Equivariant Mesh Attention Networks

Anonymous authors
Paper under double-blind review

Abstract

Equivariance to symmetries has proven to be a powerful inductive bias in deep learning research. Recent works on mesh processing have concentrated on various kinds of natural symmetries, including translations, rotations, scaling, node permutations, and gauge transformations. To date, no existing architecture is equivariant to *all* of these transformations. Moreover, previous implementations have not always applied these symmetry transformations to the test dataset. This inhibits the ability to determine whether the model attains the claimed equivariance properties. In this paper, we present an attention-based architecture for mesh data that is *provably* equivariant to all transformations mentioned above. We carry out experiments on the FAUST and TOSCA datasets, and apply the mentioned symmetries to the test set *only*. Our results confirm that our proposed architecture is equivariant, and therefore robust, to these local/global transformations.

1 Introduction

Equivariance to symmetries has proven to be a powerful inductive bias in machine learning tasks ranging across classification, regression, segmentation, and reinforcement learning. Commonly studied symmetries include permutations Keriven & Peyré (2019); Zaheer et al. (2017), rotations Cohen & Welling (2016); Weiler et al. (2018); Li et al. (2018); Veeling et al. (2018), translations LeCun et al. (1998), scaling Sosnovik et al. (2020), and gauge transformations Cohen et al. (2019a); de Haan et al. (2021).

These symmetries lead to different requirements for invariance or equivariance, depending on the learning task. For example, in a mesh shape classification task, we would like a classifier to consistently predict the same label, *invariant* to whether the input mesh is rotated, scaled or translated. In a node segmentation task, in addition to these invariances, we expect that model predictions change in concurrence with a relabelling of the nodes, being permutation *equivariant*.

Achieving models with guaranteed equivariance is of high practical interest. This geometric paradigm is a principled way of incorporating task-specific prior information, which allows the model to simultaneously handle "equivalence classes of inputs" (i.e. those inputs related by symmetry transformations), and removes the need for data augmentation during training Bronstein et al. (2021); Cohen (2021).

The main challenge in realizing this equivariance in the case of meshes lies in *adequately* handling the (arbitrary) numerical representation associated with the components of a mesh, since symmetry transformations affect these components in different ways. For example, a rotation modifies the positions of the nodes but leaves the face areas intact, while scaling leaves angles between nodes unchanged, but affects the positions, edge lengths and face areas. Therefore, the model predictions should depend *only* on the geometry of the mesh, and not on a particular embedding of it.

Thus, performing computations with geometric quantities *intrinsic* to the mesh is of critical importance towards achieving equivariant models. Our first contribution is the use of ***relative tangential features***. These features correspond to a map of absolute node positions to tangent vectors that accounts for the local geometry at each node. We show that this map is equivariant under global rotations, and invariant under translations and scaling of the ambient space $\mathbb{R}^3$. Relative tangent features constitute a simple yet effective alternative to using as inputs raw node positions.

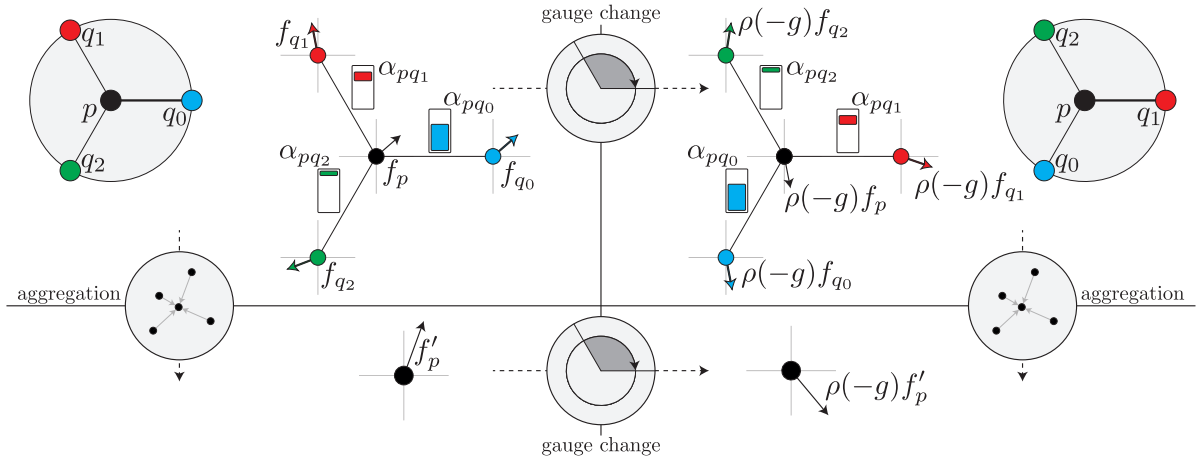

Figure 1: Message passing mechanism in Equivariant Mesh Attention Networks. For convenience, we represent a planar portion of the mesh and therefore ignore parallel transport. Tangent vectors f_p, f_{q_i} are aggregated according to the attention coefficients α_{pq_i} (on the figure, going from top to bottom). A change of gauge reference neighbor (from q_0 to q_1) determines a rotation $\rho(-g)$ on tangent vectors of angle $-g$ (on the figure, going from left to right). Attention coefficients are invariant under gauge change. Pictorially, gauge equivariance can be rephrased as: "go right, go down" is the same as "go down, go right".

Moreover, relative tangent features are *geometric features*. Although the numerical representation of these features may change depending on the choice of gauge, the geometric quantities they represent remain unchanged. The framework of gauge equivariance prescribes constraints on functions among geometric feature spaces, ensuring coherent compositionality. We leverage the gauge equivariant convolutions (GEM-CNN) of de Haan et al. (2021) as a building block to satisfy these equivariance constraints.

It is precisely the conjunction of relative tangent features and gauge equivariant convolutions that allows us to design models that exhibit equivariance/invariance to **all** the global transformations mentioned above.

Our second contribution is an extension of mesh processing algorithms to include a ***gauge equivariant attention mechanism*** in a way that provably satisfies the aforementioned requirements. We refer to this architecture as Equivariant Mesh Attention Networks (EMAN). Attention mechanisms have been a groundbreaking innovation in deep learning powering state-of-the-art results in natural language processing Vaswani et al. (2017); Devlin et al. (2019); Brown et al. (2020), computer vision Dosovitskiy et al. (2021), reinforcement learning Chen et al. (2021), multi-modal learning Jaegle et al. (2021), and graph neural networks Veličković et al. (2018); Chamberlain et al. (2021). However, attention-based methods remain largely unexplored in the context of mesh data.

We carry out experiments on the FAUST Bogo et al. (2014) and TOSCA Bronstein et al. (2008) datasets. Our models are trained on un-transformed meshes (i.e. no data augmentation is used in training) and we report their performance on several copies of the test set to which rotations, translations, scalings, permutations and changes of gauge have been applied. Our results confirm that our proposed model design is the only one to achieve equivariance, and therefore robustness, to all of these local/global transformations

Our code* is available at: `See anonymized code.`

*Based on official implementation by de Haan et al. (2021)

2 Geometry

In this section we describe meshes, geometric features, parallel transport, and equivariances and invariances.

2.1 Meshes

A ***mesh*** $\mathcal{M}$ is determined by a set of vertices, or nodes, a set of (undirected) edges, and a set of faces. We consider oriented meshes embedded in the ambient space $\mathbb{R}^3$ and require the faces to "properly glue along the edges", so that $\mathcal{M}$ is in fact a piece-wise linear sub-manifold of $\mathbb{R}^3$. One can think of the mesh $\mathcal{M}$ as a discretization (e.g., a triangulation) of a 2-dimensional manifold.

A 2-dimensional sub-manifold of $\mathbb{R}^3$ admits a ***tangent plane*** at each point. The discrete equivalent for a mesh $\mathcal{M}$, at a point $p \in \mathcal{M}$, is the plane orthogonal to the normal vector

$$n_p = \frac{\sum_{F \ni p} \mathcal{A}(F)\, n_F}{||\sum_{F \ni p} \mathcal{A}(F)\, n_F||}. \tag{1}$$

The normal vector n_p is computed as a weighted sum of the normal vectors to the adjacent faces, where faces F (which contain p) with greater area $\mathcal{A}(F)$ contribute more.

For the tangent plane $T_p\mathcal{M}$ at p, we consider a ***gauge***, or frame, $\mathbf{E}_p = \{e_{p,1}, e_{p,2}\}$. We only allow orthonormal gauges, for which the triple $\mathbf{E}_p \cup \{n_p\}$ constitutes a positively oriented orthonormal basis of $\mathbb{R}^3$. Therefore, any other *admissible* orthonormal frame of $T_p\mathcal{M}$ is obtained from the previous gauge $\mathbf{E}_p$ by a rotation $g \in \mathrm{SO}(2)$. We denote by g both the rotation and its corresponding angle, modulo 2π.

Finally, we define angles θ_{pq}, which take into account the local orientation of the neighbors of a node p. θ_{pq} is the angle between the projection of the vector $q - p$ onto $T_p\mathcal{M}$ and the reference axis $e_{p,1}$. The angles θ_{pq} are thus gauge-dependent quantities.

2.2 Geometric Features

Geometric features are a central concept in our work. Here we closely follow the presentation and notation of de Haan et al. (2021). Meshes possess a richer structure than mere graphs. An important insight in geometric deep learning is to take the mesh structure in consideration, and allow the features on the underlying space not to be simple *functions* on the space, but rather *sections of vector bundles*.

As an example, a tangential feature f on a mesh is given by a tangent vector in the plane $T_p\mathcal{M}$, for each point $p \in \mathcal{M}$. To represent such a vector, we denote by f_p the 2-dimensional vector of coordinates of the tangent vector, with respect to the gauge $\mathbf{E}_p$. Note that the coordinate vector f_p is dependent on the choice of a gauge.

Therefore, as we change the gauge at p, we should *prescribe* how the coordinate vector f_p gets modified. For tangential features, f_p changes as $f_p \mapsto \rho_1(-g)f_p$, where $\rho_1(-g)$ denotes the rotation by the angle $-g$. This transformation rule precisely characterizes tangential features. In contrast, scalar features are 1-dimensional features that *do not change* when the gauge is transformed, and thus we may write $f_p \mapsto \rho_0(-g)f_p$, where $\rho_0(-g) = 1$ for all $g \in \mathrm{SO}(2)$.

More generally, for $n \in \mathbb{N}$, $n \geq 1$, we can consider 2-dimensional features f, that change under a gauge change $g \in \mathrm{SO}(2)$ as $f_p \mapsto \rho_n(-g)f_p$, where $\rho_n(-g)$ denotes the rotation by angle $-ng$. The ***representations*** ρ_n are

$$\rho_0(g) = 1, \quad \rho_n(g) = \begin{pmatrix} \cos ng & -\sin ng \\ \sin ng & \cos ng \end{pmatrix} \text{ for } n \geq 1.$$

We say that a feature is of ***type*** ρ if it changes accordingly to the representation ρ. The ρ_n form an exhaustive list of irreducible representations, the building blocks of all finite-dimensional representations of $\mathrm{SO}(2)$. In other words, there is no loss of generality in considering *only* geometric features corresponding to direct sums of such representations, obtained by concatenation of multiple irreducible components. We consider these types of features throughout the rest of the paper. For example, $\rho = 4\rho_0 \oplus \rho_1 \oplus 3\rho_2$ corresponds to a feature type of dimension $4 \cdot 1 + 1 \cdot 2 + 3 \cdot 2$. Note also that these features are *orthogonal*, meaning that $\rho^\top = \rho^{-1}$.

2.3 Parallel Transport

Message passing updates involve processing features stored at different nodes of the mesh. However, geometric features present a challenge. For instance, tangential features stored at different nodes belong to different tangent planes, and thus are not immediately comparable. ***Parallel transport*** is a procedure from differential geometry that describes how to "coherently translate" between tangent planes at different points, respecting the curvature of the manifold.

Discrete parallel transport can be intuitively understood as follows: for a node $p \in \mathcal{M}$, and a neighbor $q \in \mathcal{N}_p$, we first translate the tangent plane $T_q\mathcal{M}$, together with the normal vector n_q, to p, along the edge joining q and p. Then, we consider the unique rotation of $\mathbb{R}^3$ that maps n_q to n_p, with fixed axis $n_q \times n_p$. Under this rotation, $T_q\mathcal{M}$ is mapped onto $T_p\mathcal{M}$, and it is now coherent to compare tangent vectors at q with tangent vectors at p.

However, a feature f on $\mathcal{M}$ is represented by coordinates f_q and f_p *with respect to two different gauges*. In general, even after rotating $T_q\mathcal{M}$ onto $T_p\mathcal{M}$, the two gauges do not coincide. Therefore, we denote by $-g_{q\to p} \in \mathrm{SO}(2)$ the 2-dimensional rotation corresponding to the gauge change from the given gauge at q (after rotating $T_q\mathcal{M}$ onto $T_p\mathcal{M}$), and the given gauge at p. It is now coherent to compare f_p with $\rho(g_{q\to p})f_q$, as they both represent coordinates of geometric features at p of the same type, *with respect to the same gauge*. If we denote by $R_{q\to p}$ the unique rotation of $\mathbb{R}^3$ that maps n_q to n_p, with fixed axis $n_q \times n_p$, we obtain:

$$g_{q\to p} = \mathrm{atan2}\big((R_{q\to p}e_{q,2})^\top e_{p,1}, (R_{q\to p}e_{q,1})^\top e_{p,1})\big).$$

2.4 Equivariances and Invariances

Throughout this section, we denote by $\mathcal{F}_{\mathrm{in}}$ and $\mathcal{F}_{\mathrm{out}}$ the spaces of features of type ρ_{in} and ρ_{out}, respectively, and by $\mathcal{F}$ a generic space of features of type ρ.

Gauge equi/invariance. To coherently define a feature mapping $\mathcal{K}\colon \mathcal{F}_{\mathrm{in}} \to \mathcal{F}_{\mathrm{out}}$, we require its computation to be independent of the choice of the gauge. Consider an arbitrary change of gauge $g \in \mathrm{SO}(2)$. Since features of type ρ_{in} transform as $f_p \mapsto \rho_{\mathrm{in}}(-g)f_p$, and similarly for ρ_{out}, we demand $\mathcal{K}$ to satisfy the ***gauge equivariance*** constraint

$$\rho_{\mathrm{out}}(-g) \circ \mathcal{K} = \mathcal{K} \circ \rho_{\mathrm{in}}(-g). \tag{2}$$

If the representation ρ_{out} is a (direct sum of) scalar feature(s) ρ_0, then we talk about ***gauge invariance***, as the resulting features are insensitive to the particular choice of gauge.

Global rotation equi/invariance. Given a gauge-equivariant feature mapping $\mathcal{K}\colon \mathcal{F}_{\mathrm{in}} \to \mathcal{F}_{\mathrm{out}}$, we study its interaction with a global rotation $R \in \mathrm{SO}(3)$. Denote by $R\mathcal{M}$ the mesh obtained by rotating $\mathcal{M}$. We write $\mathcal{K}^{\mathcal{M}}\colon \mathcal{F}_{\mathrm{in}}^{\mathcal{M}} \to \mathcal{F}_{\mathrm{out}}^{\mathcal{M}}$ and $\mathcal{K}^{R\mathcal{M}}\colon \mathcal{F}_{\mathrm{in}}^{R\mathcal{M}} \to \mathcal{F}_{\mathrm{out}}^{R\mathcal{M}}$, to distinguish between *the same feature mapping* applied to feature spaces *on two different meshes*.

If $f \in \mathcal{F}^{\mathcal{M}}$, the rotation R pushes f to a feature $R_*f \in \mathcal{F}^{R\mathcal{M}}$, as follows: if f is represented at p by the coordinate vector f_p relative to the gauge $\mathbf{E}_p$, then R_*f is represented at Rp by the same coordinate vector $(R_*f)_{Rp} = f_p$, relative to the rotated gauge $R\mathbf{E}_p$. ***Global rotation equivariance*** of the feature mapping $\mathcal{K}$ is given by:

$$R_* \circ \mathcal{K}^{\mathcal{M}} = \mathcal{K}^{R\mathcal{M}} \circ R_*. \tag{3}$$

Again, if the representation ρ_{out} is one or a sum of scalar features ρ_0, then we talk about ***global rotation invariance***.

Global translation invariance. A translation $Tp = p + x$ of $\mathbb{R}^3$, $x \in \mathbb{R}^3$, trivially pushes features on $\mathcal{M}$ to features on $T\mathcal{M}$: if $f \in \mathcal{F}^{\mathcal{M}}$ is represented by f_p in the gauge $\mathbf{E}_p$ at p, then $T_*f \in \mathcal{F}^{T\mathcal{M}}$ is represented by the same coordinate vector $(T_*f)_{Tp} = f_p$ in the same gauge $\mathbf{E}_p$ at Tp. Intuitively, it is the same feature, stored at translated points. Therefore, we say that a feature mapping $\mathcal{K}$ is ***global translation invariant*** if

$$T_* \circ \mathcal{K}^{\mathcal{M}} = \mathcal{K}^{T\mathcal{M}} \circ T_*. \tag{4}$$

Global scaling invariance. A scaling $Sp = \lambda p$ of $\mathbb{R}^3$, $\lambda > 0$, similarly allows a definition of a push-forward $S_* \colon \mathcal{F}^{\mathcal{M}} \to \mathcal{F}^{S\mathcal{M}}$: if $f \in \mathcal{F}^{\mathcal{M}}$ is represented by f_p in the gauge $\mathbf{E}_p$ at p, then $S_* f_p \in \mathcal{F}^{S\mathcal{M}}$ is represented by the same coordinates $(S_* f)_{Sp} = f_p$ in the gauge $\mathbf{E}_p$ at Sp.

While for rotations R and translations T new gauges are obtained through the differentials dR and dT, this is not the case for scalings S, since the new gauge would not be normalized. With our definition, tangential features *do not scale* as the mesh scales, and preserve their norm. For this reason, we say that a feature mapping $\mathcal{K}$ is ***global scaling invariant*** (and not equivariant) if

$$S_* \circ \mathcal{K}^{\mathcal{M}} = \mathcal{K}^{S\mathcal{M}} \circ S_*. \tag{5}$$

Node permutation equi/invariance. Consider a re-labeling of the nodes in a mesh $\mathcal{M}$. We talk about ***node permutation equivariance*** (resp. ***node permutation invariance***) when the outcome of a process computed over rearranged nodes is the rearrangement of the outcome from the original ordering, under the same permutation (resp. does not depend on the ordering of the nodes). We expect a segmentation model to be permutation equivariant, while a classification model should be permutation invariant.

3 Graph Convolution on Meshes

If we ignore the faces of a mesh and its embedding in $\mathbb{R}^3$, we obtain a graph. The works of Scarselli et al. (2009); Bruna et al. (2013); Defferrard et al. (2016); Kipf & Welling (2017) led to Graph Convolutional Networks (GCNs), which give an efficient algorithm for node classification. However, GCNs do not account for the local geometry of the mesh and use the same *isotropic* kernel to process signals from all neighbors.

Gauge Equivariant Mesh Convolutional Neural Networks (GEM-CNNs) de Haan et al. (2021) were introduced to cope with this geometric obstacle. Their method uses ***anisotropic*** kernels for the update step, that depend on the spatial disposition of the neighboring nodes. The message passing in a GEM-CNN is performed as

$$f'_p = K_{\text{self}} f_p + \sum_{q \in \mathcal{N}_p} K_{\text{neigh}}(\theta_{pq}) \rho_{\text{in}}(g_{q \to p}) f_q. \tag{6}$$

The kernel $K_{\text{neigh}}(\theta)$ depends on the angle θ_{pq} formed by the vertex q, with respect to the reference gauge at p. Moreover, the kernels K_{self} and $K_{\text{neigh}}(\theta)$ satisfy geometric constraints, so that the output feature f'_p transforms accordingly to the change of gauge. Hence, GEM-CNNs take into account the local geometry of the mesh, while also ensuring *equivariance* to change of gauge. In particular, K_{self} and $K_{\text{neigh}}(\theta)$ satisfy the equations

$$K_{\text{self}} = \rho_{\text{out}}(-g)\ K_{\text{self}}\ \rho_{\text{in}}(g), \qquad K_{\text{neigh}}(\theta - g) = \rho_{\text{out}}(-g)\ K_{\text{neigh}}(\theta)\ \rho_{\text{in}}(g).$$

For details on K_{self} and $K_{\text{neigh}}(\theta)$, please see Appendix A, and (de Haan et al., 2021).

Furthermore, GCNs present another limitation: neighbors $q \in \mathcal{N}_p$ provide information for the update at vertex p with the same influence, and no measure of similarity, or alignment, of the features f_p and f_q is incorporated. Veličković et al. (2018) introduced Graph Attention Networks (GATs). In the update step, the message passed from neighbors is scaled using attention weights α_{pq} dependent on f_p and f_q.

We compactly combine these solutions to produce a gauge equivariant model that includes attention. The convolutional update is determined by:

$$f'_p = \sum_{q \in \mathcal{N}_p} \alpha_{pq} K(\theta_{pq}) \rho_{\text{in}}(g_{q \to p}) f_q. \tag{7}$$

The anisotropic kernel $K(\theta)$ is gauge equivariant, and the attention coefficients α_{pq} are scalar features (namely, unaffected by changes of gauge). A precise description of the attention mechanism follows in Section 6.

Notice that the expected self-contribution $\alpha_{pp} K_{\text{self}} f_p$ is not present in the formula; the implementation of GEM-CNNs we used for our work did not include the self-contribution in the convolutional step. Aware of

this, we nonetheless decided not to include the self-contribution in our model, as the reached performance was already satisfying. For completeness, we provide the details of the model variant including self-contribution in Appendix G.6.

4 Relative Tangent Features

Given a mesh $\mathcal{M}$, we construct ***relative tangent features*** v_p, depending on the local geometry of the mesh $\mathcal{M}$ around a node p. We use the adjective *relative* to underline their dependency on the relative node positions $p - q$, rather than node positions themselves. We use them as initial features for our model, in order to provide global rotational equivariance, and invariance under translations and scaling of the ambient space $\mathbb{R}^3$. At a node p, we define the 3-dimensional vector v_p as:

$$v_p(r) = \frac{1}{N_p^{3/2}} \left(\sum_{q \in \mathcal{N}_p} \frac{\pi_p(q-p)}{||q-p||^r} \right) \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1} \right),$$

where N_p denotes the degree of node p. The projector π_p onto the tangent plane $T_p\mathcal{M}$ is $I - n_p n_p^\top$. We call the real number r ***relative power***. Multiple values of the relative power r provide diverse relative tangent initial features, and multiple initial tangent features may used in our model.

We focus on the central term in the product defining $v_p(r)$. As the relative power r decreases, neighbors far from p contribute more to the relative tangent feature $v_p(r)$. In contrast, as the relative power increases, neighbors close to p are more relevant in the computation of $v_p(r)$ (compare with Fig. 2). The last term of the product is included to provide invariance under scaling. For an explanation of the factor $N_p^{-3/2}$, see the Appendix B.1.

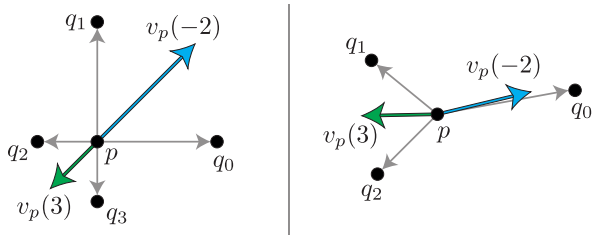

Figure 2: Examples of computation of relative tangent features. Blue and green vectors represent relative tangent features $v_p(r)$ for choices of relative power $r = -2$ and $r = 3$, respectively. For small relative powers r, neighbors far from p contribute more to the relative tangent feature $v_p(r)$, and vice versa.

Relative tangent features provide a suitable geometric input, as they satisfy the following properties:

Lemma 4.1. *For any $r \in \mathbb{R}$, the process of computing relative tangent features $v_p(r)$ is equivariant under global rotations, and invariant under translations and scaling of $\mathbb{R}^3$. Namely, if $R \in \mathrm{SO}(3)$ is a rotation, $x \in \mathbb{R}^3$ is a translation vector, and $\lambda > 0$ is a scaling factor, then*

$$v_{Rp}^{R\mathcal{M}} = R(v_p^{\mathcal{M}}), \quad v_{\lambda p}^{\lambda\mathcal{M}} = v_p^{\mathcal{M}}, \quad v_{p+x}^{\mathcal{M}+x} = v_p^{\mathcal{M}}.$$

See Appendix B.2 for a proof of this result. Thanks to their geometric properties, relative tangent features constitute a simple yet effective alternative to using raw node positions as input features.

5 Verifiably Equivariant Message Passing

In this section we empirically verify that relative tangent features, coupled with a suitable choice of bias for the convolutional layers, allow us to build models that are in fact equi/invariant to all the transformations mentioned in Section 2.4. We highlight the effect that "small" design choices, such as biases, can have when trying to integrate them towards building a fully equivariant pipeline. We also emphasize on the importance

of performing a thorough evaluation of the model by applying the transformations of interest to unseen inputs in order to reliably verify the promised equivariance properties.

Designing equivariant biases. The traditional way of including biases in standard convolutional layers involves the addition of a fixed vector across the different channels of the output tensor. However, in the context of mesh data, this procedure is *not* equivariant to changes of gauge. When considering geometric features, the addition of this "fixed" bias vector would correspond to summing a gauge-sensitive quantity to the coordinate vectors representing a feature.

As a response to this, we consider ***angular biases*** that respect gauge equivariance. Given a general representation ρ, we decompose it in its irreducible components $\{\rho_n\}$. The cumulative bias for ρ is assembled from the biases on its irreducible components. If $n = 0$, we may add a simple (non-angular) bias b to the 1-dimensional scalar feature f. For $n > 0$, we instead rotate the 2-dimensional coordinate vector f_p by an angular bias b, or equivalently we consider $\rho_n(b)f_p$ (c.f. Fig. 3). Therefore, for a feature of type ρ, the number of involved biases is the same as the number of irreducible components in ρ. This choice of bias is therefore suitable for our goal of designing a fully equivariant model.

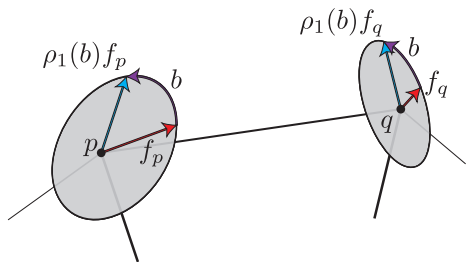

Figure 3: Angular bias applied to the tangential features in the convolution blocks. Original features are coloured in red, bias in purple, new features in blue. In order to preserve equivariance, features of higher type are instead rotated by an angular bias.

Verifying model equivariance. Table 1 compares the violation of equivariance to global/local transformations exhibited by randomly initialized models. We consider a SpiralNet++ (Gong et al., 2019) using raw positions as inputs; and a GEM-CNN (de Haan et al., 2021) using raw positions or relative tangent features as inputs, as well as the non-equivariant and angular biases mentioned above. Since we are interested in obtaining a model that is equivariant to arbitrary gauge changes, and not only to multiples of a given rotation, we do not employ (quantized) regular non-linearities (de Haan et al., 2021, §4).

Table 1: Equivariance error of randomly initialized models for various transformations on FAUST meshes. The error is computed as the MSE between the logits for the outputs corresponding to the *same* mesh with or without a given transformation. The mean is computed across all test meshes. Entries with small (resp. high) error are shown in blue (resp. red). Values in this table should be compared within the same column, based on the order of magnitude of the realized errors.

Model	Init. Feat.	Bias	Gauge	Rot-Tr-Scale	Perm
SpiralNet++	XYZ	—	0.0	$7.5 \cdot 10^{-2}$	$1.5 \cdot 10^{-15}$
GEM-CNN	XYZ	Non-Eq	$1.4 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$	$2.18 \cdot 10^{-13}$
		Angular	$7.9 \cdot 10^{-6}$	1.2	$1.5 \cdot 10^{-13}$
	RelTan	Angular	$1.3 \cdot 10^{-5}$	$3.63 \cdot 10^{-9}$	$1.85 \cdot 10^{-13}$

A GEM-CNN model with relative tangent features and angular bias is the only configuration that achieves equivariance to *all* the considered transformations. Note that SpiralNet++ is perfectly gauge equivariant as it only employs scalar features. The larger magnitude observed for the minimum error achieved for gauge transformations is explained by the fact that gauge transformations affect *every* convolutional layer of the model, causing numerical errors to accumulate.

Table 2: Accuracy for models trained on FAUST with various transformations on the test set. The blue entries show robustness to transformations for each column, whereas the red entries correspond to poor performance.

Model	Init. Feat.	Bias	RelPower	Train	Test	Gauge	Rot-Tr-Scale	Perm
SpiralNet++	XYZ	—	—	99.98	99.83	99.83	0.23	99.83
GEM-CNN	XYZ	Non-Eq	—	99.99	99.90	0.06	11.55	99.90
		Angular	—	99.45	97.97	96.85	0.16	97.97
	RelTan	Angular	[0.7]	99.68	98.69	98.20	98.69	98.69
			[0.5, 0.7]	99.63	98.36	97.84	98.36	98.36

We do not include accuracies for the SpiralNet++ model with relative tangent features. However, one finds that this model with relative tangent features is invariant to translations and scalings, as relative tangent features are invariant to these transformations. Nonetheless, the SpiralNet++ model with relative tangent features is not global rotation equivariant, since its layers are not designed to be compatible with rotations of the ambient space $\mathbb{R}^3$. We included a short note on node permutation equivariance of the SpiralNet++ model in Appendix C.

We complete this section by studying the robustness of a similar set of models *after training* on the FAUST dataset. Experimental details can be found in Appendix D. Note that we do not apply any transformations to meshes in the training set. The results are displayed in Table 2.

Note that the pattern of robustness to these transformations carries out verbatim from Table 1 to Table 2. However, the shortcomings of the non-equivariant models **cannot be detected** by looking *only* at the un-transformed training and test set accuracies.

Therefore, when evaluating equivariance-aimed models, applying the transformations of interest is crucial for successfully validating whether a model indeed satisfies the desired equivariance properties.

6 Equivariant Mesh Attention Layer

In this section we present the details of the Equivariant Mesh Attention layer. We start by considering an auxiliary representation $\rho_{\text{att}}\colon \mathrm{SO}(2) \to \mathbb{R}^{C_{\text{att}}}$. Once fixed a point p on the mesh, for any $q \in \mathcal{N}_p$, we consider the quantities

$$\mathbf{Q}_p = K_{\text{query}} f_p, \qquad \mathbf{K}_{pq} = K_{\text{key}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q, \qquad \mathbf{V}_{pq} = K_{\text{value}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q. \tag{8}$$

Here, K_{query} is a $C_{\text{att}} \times C_{\text{in}}$ matrix, $K_{\text{key}}(\theta)$ is a family of $C_{\text{att}} \times C_{\text{in}}$ matrices, while $K_{\text{value}}(\theta)$ is a family of $C_{\text{out}} \times C_{\text{in}}$ matrices. We require these matrices to satisfy, for all $g \in \mathrm{SO}(2)$, the constraints

$$\begin{gathered} K_{\text{query}} = \rho_{\text{att}}(-g)\ K_{\text{query}}\ \rho_{\text{in}}(g), \\ K_{\text{key}}(\theta - g) = \rho_{\text{att}}(-g)\ K_{\text{key}}(\theta)\ \rho_{\text{in}}(g), \quad K_{\text{value}}(\theta - g) = \rho_{\text{out}}(-g)\ K_{\text{value}}(\theta)\ \rho_{\text{in}}(g), \end{gathered} \tag{9}$$

in order to provide gauge equivariance. The solutions to these equations are provided in Appendix A.

We define $\mathbf{K}_p$ as the $C_{\text{att}} \times N_p$ matrix obtained by concatenating as columns the vectors $\mathbf{K}_{pq}$, as q varies; similarly, $\mathbf{V}_p$ is the $C_{\text{out}} \times N_p$ matrix obtained via the same procedure from the column vectors $\mathbf{V}_{pq}$, as q varies:

$$\mathbf{K}_p = \mathrm{Concat}(K_{\text{key}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q \text{ for } q \in \mathcal{N}_p), \qquad \mathbf{V}_p = \mathrm{Concat}(K_{\text{value}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q \text{ for } q \in \mathcal{N}_p).$$

The outcome f'_p of dimension C_{out} is then given by

$$\mathrm{Att}(\mathbf{Q}_p, \mathbf{K}_p, \mathbf{V}_p) = N_p \cdot \mathbf{V}_p \cdot \mathrm{softmax}\left(\frac{\mathbf{K}_p^\top \mathbf{Q}_p}{\sqrt{C_{\text{att}}}}\right). \tag{10}$$

While the factor N_p is not present in Veličković et al. (2018), here it is introduced so that Eq. (10) precisely generalizes GEM-CNN convolution (with no self-contribution), in the case where the components of the softmax vector are all equal to $1/N_p$ (compare with discussion in Section 3). Algorithm 1 in Appendix G provides a concise description of the design of our attention mechanism.

Thanks to the constraints satisfied by the kernels $K_{\text{self}}, K_{\text{key}}(\theta)$, and $K_{\text{value}}(\theta)$, we obtain:

Lemma 6.1. *The convolutional update in Eq. (10) is gauge equivariant.*

Lemma 6.2. *With this choice of kernels, the Equivariant Mesh Attention convolutional update in Eq. (10) is equivariant to global rotations, and invariant to translations and scalings of the ambient space $\mathbb{R}^3$.*

Equi/in-variant convolutional steps, together with relative tangent features and angular biases, provide multiple equi/in-variances of the Equivariant Mesh Attention Network:

Theorem 6.3. *With initial relative tangent features, Equivariant Mesh Attention Networks are equivariant to node permutations, and invariant under global rotations, translations and scalings of the ambient space $\mathbb{R}^3$, and under arbitrary gauge changes.*

See Appendix G for a proof of these results. Our model can also incorporate multi-head attention. Details are provided in Appendix G.5. However, we did not notice an improvement in performance when integrating this factor in our implementation for the considered tasks. Results below consider single-head attention only.

7 Related Work

Equivariance on graphs and manifolds has been studied from numerous perspectives. Cohen et al. (2018; 2019b) propose novel gauge equivariant convolution techniques on manifolds such as spheres and platonic solids. Satorras et al. (2021) propose E(n)-equivariant graph neural network models, which are applied to equivariant normalizing flows by Köhler et al. (2020). Moreover, de Haan et al. (2021) point out a relation between different types of equivariances: their gauge equivariant model is also equivariant to the group of isometries of the given mesh. The equivariance with respect to this group of isometries has been extensively studied by Weiler et al. (2021); Cohen (2021). In contrast, our approach effectively combines equivariance with respect to gauge transformations *and* global transformations of $\mathbb{R}^3$.

Several works propose alternative forms of equivariant attention. Hutchinson et al. (2021) introduce an attention architecture that is equivariant to Lie group actions. SE(3)-Transformer Fuchs et al. (2020) is a roto-translation equivariant attention network, designed for point-cloud data and graphs, and not meshes. Wang et al. (2020) propose a self-supervised equivariant attention mechanism in image segmentation that generates more consistent class activation maps over rescaling. Romero & Cordonnier (2021) propose a general group-equivariant self-attention formulation for processing images.

Attention has been used to provide more expressive filters over the isotropic convolutions in GCN Kipf & Welling (2017). Veličković et al. (2018); Shi et al. (2021); Chamberlain et al. (2021) implement attention mechanisms to give anisotropic filters. Recent research work He et al. (2021) also considers the idea of gauge equivariant attention by employing input features that provide global rotation equivariance. In contrast, our proposed relative tangent features, by detecting the local geometry of the mesh, additionally provide invariance to scaling and translations. Moreover, He et al. (2021) provide a general and elegant framework that is nonetheless only approximately gauge equivariant. Instead, our perspective provides a simple gauge equivariant convolution with angle-dependent kernels that is equivariant to *arbitrary* angles in SO(2).

8 Experiments

We carry out experiments on the FAUST Bogo et al. (2014) and TOSCA Bronstein et al. (2008) datasets for segmentation and classification tasks, respectively. Details on our experimental settings can be found in Appendix D. All our experiments involving test accuracy report test results applying different transformations to the unseen meshes. No transformations are applied to the training data. Since we are interested in obtaining a model that is equivariant to arbitrary gauge changes, and not only to multiples of a given rotation,

Table 3: Means (and standard deviations) of the segmentation accuracy on the FAUST dataset. Results comprise 5 random seeds. No transformations (i.e. data augmentation) are applied during training. Last 4 columns represent performance on the test set under different transformations. All models use angular biases, which combined with RelTan features make the models robust to listed transformations.

Model (Init. Feat.)	**Rel. Pow.**	**Train**	**Test**	**Gauge**	**Rot-Tr-Scale**	**Perm**
GEM-CNN (XYZ)	—	99.42 (0.15)	97.92 (0.30)	96.90 (0.25)	**02.14** (1.49)	97.92 (0.30)
GEM-CNN (RelTan)	[0.7]	99.69 (0.05)	98.62 (0.06)	98.04 (0.12)	98.62 (0.06)	98.62 (0.06)
	[0.5, 0.7]	**99.70** (0.09)	98.64 (0.22)	97.99 (0.18)	98.64 (0.22)	98.64 (0.22)
EMAN (XYZ)	—	99.62 (0.09)	98.46 (0.15)	97.26 (0.34)	**00.02** (0.00)	98.46 (0.15)
EMAN (RelTan)	[0.7]	99.27 (1.01)	98.13 (1.19)	97.44 (1.26)	98.13 (1.19)	98.13 (1.19)
	[0.5, 0.7]	99.68 (0.00)	**98.66** (0.07)	**98.41** (0.25)	**98.66** (0.07)	**98.66** (0.07)

we do not employ (quantized) regular non-linearities (de Haan et al., 2021, §4). In Appendix E we provide performance comparison between non-equivariant models with data augmentation and equivariant models; and in Appendix F we provide time-complexity comparison between GEM-CNN and EMAN.

8.1 Segmentation

The FAUST dataset consists of 100 3D human meshes with 6890 vertices, which is divided into 80 training and 20 test meshes. Nodes in each mesh are numbered so that nodes corresponding to the same location of the human mesh are labelled with the same number.

Table 3 illustrates the performance of various gauge equivariant models on FAUST. We compare models with and without attention, as well as using or not relative tangent features. All models in this section use angular bias. Note that models employing raw positions as input (marked XYZ) fail to be equivariant to rotations, translations and scaling transforms (Rot-Tr-Scale). This challenge is reliably overcome by the use of relative tangent features.

For models using relative tangent features, we consider two choices of relative powers. Relative tangent features allow us to choose a different relative power to be used for each of the channels in the input feature (see Section 4). This increases the expressivity of the model as different relative powers induce a processing of the local geometry at a point in the mesh. We find that using multiple channels with different relative powers translates into improvement in (transformed) test performance.

8.2 Classification

TOSCA consists of meshes belonging to nine different classes such as cats, men, women, centaurs, etc. While figures in each class are similarly meshed, each class has a varying number of nodes and edges. The dataset consists of 80 meshes, which we uniformly split into a train set of 64 meshes and a test set of 16 meshes respectively.

We do not apply permutations to the test meshes since the different number of nodes across classes made the implementation cumbersome. In addition to our theoretical guarantees, we do not expect node permutations to significantly affect the behavior of the models for a shape *classification* task given our use of a mean pooling layer for aggregating the information across the nodes.

We find that when using raw positions, GEM-CNNs outperform EMANs. This seems to point to a higher sensitivity of EMANs to un-normalized data. This sensitivity is not present when using relative tangent features. In fact, although both GEM-CNNs and EMANs (with relative powers $[0.5, 0.7]$) achieve perfect training and (transformed) test performance, the behavior for EMANs is more robust (lower variance) than that of the best-performing GEM-CNNs.

Table 4: Means (and standard deviations) of the classification accuracy on the TOSCA dataset. Results comprise 5 random seeds. No transformations (i.e., data augmentation) are applied during training. Last 4 columns represent performance on the test set under different transformations. All models use angular biases, which combined with RelTan features make the models robust to listed transformations.

Model (Init. Feat.)	**Rel. Pow.**	**Train**	**Test**	**Gauge**	**Rot-Tr-Scale**
GEM-CNN (XYZ)	—	93.12 (6.4)	88.24 (10.1)	88.24 (10.1)	**9.8** (3.4)
GEM-CNN (RelTan)	[0.7][a]	90.48 (0.00)	94.12 (0.00)	**94.12** (0.00)	**94.12** (0.00)
	[0.5, 0.7]	92.06 (13.7)	90.20 (16.9)	90.20 (16.9)	90.20 (16.9)
EMAN (XYZ)	—	61.38 (6.6)	52.94 (5.8)	64.71 (5.8)	**29.41** (10.1)
EMAN (RelTan)	[0.7]	85.19 (18.0)	88.24 (15.5)	90.2 (12.2)	90.2 (12.2)
	[0.5, 0.7]	**95.77** (7.3)	**96.08** (6.9)	**94.12** (10.1)	**94.12** (10.1)

[a]The zero standard deviation in this row is purely coincidental. We manually verified the different behavior for the random seeds used in this experiment. The performance at the end of training happens to match overlap for all runs.

We do not normalize the XYZ data in order to emphasize the fact that finding a good normalization technique becomes unnecessary when using relative tangent features, in addition to their scaling-invariance properties.

9 Conclusion

In this work, we propose Equivariant Mesh Attention Networks (EMAN), an attention-based model that is equi/invariant to node permutations, local gauge transformations, as well as global transformations such as rotations, translations, scalings. Our model consists of two major components: relative tangent features as input types and a message passing algorithm based on a gauge equivariant attention mechanism. We also emphasize the importance of rigorous testing of the overall assembled model, since small design choices – such as biases – can result in an non-equivariant model, damaging its robustness transformations in the data. We verify the equi/invariance of our overall model theoretically and empirically. EMANs achieve competitive performance on the FAUST and TOSCA datasets while maintaining equivariance to all the aforementioned transformations.

References

Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and Evaluation for 3D Mesh Registration. In *CVPR*, 2014.

Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Science & Business Media, 2008.

Michael Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478*, 2021.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs. *arXiv:1312.6203*, 2013.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph Neural Diffusion. In *ICML*, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *NeurIPS*, 2021.

Taco Cohen. *Equivariant Convolutional Networks*. PhD thesis, University of Amsterdam, 2021.

Taco Cohen and Max Welling. Group Equivariant Convolutional Networks. In *ICML*, 2016.

Taco Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *ICLR*, 2018.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In *ICML*, 2019a.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In *ICML*, 2019b.

Pim de Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge Equivariant Mesh CNNs: Anisotropic convolutions on geometric graphs. In *ICLR*, 2021.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *NeurIPS*, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. In *NeurIPS*, 2020.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *AISTATS*, 2011.

Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A Fast and Highly Efficient Mesh Convolution Operator. In *IEEE*, 2019.

Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge Equivariant Transformer. In *NeurIPS*, 2021.

Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant Self-Attention for Lie Groups. In *ICML*, 2021.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General Perception with Iterative Attention. In *ICML*, 2021.

Nicolas Keriven and Gabriel Peyré. Universal Invariant and Equivariant Graph Neural Networks. In *NeurIPS*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities. In *ICML*, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE*, 1998.

Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai. Deep Rotation Equivariant Network. *Neurocomputing*, 290:26–33, 2018.

David W Romero and Jean-Baptiste Cordonnier. Group Equivariant Stand-Alone Self-Attention For Vision. In *ICLR*, 2021.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n)-Equivariant Graph Neural Networks. In *ICML*, 2021.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *IJCAI*, 2021.

Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-Equivariant Steerable Networks. In *ICLR*, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*, 15(1):1929–1958, 2014.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor Field Networks: Rotation- and Translation-Equivariant Neural Networks for 3D Point Clouds. *arXiv:1802.08219*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation Equivariant CNNs for Digital Pathology. In *MICCAI*, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-Supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation. In *CVPR*, 2020.

Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *NeurIPS*, 2019.

Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning Steerable Filters for Rotation Equivariant CNNs. In *CVPR*, 2018.

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate Independent Convolutional Networks - Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds. *arXiv:2106.06020*, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In *NeurIPS*, 2017.

Appendix

A Geometric Kernel Constraints in GEM-CNNs

Table 5: Solutions to the angular kernel constraint for kernels that map from ρ_n to ρ_m, where $c_\pm = \cos((m \pm n)\theta)$ and $s_\pm = \sin((m \pm n)\theta)$. This table is directly taken from de Haan et al. (2021).

$\rho_{\text{in}} \to \rho_{\text{out}}$	linearly independent solutions for $K_{\text{neigh}}(\theta)$
$\rho_0 \to \rho_0$	(1)
$\rho_n \to \rho_0$	$(\cos n\theta \;\; \sin n\theta), \; (\sin n\theta \;\; -\cos n\theta)$
$\rho_0 \to \rho_m$	$\begin{pmatrix} \cos m\theta \\ \sin m\theta \end{pmatrix}, \begin{pmatrix} \sin m\theta \\ -\cos m\theta \end{pmatrix}$
$\rho_n \to \rho_m$	$\begin{pmatrix} c_- & -s_- \\ s_- & c_- \end{pmatrix}, \begin{pmatrix} s_- & c_- \\ -c_- & s_- \end{pmatrix}, \begin{pmatrix} c_+ & s_+ \\ s_+ & -c_+ \end{pmatrix}, \begin{pmatrix} -s_+ & c_+ \\ c_+ & s_+ \end{pmatrix}$
$\rho_{\text{in}} \to \rho_{\text{out}}$	linearly independent solutions for $K_{\text{self}}(\theta)$
$\rho_0 \to \rho_0$	(1)
$\rho_n \to \rho_n$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$

Message passing in GEM-CNNs de Haan et al. (2021) is defined by the equation

$$f'_p = K_{\text{self}} f_p + \sum_{q \in \mathcal{N}_p} K_{\text{neigh}}(\theta_{pq}) \rho_{\text{in}}(g_{q \to p}) f_q.$$

Gauge equivariance translate on the kernels K_{self} and $K_{\text{neigh}}(\theta)$ as

$$K_{\text{self}} = \rho_{\text{out}}(-g) \; K_{\text{self}} \; \rho_{\text{in}}(g), \quad K_{\text{neigh}}(\theta - g) = \rho_{\text{out}}(-g) \; K_{\text{neigh}}(\theta) \; \rho_{\text{in}}(g).$$

The representation ρ_{in} decomposes in irreducible components as $\oplus \rho_{n_j}$, and the representation ρ_{out} as $\oplus \rho_{m_i}$. Then, the kernels K_{self} and $K_{\text{neigh}}(\theta)$ are block matrices obtained by combining possible block kernels $(K_{\text{self}})_{ij}$ and $(K_{\text{neigh}}(\theta))_{ij}$ from features of type ρ_{n_j} to features of type ρ_{m_i}. Linearly independent solutions of the kernel constraint are listed in Table 5 (derived in Weiler & Cesa (2019)). A generic kernel from features of type ρ_n to feature of type ρ_m is a linear combination of learnable parameters of the given basis. Table 5 also provides solutions for the kernel equations in Section 6.

B More on Relative Tangent Features

B.1 The exponent $3/2$

We explain the choice of the exponent $3/2$ in the expression for relative tangent features. We rewrite the relative tangent feature v_p as

$$v_p = \frac{1}{N_p^{3/2}} \cdot \pi_p \left[\left(\sum_{q \in \mathcal{N}_p} \frac{(q-p)}{||q-p||^r} \right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1} \right) \right],$$

and we focus on the expression inside the square brackets. We view the vectors $q - p$, as $q \in \mathcal{N}_p$, as random vectors X_i, for $i = 1, \ldots, N$, where $N = N_p$ is the degree of p. The expression therefore becomes

$$\sum_{i=1}^{N} \frac{X_i}{||X_i||^r} \cdot \sum_{j=1}^{N} ||X_j||^{r-1}.$$

The expected value of the squared norm of this vector is

$$\mathbb{E}\left[\left\langle \sum_{i=1}^{N} \frac{X_i}{||X_i||^r} \cdot \sum_{j=1}^{N} ||X_j||^{r-1}, \sum_{i'=1}^{N} \frac{X_{i'}}{||X_{i'}||^r} \cdot \sum_{j'=1}^{N} ||X_{j'}||^{r-1} \right\rangle\right] = \sum_{i,j,i',j'=1}^{N} \mathbb{E}\left[\frac{||X_j||^{r-1}||X_{j'}||^{r-1}}{||X_i||^r||X_{i'}||^r} \langle X_i, X_{i'}\rangle\right].$$

It is reasonable to assume that the terms with $i \neq i'$ in the sum cancel out. For instance, all the terms with $i \neq i'$ are zero if the variables X_i are supposed to be independent and identically distributed, and their probability density function factors into radial and angular components, and the distribution of the angular component satisfies some symmetry conditions. The considered expected value then simplifies (only considering $i = i'$) as

$$\sum_{i,j,j'=1}^{N} \mathbb{E}\left[\frac{||X_j||^{r-1}||X_{j'}||^{r-1}}{||X_i||^{2(r-1)}}\right].$$

This sum presents N^3 addends. Notice that these terms are potentially different. However, if we assume that the variables are independent and identically distributed, at least $N^3 - 3N^2 + 2N$ terms – for which i, j, j' are all distinct – are the same. Therefore, the sum scales with a factor of N^3. We introduce a factor $1/N^{3/2}$ into the original vector, so that the considered expected value neither explodes nor vanishes as N becomes large.

B.2 Proof of Lemma 4.1

The relative tangent features to the mesh $\mathcal{M}$ are defined as (compare with Equation 4)

$$v_p^{\mathcal{M}} = \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{\pi_p^{\mathcal{M}}(q-p)}{||q-p||^r}\right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1}\right),$$

where we make explicit the given mesh $\mathcal{M}$.

Equivariance under global rotations of $\mathbb{R}^3$. Let $R \in \mathrm{SO}(3)$ be a global rotation in $\mathbb{R}^3$, and denote by $R\mathcal{M}$ the mesh obtained by rotating $\mathcal{M}$ according to R. Notice that the set of neighbors $q' \in \mathcal{N}_{Rp}$ of the node Rp in the mesh $R\mathcal{M}$ is the set of points Rq, as $q \in \mathcal{N}_p$ for the mesh $\mathcal{M}$. Also, the normal vector $n_{Rp}^{R\mathcal{M}}$ to the mesh $R\mathcal{M}$ at the node Rp is nothing but $Rn_p^{\mathcal{M}}$. Consequently,

$$\pi_{Rp}^{R\mathcal{M}} = I - n_{Rp}^{R\mathcal{M}} \left(n_{Rp}^{R\mathcal{M}}\right)^\top = I - Rn_p^{\mathcal{M}} \left(n_p^{\mathcal{M}}\right)^\top R^\top = R\pi_p^{\mathcal{M}} R^\top,$$

where we used that $RR^\top = I$. The relative tangent feature at node Rp for the mesh $R\mathcal{M}$ is

$$\begin{aligned} v_{Rp}^{R\mathcal{M}} &= \frac{1}{N_{Rp}^{3/2}} \cdot \left(\sum_{q' \in \mathcal{N}_{Rp}} \frac{\pi_{Rp}^{R\mathcal{M}}(q'-Rp)}{||q'-Rp||^r}\right) \cdot \left(\sum_{q' \in \mathcal{N}_{Rp}} ||q'-Rp||^{r-1}\right) \\ &= \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{R\pi_p^{\mathcal{M}} R^\top (R(q-p))}{|R(q-p)|^r}\right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||R(q-p)||^{r-1}\right) \\ &= R \cdot \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{\pi_p^{\mathcal{M}}(q-p)}{||q-p||^r}\right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1}\right) = Rv_p^{\mathcal{M}}, \end{aligned}$$

that is equivariance under global rotations.

Invariance under translations of $\mathbb{R}^3$. The argument is similar. If $x \in \mathbb{R}^3$ determines a translation, denote by $\mathcal{M} + x$ the translated mesh. The set of neighbors $q' \in \mathcal{N}_{p+x}$ of the node $p + x$ in the mesh $\mathcal{M} + x$ is the set of points $q + x$, as $q \in \mathcal{N}_p$ for the mesh $\mathcal{M}$. Also, the normal vector $n_{p+x}^{\mathcal{M}+x}$ to the mesh $\mathcal{M} + x$ at the

node $p+x$ is the original $n_p^{\mathcal{M}}$, and therefore $\pi_{p+x}^{\mathcal{M}+x} = \pi_p^{\mathcal{M}}$. Hence,

$$\begin{aligned} v_{p+x}^{\mathcal{M}+x} &= \frac{1}{N_{p+x}^{3/2}} \cdot \left(\sum_{q' \in \mathcal{N}_{p+x}} \frac{\pi_{p+x}^{\mathcal{M}+x}(q' - (p+x))}{||q' - (p+x)||^r} \right) \cdot \left(\sum_{q' \in \mathcal{N}_{p+x}} ||q' - (p+x)||^{r-1} \right) \\ &= \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{\pi_p^{\mathcal{M}}(q-p)}{||q-p||^r} \right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1} \right) = v_p^{\mathcal{M}}, \end{aligned}$$

that is invariance under translations.

Invariance under scaling of $\mathbb{R}^3$. Again, a similar argument. Let $\lambda > 0$ be the scaling factor, determining the map $p \mapsto \lambda p$ for $p \in \mathbb{R}^3$. The set of neighbors $q' \in \mathcal{N}_{\lambda p}$ of the node λp in the mesh $\lambda\mathcal{M}$ is the set of points λq, as $q \in \mathcal{N}_p$ for the mesh $\mathcal{M}$. As above, we deduce that $\pi_{\lambda p}^{\lambda\mathcal{M}} = \pi_p^{\mathcal{M}}$. Thus, as $\lambda > 0$,

$$\begin{aligned} v_{\lambda p}^{\lambda\mathcal{M}} &= \frac{1}{N_{\lambda p}^{3/2}} \cdot \left(\sum_{q' \in \mathcal{N}_{\lambda p}} \frac{\pi_{\lambda p}^{\lambda\mathcal{M}}(q' - (\lambda p))}{||q' - (\lambda p)||^r} \right) \cdot \left(\sum_{q' \in \mathcal{N}_{\lambda p}} ||q' - (\lambda p)||^{r-1} \right) \\ &= \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{\lambda \pi_p^{\mathcal{M}}(q-p)}{\lambda^r ||q-p||^r} \right) \cdot \left(\sum_{q \in \mathcal{N}_p} \lambda^{r-1} ||q-p||^{r-1} \right) \\ &= \frac{1}{N_p^{3/2}} \cdot \left(\sum_{q \in \mathcal{N}_p} \frac{\pi_p^{\mathcal{M}}(q-p)}{||q-p||^r} \right) \cdot \left(\sum_{q \in \mathcal{N}_p} ||q-p||^{r-1} \right) = v_p^{\mathcal{M}}, \end{aligned}$$

that is invariance under scaling.

C Node permutation equivariance of SpiralNet++

The SpiralNet++ convolution on meshes makes use of *spiral sequences* around nodes (Gong et al., 2019). Given a node, a spiral length, an orientation, and a preferred starting neighbor, the node sequence that constitutes the spiral is uniquely determined[†]. The authors consider a fixed counter-clockwise orientation for all spiral sequences, and the choice of starting neighbor is arbitrary. With these choices, let $S(i, \lambda)$ denote the indices of the nodes belonging to the spiral sequence of length λ starting at node i. The feature update follows the rule $x_i' = \texttt{MLP}\left(\|_{j \in S(i,\lambda)} x_j \right)$.

In Section 5, we analyze the response of SpiralNet++ to different types of transformations, including node permutation. This is equivalent to saying that the indices of the preferred neighbor choices transform according to the permutation. This may represent an issue for node permutation equivariance: since the choice of preferred neighbor is arbitrary, unless stored explicitly along with the mesh, it is not possible to guarantee that the "same" choice of starting neighbors will be made *after* having permuted the nodes. For example, consider a common mesh along with two different labelings A and B of its nodes, and an arbitrary choice of starting neighbors under labeling A. It is not possible to infer the *arbitrary* choice of starting neighbors under labeling B based only on the permutation relating the change of labeling $A \to B$.

This challenge can be easily resolved by taking into account the geometry of the mesh when choosing starting neighbors: e.g., choosing the closest neighbor in the Euclidean distance as preferred neighbor.

D Experimental Details

Inputs. The input feature type for GEM-CNN (XYZ) and EMAN (XYZ) is $3 \times \rho_0$, and for models using relative tangent features, GEM-CNN (RelTan) and EMAN (RelTan), it is $\rho_0 \oplus \rho_1$, where relative tangent features are stored in the ρ_1 part of the feature and the scalar ρ_0 is set to zero.

[†]See Section 3.2 of (Gong et al., 2019) for details on the construction of the spirals.

Models. For both segmentation and classification tasks, our models consist of two blocks: a convolution block and a dense block. The convolution block further consists of three sequential gauge equivariant residual blocks. Each residual block consists of two gauge equivariant convolutions followed by a summation of the input to the output of the block. The nature of message passing in these convolutions correspond to the choice of the model, e.g. GEM-CNN consists of convolutions of the form equation 6, whereas EMAN consists of attention mechanism equation 10.

For each model, the feature type of the input layer matches the feature type of the input features. The final layer of the sequence of residual blocks is of feature type $16 \times \rho_0$, i.e., 16 channels with only scalar features. All the intermediate feature types in the model are fixed to $16 \times (\rho_0 \oplus \rho_1 \oplus \rho_2)$. The second block consists of two dense layers. The first dense layer is of dimension 16×256 and the second dense layer maps to the target dimension followed by a softmax function. The output of the first layer is also passed through ReLU Glorot et al. (2011) and a dropout layer with parameter 0.5 Srivastava et al. (2014). The target dimension for the segmentation task on FAUST is 6890 and for classification task on TOSCA is 9. Further, in the case of classification, we also use mean pooling of the output of the first dense layer over the nodes.

Hyperparameters. We train using a learning rate of 0.01 for 100 epochs for segmentation tasks and a learning rate of 0.002 for 50 epochs for classification tasks respectively. We use Adam optimizer Kingma & Ba (2015) and negative log-likelihood loss function.

E Equivariance vs Data Augmentation

Equivariance in machine learning has arisen as a principled alternative to data-augmentation. Section 5.1 of (Thomas et al., 2018) shows that "rotation equivariance eliminates the need for rotational data augmentation" for point cloud data. Here, we show that the same applies to mesh data as well. To this end, we perform experiment on the FAUST dataset with data augmentation applied to both the training and test sets and compare the improvements brought by equivariance and data augmentation.

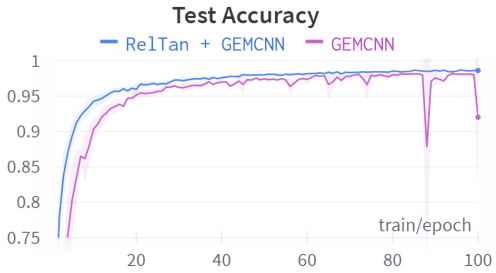

Figure 4: Effect of relative tangent features on test accuracy during training for a GEM-CNN model (averaged over 3 seeds). The results show better performance and lower variance for the model trained using relative tangent features.

Fig. 4 shows the accuracy (over 3 runs) for GEM-CNN models with initial XYZ and relative tangent features trained using roto-translation augmentations on the training set. This experiment confirms that *data augmentation* improves the generalization of non-equivariant model to unseen roto-translations: compare purple line above 90% with 1% test accuracy for GEM-CNN[XYZ] in Table 3.

Despite this improvement, *data augmentation is outperformed by equivariance*: the equivariant model (using relative tangent features) learns faster, has better final performance and lower variance during training.

F Time Comparison between EMAN and GEM-CNN

Here we provide a high-level time complexity analysis of EMAN compared to GEM-CNN. The bottleneck computations for both GEM-CNN and EMAN are expressions involving multiple matrix multiplications, of the type $K(\theta_{pq})\rho_{\text{in}}(g_{q\to p})f_q$, computed for every orientation of each edge $q \to p$. For EMAN, additionally,

attention coefficients are computed for every orientation of each edge. Computation of attention coefficients involve the same type of matrix multiplication as in GEM-CNN. Therefore, we observe an increased time to process features in EMAN than in GEM-CNN. Moreover, as both EMAM and GEM-CNN time complexities are proportional to the number of such operations involved in the models, the ratio of runtimes for EMAN and GEM-CNN scales as a constant in the limit of number of edges.

In practice, from Fig. 5 we find that EMAN takes twice as much time as GEM-CNN for the FAUST dataset. However, because of the use of attention mechanism, EMAN surpasses the performance of GEM-CNN within the 30 minutes that GEM-CNN takes to complete its 100 epochs. Hence, even though EMAN has higher time-complexity, it can outperform GEM-CNN within a short window of training time.

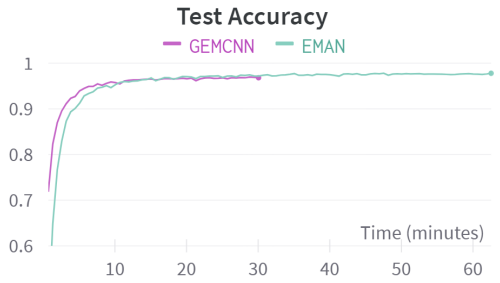

Figure 5: Wall-time comparison between GEM-CNN and EMAN. EMAN requires twice as long per epoch, due to the use of attention. However, given a fixed budget of 30 minutes, the accuracy for EMAN on the FAUST dataset surpasses that of GEM-CNN.

G Equivariant Mesh Attention Layer: algorithm, proofs, multi-head, and self-contribution

G.1 Equivariant Mesh Attention Layer algorithm

Algorithm 1 Equivariant Mesh Attention

Forward $((f_p)_{p\in\mathcal{M}},\ K_{\text{query}},\ K_{\text{key}}(\theta),\ K_{\text{value}}(\theta))$**:**
- **for** $p \in \mathcal{M}$**:**
  - $\mathbf{Q}_p \leftarrow K_{\text{query}} f_p$
  - $\mathbf{K}_p \leftarrow \texttt{cat}(K_{\text{key}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p})f_q \text{ for } q \in \mathcal{N}_p)$
  - $\mathbf{V}_p \leftarrow \texttt{cat}(K_{\text{value}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p})f_q \text{ for } q \in \mathcal{N}_p)$
  - $f'_p \leftarrow N_p \cdot \mathbf{V}_p \cdot \texttt{softmax}\left(\frac{\mathbf{K}_p^\top \mathbf{Q}_p}{\sqrt{C_{\text{att}}}}\right)$
- **Output:** $(f'_p)_{p\in\mathcal{M}}$

G.2 Proof of Lemma 6.1

The proof boils down to underlying two steps. First, the softmax-argument is invariant under gauge transformation. In addition, multiplying the matrix $\mathbf{V}_p$ (that transforms as a feature of type ρ_{out}) with the invariant softmax-vector produces a feature of type output. Here we provide a detailed proof.

Under a gauge transformation $g \in \text{SO}(2)$, the coordinate vectors f_p and $\rho_{\text{in}}(g_{q\to p})f_q$ at p transform as

$$f_p \mapsto \rho_{\text{in}}(-g)f_p, \quad \rho_{\text{in}}(g_{q\to p})f_q \mapsto \rho_{\text{in}}(-g)\rho_{\text{in}}(g_{q\to p})f_q.$$

Also, the angle θ changes as $\theta \mapsto \theta - g$ under the same gauge transformation. Using these relations, together with the ones expressed in Equation 9, we see that $\mathbf{Q}_p$ and the $\mathbf{K}_{pq}$ transform as features of type ρ_{att}, while

the $\mathbf{V}_{pq}$ transform as features of type ρ_{out}:

$$\mathbf{Q}_p \mapsto \rho_{\text{att}}(-g)\mathbf{Q}_p, \qquad \mathbf{K}_{pq} \mapsto \rho_{\text{att}}(-g)\mathbf{K}_{pq}, \qquad \mathbf{V}_{pq} \mapsto \rho_{\text{out}}(-g)\mathbf{V}_{pq}.$$

We see this explicitly, for instance, for the case of $\mathbf{K}_{pq}$:

$$\begin{aligned}
\mathbf{K}_{pq} &= K_{\text{key}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p})f_q \\
&\mapsto K_{\text{key}}(\theta_{pq} - g)\rho_{\text{in}}(g^{-1})\rho_{\text{in}}(g_{q\to p})f_q \\
&= \rho_{\text{att}}(-g)K_{\text{key}}(\theta)\rho_{\text{in}}(g)\rho_{\text{in}}(g^{-1})\rho_{\text{in}}(g_{q\to p})f_q \\
&= \rho_{\text{att}}(-g)K_{\text{key}}(\theta)\rho_{\text{in}}(g_{q\to p})f_q \\
&= \rho_{\text{att}}(-g)\mathbf{K}_{pq},
\end{aligned}$$

where we made use of the constraint $K_{\text{key}}(\theta - g) = \rho_{\text{att}}(-g)K_{\text{key}}(\theta)\rho_{\text{in}}(g)$. Computations for the other cases are similar. Being obtained by column-concatenation from $\mathbf{K}_{pq}$ and $\mathbf{V}_{pq}$, the matrices $\mathbf{K}_p$ and $\mathbf{V}_p$ undergo the same transformations as well:

$$\mathbf{K}_p \mapsto \rho_{\text{att}}(-g)\mathbf{K}_p, \quad \mathbf{V}_p \mapsto \rho_{\text{out}}(-g)\mathbf{V}_p.$$

Finally, the convolutional outcome transforms as

$$\begin{aligned}
f'_p &\mapsto \rho_{\text{out}}(-g) \cdot N_p \cdot \mathbf{V}_p \cdot \text{softmax}\left(\frac{\mathbf{K}_p^\top \rho_{\text{att}}(-g)^\top \rho_{\text{att}}(-g)\mathbf{Q}_p}{\sqrt{C_{\text{att}}}}\right) \\
&= \rho_{\text{out}}(-g) \cdot N_p \cdot \mathbf{V}_p \cdot \text{softmax}\left(\frac{\mathbf{K}_p^\top \cdot \mathbf{Q}_p}{\sqrt{C_{\text{att}}}}\right) \\
&= \rho_{\text{out}}(-g) \cdot f'_p,
\end{aligned}$$

where we used the orthogonality of the representation ρ_{att}, namely $\rho_{\text{att}}(-g)^\top = \rho_{\text{att}}(-g)^{-1}$. In conclusion, $f'_p = \text{Attention}(\mathbf{Q}_p, \mathbf{K}_p, \mathbf{V}_p)$ transforms as a feature of type ρ_{out}, and the proposed method is gauge equivariant.

G.3 Proof of Lemma 6.2

Suppose that $R \in \text{SO}(3)$ is a global rotation of $\mathbb{R}^3$, mapping the mesh $\mathcal{M}$ to the mesh $R\mathcal{M}$. Given a feature f of type ρ_{in} on $\mathcal{M}$, we represent it at a point p by its coordinates f_p, with respect to a gauge $\mathbf{E}_p$. Then, the rotation R defines a feature R_*f on $R\mathcal{M}$, with coordinates $(R_*f)_{Rp} = f_p$ with respect to the gauge $R\mathbf{E}_p$. Here comes the key remark: the quantities $\theta_{Rp,Rq}$ and $g_{Rq\to Rp}$ with respect to the gauge $R\mathbf{E}_p$ at Rp for $R\mathcal{M}$ are precisely the quantities θ_{pq} and $g_{q\to p}$ with respect to the gauge $\mathbf{E}_p$ at p for $\mathcal{M}$. Indeed, $g_{q\to p}$ can be computed as

$$g_{q\to p} = \text{atan2}\big((R_{q\to p}e_{q,2})^\top e_{p,1}, (R_{q\to p}e_{q,1})^\top e_{p,1})\big),$$

where $R_{q\to p}$ is the unique rotation of $\mathbb{R}^3$ that maps n_q to n_p, with fixed axis $n_q \times n_p$,

For θ_{pq}, instead, we notice that the angle can be written as

$$\theta_{pq} = \text{atan2}(e_{p,2}^\top \log_p(q), e_{p,1}^\top \log_p(q)),$$

where $\log_p$ is the norm-preserving discrete logarithmic map

$$\log_p(q) = ||q - p|| \frac{(I - n_p n_p^\top)(q - p)}{||(I - n_p n_p^\top)(q - p)||}.$$

Therefore, the outcome $(R * f)'_{Rp}$ of the convolution at Rp for $R\mathcal{M}$ is equal to the outcome of the convolution f'_p at p for $\mathcal{M}$. In other words, the feature mapping defined by the convolutional update is global rotation equivariant.

A similar argument can be applied to global translation T and scaling S equivariance: the coordinate vector of the feature do not change under T_* or S_*, the gauge is left unchanged, and the quantities θ_{pq} and $g_{q\to p}$ are not modified. In conclusion, the convolutional step is also translation and scaling invariant. Notice that a key property, implicitly used when considering scaling, is the dependence of the kernel only on angles, and not on the radial component.

G.4 Proof of Theorem 6.3

We prove the result for the designed Equivariant Mesh Attention models for segmentation and classification tasks, whose details are described in Appendix D.

Thanks to Lemma 6.1 and Lemma 6.2, each of the three blocks in the convolutional block is gauge and global rotation equivariant, and global translation and global scaling invariant. Therefore, the whole convolutional block satisfies the same properties, and it outputs a sum of scalar features. As operations in the dense block are defined on scalar features only, and not involving any quantities related to the geometry of the mesh, the same equi/in-variant properties hold also for the dense block. Finally, thanks to Lemma 4.1, the process of computing relative tangent features is consistent with the above properties, and the whole model is gauge invariant, global rotation equivariant, and global translation and global scaling invariant.

Regarding equivariance under permutation, it is enough to notice that all the operations involved in the convolutional block, and the process of computing relative tangent features, are permutation equivariant. Moreover, the operations in the dense block are defined node-wise, and the same operation on features is applied at each node. In conclusion, the whole model is equivariant under permutation (in the segmentation task, and invariant in the classification task).

G.5 Multi-head

It is feasible to incorporate multi-head attention in the Equivariant Mesh Attention layer, and we present here how. However, we did not notice an improvement in performance when integrating this factor in our implementation for the considered tasks.

Choose h and $d = d_{\text{model}}$ such that $C_{\text{out}} = dh$. For each $i = 1, \dots, h$, consider projection matrices of size $d \times C_{\text{att}}$, denoted by $W^i_{\text{query}}, W^i_{\text{key}}$, and a projection matrix of size $d \times C_{\text{out}}$, denoted be W^i_{value}. Also, for each $i = 1, \dots, h$ we fix a representation $\rho_i \colon \mathrm{SO}(2) \to \mathbb{R}^d$. For these matrices, we require the gauge equivariant conditions

$$W^i_{\text{query}} = \rho_i(-g) W^i_{\text{query}} \rho_{\text{att}}(g), \qquad W^i_{\text{key}} = \rho_i(-g) W^i_{\text{key}} \rho_{\text{att}}(g), \qquad W^i_{\text{value}} = \rho_i(-g) W^i_{\text{value}} \rho_{\text{out}}(g).$$

Finally, we consider a $C_{\text{out}} \times C_{\text{out}}$ matrix W^O. We define the representation $\rho_{\text{diag}} \colon \mathrm{SO}(2) \to \mathbb{R}^{C_{\text{out}}}$ by block-diagonal concatenation of the h representations ρ_i. The gauge equivariant condition satisfied by W^O is therefore

$$W^O = \rho_{\text{out}}(-g) W^O \rho_{\text{diag}}(g).$$

Then, the multihead attention outcome is defined by

$$\mathrm{MultiHead}(\mathbf{Q}_p, \mathbf{K}_p, \mathbf{V}_p) = W^O \cdot \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_h),$$

where

$$\mathrm{head}_i = \mathrm{Att}(W^i_{\text{query}} \mathbf{Q}_p, W^i_{\text{key}} \mathbf{K}_p, W^i_{\text{value}} \mathbf{V}_p).$$

G.6 Equivariant Mesh Attention Layer with Self-Contribution

Here we provide the details of the convolutional update variant including self-contribution (compare with Equation 7):

$$f'_p = \alpha_{pp} K_{\text{self}} f_p + \sum_{q \in \mathcal{N}_p} \alpha_{pq} K_{\text{neigh}}(\theta_{pq}) \rho_{\text{in}}(g_{q\to p}) f_q.$$

In line with Section 6, we consider the quantities

$$\mathbf{Q}_p = K_{\text{query}} f_p, \quad \mathbf{K}_{pp} = K_{\text{key}}^{\text{self}} f_p, \quad \mathbf{K}_{pq} = K_{\text{key}}^{\text{neigh}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q,$$
$$\mathbf{V}_{pp} = K_{\text{value}}^{\text{self}} f_p, \quad \mathbf{V}_{pq} = K_{\text{value}}^{\text{neigh}}(\theta_{pq})\rho_{\text{in}}(g_{q\to p}) f_q.$$

Here, K_{query}, $K_{\text{key}}^{\text{self}}$, and $K_{\text{key}}^{\text{neigh}}(\theta)$ are $C_{\text{att}} \times C_{\text{in}}$ matrices, while $K_{\text{value}}^{\text{self}}$ and $K_{\text{value}}^{\text{neigh}}(\theta)$ are $C_{\text{out}} \times C_{\text{in}}$ matrices. We define $\mathbf{K}_p$ as the $C_{\text{att}} \times (N_p + 1)$ matrix obtained by concatenating as columns the column vectors $\mathbf{K}_{pp}$ and $\mathbf{K}_{pq}$, as q varies. Similarly, $\mathbf{V}_p$ is the $C_{\text{out}} \times (N_p + 1)$ matrix obtained via the same procedure from the column vectors $\mathbf{V}_{pp}$ and $\mathbf{V}_{pq}$, as q varies. The outcome

$$\text{Att}(\mathbf{Q}_p, \mathbf{K}_p, \mathbf{V}_p) = (N_p + 1) \cdot \mathbf{V}_p \cdot \text{softmax}\left(\frac{\mathbf{K}_p^\top \cdot \mathbf{Q}_p}{\sqrt{C_{\text{att}}}}\right)$$

is a column vector of length C_{out}, and in fact a feature of type output.

Gauge equivariance for the matrices K_{query}, $K_{\text{key}}(\theta)$, and $K_{\text{value}}(\theta)$, translates as:

$$\begin{aligned}
K_{\text{query}} &= \rho_{\text{att}}(-g) K_{\text{query}} \rho_{\text{in}}(g), \\
K_{\text{key}}^{\text{self}} &= \rho_{\text{att}}(-g) K_{\text{key}}^{\text{self}} \rho_{\text{in}}(g), \\
K_{\text{key}}^{\text{neigh}}(\theta - g) &= \rho_{\text{att}}(-g) K_{\text{key}}^{\text{neigh}}(\theta) \rho_{\text{in}}(g), \\
K_{\text{value}}^{\text{self}} &= \rho_{\text{out}}(-g) K_{\text{value}}^{\text{self}} \rho_{\text{in}}(g), \\
K_{\text{value}}^{\text{neigh}}(\theta - g) &= \rho_{\text{out}}(-g) K_{\text{value}}^{\text{neigh}}(\theta) \rho_{\text{in}}(g).
\end{aligned}$$