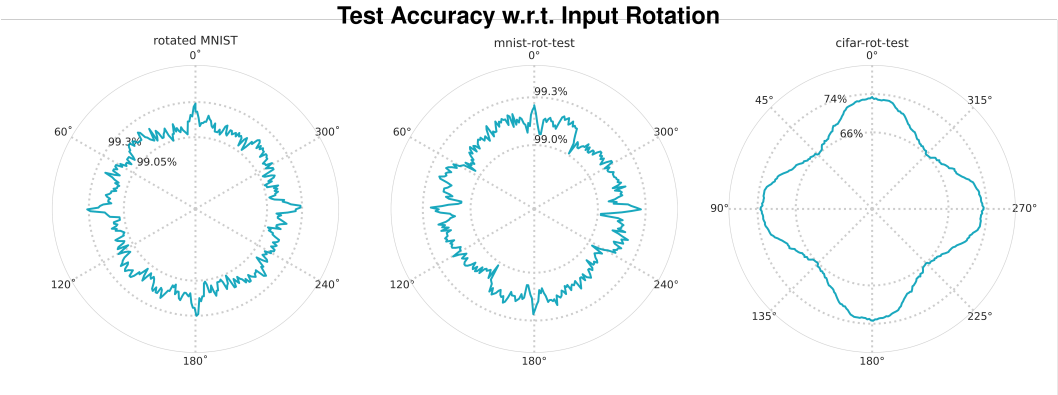

Figure 9: Classification stability with respect to input image rotation. The angle in the circular plots represents the rotation angle of the input image, and the radius represents the test accuracy on specified benchmarks.

For comparison with previous invariant models, we have included Table 7. For the mnist-rot-test, there is a significant improvement in Δ, which represents the difference in accuracy between the interpolation-free and interpolation-affected images. However, for the cifar-rot-test, the gap between these cases remains almost the same. For the MNIST datasets, the results are almost the same whether training with rotated or unrotated data. This leads to the hypothesis that if the resolution of the dataset

Table 7: Test accuracy comparison of the Harmformer and H-NeXt [27] at specific angles of rotation of inputs.

	mnist-rot-test			cifar-rot-test		
Model	$0°$	$45°$	Δ	$0°$	$45°$	Δ
H-NeXt [27]	98.9%	97.8%	1.1%	64.5%	57.4%	7.1%
Harmformer	99.2%	99.1%	0.1%	73.4%	66.1%	7.3%

matches the complexity of the recognition task, the Harmformer should not suffer from interpolation errors. However, this hypothesis would require further testing on large-scale datasets, which is beyond the scope of this paper.

B.5 Experiments: Evaluating the Role of Harmonic Convolutions

To investigate the importance of the convolutional stem and the encoder, we conducted an experiment using a minimal stem stage with an enlarged attentive field. The purpose of this setup was to ensure that the recognition was not due to convolution alone. This configuration contained only three convolutional layers and a single pooling layer.

The results, as shown in Table 8, indicate that the model performance remains within the expected error range despite the simplified convolutional stem. This suggested that the encoder plays a crucial role in the final classification. Notably, the inclusion of a single pooling layer significantly enhances the complexity of subsequent attention mechanisms. Due to the increased GPU RAM requirements, this configuration was exclusively tested on the rotated MNIST dataset.

Table 8: Performance comparison of the Harmformer architecture with a shallow stem stage on the rotated MNIST dataset.

Model	SA Input Shape	Test Error	Params.
Shallow Stem Stage	$32 \times 32 \times 16$	1.29%	40k
Harmformer	$16 \times 16 \times 16$	$1.26\% \pm 0.055$	30k

C Experimental Setup

C.1 Compute Resource

Each experiment was run on a single GPU within our shared, small but diverse cluster comprising 17 GPUs. The cluster includes Tesla P100, V100, and A100 models, NVIDIA GeForce RTX 2080 Ti, 3080, 4090, RTX A5000, and a Quadro P5000. Despite its limited size, our setup allowed for flexible and scalable computation using various GPU configurations. To provide a better overview, Table 9 lists the epoch training time across each experiment on the NVIDIA GTX 4090.

Table 9: Training time of one epoch across different benchmarks on the NVIDIA GTX 4090.

GPU Model	mnist-rot-test	cifar-rot-test	rotated MNIST	PCam
Epoch Training time (mm:ss)	01:02	02:18	00:16	37:40
Number of Training Samples	52k	42k	10k	262k
Batch Size	32	32	32	8

C.2 Computation Complexity w.r.t. Non-Equivariant Convolution and SA Mechanism

In general, equivariant networks usually due to their properties impose higher computation complexity than their classical counterparts. For example, a single classical convolution has complexity $O(N^2 \cdot n^2)$, where $N \times N$ is the spatial dimension of the output feature map and $n \times n$ is the size of

the filter. In contrast, the original G-CNN equivariant to rotation and translation has complexity $O(N^2 \cdot n^2 \cdot |\theta|^2)$, where θ is the number of elements in the rotation group. Thus, a G-CNN equivariant to 90-degree rotations and translation would have $|\theta| = 4$.

Harmformer Convolution In Harmformer stem stage, we use convolution layers similar to H-Nets. This has a complexity of $O(N^2 \cdot n^2 \cdot |o|^2)$, where $|o|$ is the number of rotation orders of the input and output feature maps.

Harmformer SA mechanism Classical global SA mechanism has a complexity of $O(N^2 \cdot d + N \cdot d^2)$, where N is the number of patches and each patch has dimension d. Our SA mechanism, as shown in Figure 4b, adds multiplication by rotation orders o for matrix multiplication and dot product, resulting in a complexity of $O(o \cdot N^2 \cdot d + o \cdot N \cdot d^2)$.

Additional Computational Considerations Harmformer operates in the complex domain, where each multiplication requires four times and each addition requires two times more operations than their real counterparts. Additionally, the computational load increases due to upscaling the input and using large convolution kernels, as recommended in H-NeXt [27]. These factors also contribute to the overall complexity of Harmformer.

C.3 Configurations of Experiments

This subsection details the specific configurations of the Harmformer architecture used in the experiments described in Section 6. For convenience, Figure 10, which depicts the complete Harmformer architecture, is included. The parameters for each dataset are enumerated in the following tables: Tables 12 and 13 for the MNIST-rot-test and rotated MNIST datasets; Tables 14 and 15 for the CIFAR-rot-test dataset; and Tables 10 and 11 for the PCam dataset.

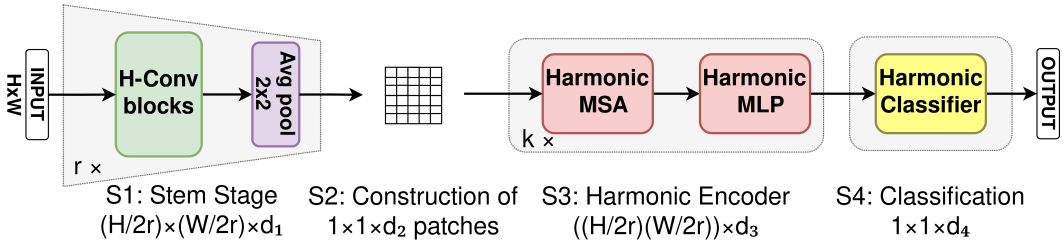

Figure 10: Reproduced from Figure 2 to detail the number of parameters for each experiment.

Table 10: PCam: Architecture

Parameter	Value
Number of S1 Blocks (r)	4
Convolution per S1 Block	2
Number of S1 Channels (d_1)	$[4, 8, 16, 32]$
S1 Channels Dropout	$[0, 0.2, 0.3, 0.4]$
Number of S3 Encoders (k)	4
Number of S3 Heads	4
Shape of S3 Patches (d_3)	8
MSA&MLP Dropout	0.4

Table 11: PCam: Training Settings

Parameter	Value
Epochs	100
Batch Size	8
Learning Rate	0.0007
Label Smoothing	0.1
Scheduler	Cosine
Optimizer	AdamW
Weight Decay	0.01
Runs	5
Input Padding	0

Table 12: MNIST datasets: Architecture

Parameter	Value
Number of S1 Blocks (r)	2
Convolution per S1 Block	2
Number of S1 Channels (d_1)	$[8, 16]$
S1 Channels Dropout	$[0, 0]$
Number of S3 Encoders (k)	3
Number of S3 Heads	1
Shape of S3 Patches (d_3)	16
MSA&MLP Dropout	0.1

Table 13: MNIST datasets: Training Settings

Parameter	Value
Epochs	100
Batch Size	32
Learning Rate	0.007
Label Smoothing	0.1
Scheduler	Reduce LR on Plateau
Optimizer	AdamW
Weight Decay	0.01
Runs	5
Input Padding	2

Table 14: cifar-rot-test: Architecture

Parameter	Value
Number of S1 Blocks (r)	2
Convolution per S1 Block	3
Number of S1 Channels (d_1)	$[8, 16]$
S1 Channels Dropout	$[0, 0.1]$
Number of S3 Encoders (k)	4
Number of S3 Heads	4
Shape of S3 Patches (d_3)	8
MSA&MLP Dropout	0.2

Table 15: cifar-rot-test: Training Settings

Parameter	Value
Epochs	200
Batch Size	32
Learning Rate	0.007
Label Smoothing	0.1
Scheduler	Cosine
Optimizer	AdamW
Weight Decay	0.01
Runs	5
Input Padding	0

D Segmentation experiment: Retina blood vessel segmentation

To demonstrate the generalizability and scalability of our architecture beyond classification tasks, we introduce Harmformer for retinal blood vessel segmentation using the DRIVE dataset [50]. The DRIVE is binary segmentation task, where the goal is to extract retinal blood vessels from an RGB image.

The dataset contains 262,080 samples for training and 65,520 samples for validation, similar to the settings of [51]. Each sample consists of an input image of size $3 \times 64 \times 64$ and a target segmentation mask of size 64×64. These samples were generated from 17 training images and 3 validation images, each of which is 768×584 pixels and represents a different patient.

To use Harmformer as an image-to-image model, we adopt a U-Net [52] architecture in Fig 11a. Unlike our classification models (Section 6), this model processes the images at their original resolution, without any upscaling before they enter the network. For the output, we use only the magnitude of the final feature maps. To merge the hidden features (channels) into a single output layer, we apply a standard 2D convolution layer at the end.

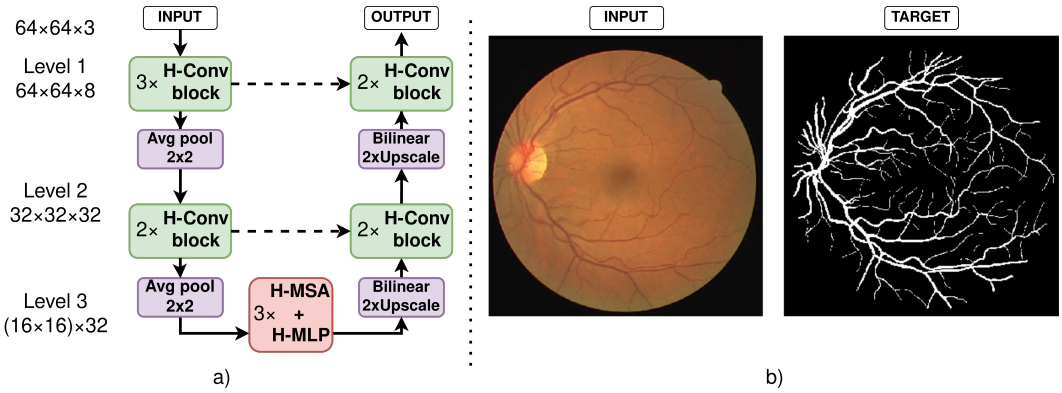

Figure 11: (a) Diagram of the Harmformer architecture for image segmentation. (b) Example of an image from the DRIVE dataset: the RGB image and the target segmentation mask.

We trained the U-Net Harmformer for 20 epochs with the AdamW optimizer, a learning rate of 0.001 and 64 batch size. For augmentation, we used horizontal and vertical flipping, color jitter, and auto-contrast. We ran 4 different experiments with different seeds.

The results are shown in Table 16, using the area under the receiver operating characteristic curve (AUC) as the evaluation metric. For completeness, we have also included the performance of G-CNNs and the current state-of-the-art model FR-UNet [53]. As expected, these results are consistent with the findings in the paper, with Harmformer slightly underperforming compared to equivariant convolution architectures. Nevertheless, we show that our architecture is versatile and can also be applied to non-classification tasks.

Table 16: AUC for DRIVE segmentation [50].

Model	AUC	Equivariant model
Harmformer	0.9746 ± 0.0002	✓
G-CNNs [51]	0.9784 ± 0.0001	✓
FR-UNet [53]	0.9889	✗

E Differences Between 2D and 3D Equivariant Transformers

While 2D equivariant transformers [21, 22] have been relatively understudied, 3D equivariant transformers [17, 14, 15, 18, 12, 10] have received more attention. In this section, we aim to highlight the key differences that make the 2D case unique, and compare Harmformer with the most closely related $SE(3)$-Transformer, which operates in 3D but also uses steerable basis representations.

An important distinction lies in the nature of the input data, which directly influences the transformer architecture. While 2D datasets typically consist of dense pixels with highly correlated neighborhoods, 3D equivariant datasets, often represented as graphs or point clouds, tend to be sparse. In 3D, neighboring elements can vary significantly; for example, in molecular graphs [14, 15], atoms can fulfill entirely different roles within the structure.

Patches The properties of the input data determine how to prepare patches in an equivariant manner. In the case of the $SE(3)$-Transformer, each node of the graph can be directly treated as a patch, eliminating the need for a stem stage. For Harmformer, on the other hand, it is necessary to aggregate low-level correlated data into a higher-level representation. Additionally, the classical (16×16) ViT [30] grid cannot be used, as discussed in Section 3. Therefore, we employ a convolutional stem stage, where the convolution kernels are expressed using circular harmonics to maintain equivariance.

This reliance on harmonic representations is a common feature between Harmformer and the $SE(3)$-Transformer. While Harmformer uses circular harmonics, the $SE(3)$-Transformer uses spherical harmonics. Both approaches leverage steerable basis functions [23], which are widely used in equivariant networks [39, 34, 54]. These steerable bases change predictably under rotation, allowing the effects of rotation to be effectively neutralized—via phase shifts in circular harmonics and via the Wigner-D matrix in spherical harmonics. It is important to note that the use of steerable bases predates both transformers, as shown in [23, 39, 54].

Queries, Keys, and Values In Harmformer, queries (Q), keys (K), and values (V) are generated independently from individual patches through a linear layer, we proposed in Section 5.2. In contrast, the $SE(3)$-Transformer creates them by applying convolutions across points (i.e., patches), using steerable spheres that aggregate information from the local neighborhood.

Attention The $SE(3)$-Transformer focuses exclusively on invariant attention (type-0) and applies only local attention. In contrast, Harmformer explores multiple strategies for mixing attention and values of various orders (types), while performing global attention across the entire image. Additionally, Harmformer introduces an equivariant layer normalization at the beginning of the attention layer, while the $SE(3)$-Transformer does not use any layer normalization. Other minor distinctions include Harmformer's use of an improved activation function and relative embeddings.

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: All claims are supported by either theoretical proofs or experiments.

2. **Limitations**

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

   Answer: [Yes]

   Justification: All the limitations are discussed in the last section of the paper and all assumptions are clearly stated.

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: All theoretical statements are accompanied by proofs. The proofs are included in the Appendix.

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: The code is included in the submission, we only use open source datasets. All trained models are also included and their settings are clearly stated in the appendix.

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: Yes, we have included our source code with the submission. If the submission is accepted, we will publish the code on github. The code includes preparation, download scripts for all datasets, and the experimental settings.

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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This information is listed in the Appendix and is also included in the code submission.

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: We show the best results along with the average and standard deviation of at least 5 runs with the performance of our models.

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 enlisted the compute resources in appendix.

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: To the best of our knowledge, we have not violated any part of the NeurIPS Code of Ethics.

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: Our paper focuses on representation learning and model robustness to transformations. We believe that there is no direct path to negative applications.

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks that we are aware of.

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: We provide a citation to the paper presenting the benchmarks, models, and the code included in the supplemental materials contains references to the libraries or repositories from which we drew inspiration.

13. **New Assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

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: [NA]

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

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: [NA]

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