Rotation Equivariant Graph Convolutional Network for Spherical Image Classification

Qin Yang[1], Chenglin Li[1], Wenrui Dai[1], Junni Zou[1], Guo-Jun Qi[2], Hongkai Xiong[1]
[1]Shanghai Jiao Tong University, [2]Futurewei Technologies
{yangqin, lcl1985, daiwenrui, zoujunni, xionghongkai}@sjtu.edu.cn, guojunq@gmail.com

Abstract

Convolutional neural networks (CNNs) designed for low-dimensional regular grids will unfortunately lead to non-optimal solutions for analyzing spherical images, due to their different geometrical properties from planar images. In this paper, we generalize the grid-based CNNs to a non-Euclidean space by taking into account the geometry of spherical surfaces and propose a Spherical Graph Convolutional Network (SGCN) to encode rotation equivariant representations. Specifically, we propose a spherical graph construction criterion showing that a graph needs to be regular by evenly covering the spherical surfaces in order to design a rotation equivariant graph convolutional layer. For the practical case where the perfectly regular graph does not exist, we design two quantitative measures to evaluate the degree of irregularity for a spherical graph. The Geodesic ICOsahedral Pixelation (GICOPix) is adopted to construct spherical graphs with the minimum degree of irregularity compared to the current popular pixelation schemes. In addition, we design a hierarchical pooling layer to keep the rotation-equivariance, followed by a transition layer to enforce the invariance to the rotations for spherical image classification. We evaluate the proposed graph convolutional layers with different pixelations schemes in terms of equivariance errors. We also assess the effectiveness of the proposed SGCN[1] in fulfilling rotation-invariance by the invariance error of the transition layers and recognizing the spherical images and 3D objects.

1. Introduction

Omnidirectional cameras generate spherical images with 360-degree view of the scenes that enable an immersive experience for users by freely adjusting their viewing orientations. Recently, omnidirectional cameras have become popular in virtual reality (VR) and augmented reality (AR) systems for applications ranging from robots [23, 27] to autonomous cars [15, 16], which results in an increasing demand for the analysis of spherical images. Convolutional neural networks (CNNs) have achieved significant improvement in analysis tasks related to planar images, e.g., image recognition [10], object detection [8], and image segmentation [9]. However, it is still challenging to generalize CNNs to analyzing spherical images defined on the non-Euclidean spheres, as distortions may be incurred when spherical images are projected onto a flat Euclidean surface to accommodate the grid-based architectures in CNNs [3].

CNNs commonly adapt to the non-Euclidean spherical images in two different ways. The first approach projects the spherical images into the planar format that can be processed directly by CNNs. Various projection methods have been studied, including the equirectangular projection (ERP) and the cube map projection [24], which lead to the inevitable projection distortions. For ERP, filter kernels are further designed for CNNs to compensate for the projection distortion [5, 26, 32]. [26] proposed to learn different kernels with variable size for each row in the projected images, however, the model size increases dramatically with the growth of image resolution. In [5, 32], the sampling location of filter kernel is changed to adapt to the distortion level. Without the guidance of rotation-equivariance, although model parameters could be reduced by sharing the kernels across all pixels, the model performance declines inevitably.

The other approaches [3, 7] extend CNNs to non-Euclidean domains to avoid the projection distortions. Although CNNs have strong capability to exploit the local translation equivariance and some works seek to capture various transformation equivariant representations of regular 2D images [20, 21, 30], they do not adapt to the 3D rotation of spherical images properly. Therefore, it is important to explore rotation-equivariance in spherical image analysis. [3] and [7] develop spherical CNNs by introducing the rotation-equivariant spherical cross-correlation in the spectral domain. However, Fourier transform is required for the

[1]Code is aviable at https://github.com/QinYang12/SGCN.
This work was supported in part by the NSFC under Grants 61931023, 61871267, 61972256, 61720106001, 61831018, and 91838303.

spherical correlation in each step, leading to high computational cost and significant memory overhead. [17] proposes a graph convolutional neural network for cosmological data that often come as spherical maps tailored by the Hierarchical Equal Area isoLatitude Pixelation (HEALPix). However, the irregular feature map in the HEALPix scheme sitll does not maintain the rotation-equivariance.

As an almost uniform discretization of the sphere, icosahedron [1] has been adopted to represent the spherical domain [4, 11, 14, 29]. In [4, 29], the spherical signal is projected on the icosahedron mesh with 20 basic planar regions, which is further analyzed with the gauge-equivariant and orientation-aware CNNs. The distortion is however still large, and discontinuities between the basic planar regions need to be handled by carefully designed schemes such as the gauge transformation on the features [4] and padding [29]. Based on geodesic icosahedron with smaller distortion, [14] designs the convolution and pooling kernels of CNNs and [11] presents parameterized differential operators on the unstructured grids. Although the convolution kernel is flipped by 180 degrees when applied to the next adjacent triangle [14], the convolution kernels in [11, 14] are still anisotropy and thus not rotation equivariant.

In this paper, we propose a Spherical Graph Convolutional Network (SGCN) to encode rotation-equivariance for spherical image analysis. Specially, we develop a graph convolutional layer through exploring the isometric transformation equivariance of the graph Chebyshev polynomial filters which is isotropy, a hierarchical pooling layer to exploit the multi-scale resolutions of the spherical images and keep the rotation-equivariance, and a transition layer to calculate the rotation-invariant statistics across multiple feature maps of the hierarchical pooling layer.

To enforce rotation-equivariance in the proposed polynomial graph convolutional layer, we propose a spherical graph construction criterion based on regularity, and show that given the number of vertices, a regular graph (i.e., vertices distribute uniformly on the surface of the spherical image) is equivariant to more rotations than an irregular one. For the practical case where the perfectly regular graph does not exist, we design two quantitative measures to evaluate the degree of irregularity for a spherical graph, and empirically show that a graph construction scheme with a lower degree of irregularity will result in smaller equivariance errors of the graph convolutional layers. Further the Geodesic ICOsahedral Pixelation (GICOPix) scheme is adopted to construct the spherical graph, which empirically demonstrates to achieve a lower degree of irregularity with the least weight variance for edges and least degree variance for vertices.

To demonstrate the effectiveness of the proposed criterion, we evaluate the equivariance errors of the graph convolutional layers by different graph construction schemes. We also assess the invariance errors of the proposed transition layers for the ability of capturing rotation-invariance and recognizing the spherical images. We further employ the proposed SGCN in spherical image classification, demonstrating that SGCN outperforms the state-of-the-art models on the Spherical MNIST (S-MNIST), Spherical CIFAR-10 (S-CIFAR-10) and achieve comparable performance to the 3D models on the ModelNet40 datasets in terms of rotation invariance classification accuracy.

2. Preliminaries

We represent a spherical image as an undirected and connected graph $G = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is a set of $|\mathcal{V}| = N$ vertices, $\mathcal{E}$ is a set of edges, and A is a weighted adjacency matrix with each element $a_{ij} = w(v_i, v_j)$ representing the connection weight between two vertices v_i and v_j. The weight a_{ij} is zero if vertices v_i and v_j are not connected. The normalized graph Laplacian is then defined as $L = I - D^{-1/2}AD^{-1/2}$, where $D \in \mathbb{R}^{N\times N}$ is a diagonal degree matrix with $D_{ii} = \sum_{j=1}^{N} a_{ij}$, and I is the identity matrix.

By recursively computing a Chebyshev polynomial to approximate the convolution kernel [6], the spectral convolution with a spherical signal x can be written as

$$y = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x, \tag{1}$$

where $\tilde{L} = 2L/\lambda_{max} - I$, λ_{max} is the largest eigenvalue of L, and θ_k denotes the Chebyshev polynomial coefficient which is a learnable parameter. Consequently, the Chebyshev polynomial $T_k(\tilde{L}) \in \mathbb{R}^{N\times N}$ can be recursively computed through $T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L})$ with $T_0 = I$ and $T_1 = \tilde{L}$. The spectral convolution with a K-th order polynomial is K-localized, i.e., the response of a vertex to the polynomial filter only depends on all the vertex values and edge weights on a path of length $k < K$.

It has been shown that a polynomial filter is equivariant to graph isometric transformations [13]. In the following, we give the definition of the graph isometric transformation and the graph isometric transformation equivariance.

Definition 1. Graph isometric transformation [13]. A graph isometric transformation g is a bijective mapping $g : \mathcal{V} \to \mathcal{V}$ that preserves the distance between two adjacent vertices on the graph. The corresponding transformation operator L_g makes a permutation of signal x by preserving their neighbourhoods. It can be formally depicted as

$$\forall v_k \in \mathcal{V}, \exists! v_j \in \mathcal{V} : [L_g x](v_k) = x(v_j), \tag{2}$$

where $L_g x$ is the transformed signal of x, and $\exists!$ indicates that there exists and only exists a vertex v_j corresponding to v_k.

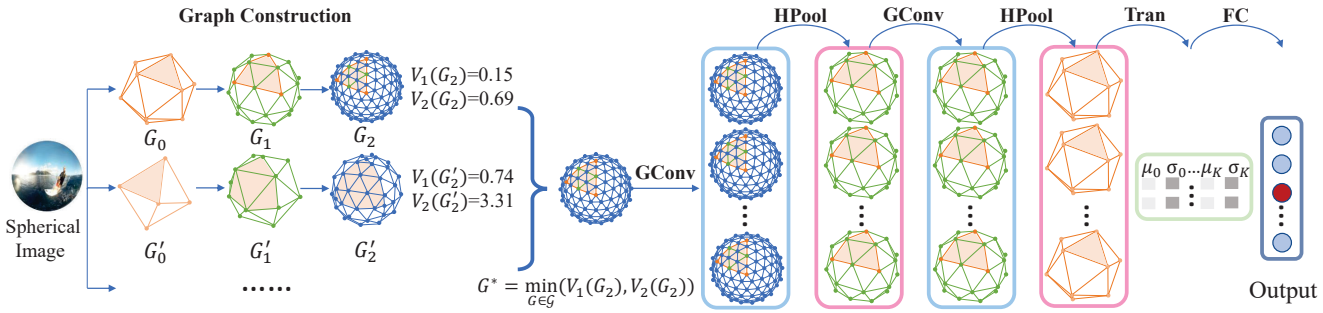


Figure 1. The proposed SGCN architecture with a two-level graph construction, which comprises two stacks of graph convolutional layer (GConv) and hierarchical pooling layer (HPool) followed by a transition layer (Tran) and a fully-connected layer (FC). The input spherical image is represented as a level-2 graph G_2^* based on GICOPix to achieve the rotation-equivariance. $[\mu_0, \sigma_0, ..., \mu_K, \sigma_K]$ is the multi-scale statistics across the feature maps. The output is the distribution over the classes of the datasets.

Definition 2. Graph isometric transformation equivariance. A graph convolutional layer is equivariant to a graph isometric transformation g, if transforming the input signal by the graph isometric transformation operator L_g and then feeding it through the graph convolution layer results in the same response as feeding the original signal through the graph convolutional layer followed by a corresponding transformation of the resultant feature maps, i.e.,

$$[\Phi(L_g x)](v) = [L_g(\Phi(x))](v), \tag{3}$$

where Φ represents the graph convolutional layer, and x is the input spherical signal.

We consider a unit sphere with the radius r set to 1 in this paper. Any point v on the sphere is then uniquely defined by its longitude θ and latitude ϕ, with $-\pi \leq \theta \leq \pi$ and $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$.

Definition 3. 3D rotation group. The spherical 3D rotation group is denoted by $SO(3)$, which is continuous. A rotation $R \in SO(3)$ is a rigid transformation preserving the Euclidean distance and orientation, which can be represented by a 3×3 matrix.

Since a vertex v on the rotated graph corresponds to the vertex $R^{-1}v$ on the original graph, we have

$$[L_R x](v) = x(R^{-1}v), \tag{4}$$

where L_R denotes the rotation transformation corresponding to R. In this paper, we will consider a rotation subgroup $\mathcal{R}$ of $SO(3)$ with a finite number of elements. If a graph convolution layer is equivariant to all the rotations $R \in \mathcal{R}$, i.e., $[\Phi(L_R x)](v) = [L_R \Phi(x)](v)$, it is then equivariant to the rotation group $\mathcal{R}$.

3. Rotation-Equivariant Spherical GCN

Fig. 1 depicts the proposed Spherical Graph Convolutional Network (SGCN) that encodes the graph rotation-equivariance for spherical image classification.

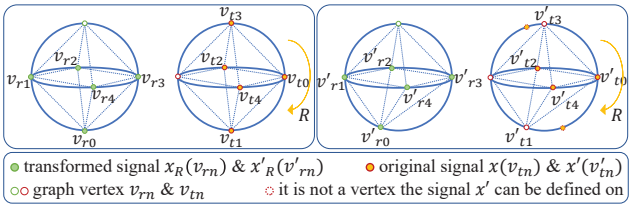
3.1. Regularity Constrained Graph Construction

As proved in [13], the polynomial filter in a graph convolution layer is equivariant to the graph isometric transformation. Therefore, the critical criterion here is to construct the graph with a rotation group that contains the largest possible number of graph isometric transformations. In the following, we show from two simple examples that given the number of vertices, such a graph can be constructed by ensuring its regularity.

Definition 4. Graph regularity. A spherical graph is regular if its vertices are distributed uniformly on the spherical surface. In detail, two principles are considered to define graph regularity: i) the distance between any two adjacent vertices is identical, and ii) all the vertices share the same number of neighbors.

Example 1: Regular graph with six vertices. In Fig. 2(a), we illustrate an example of regular spherical graph with 6 vertices, which is constructed based on a spherical octahedron with a rotation group $\mathcal{R}_o$ of order 24. Each rotation R in the octahedral rotation group $\mathcal{R}_o$ is an invertible mapping of vertices in the three-dimensional Euclidean space that preserves all the relevant structure of the spherical octahedron. For simplicity, we consider a 5-point image pattern x, where a vertex v_{t0} is connected with vertices $v_{t1} \sim v_{t4}$. After an exemplary rotation $R \in \mathcal{R}_o$, the image pattern x is transformed into $x_R = L_R x$. As illustrated in Fig. 2(a), for all the vertices $v_{r0} \sim v_{r4}$ in the rotated image x_R, there exists and only exists a set of vertices $v_{t0} \sim v_{t4}$ in the original image x satisfying $x_R(v_{rn}) = x(v_{tn}), \forall n = 0, 1, ..., 4$. The illustrated rotation R is therefore a graph isometric transformation. It can be further verified that all the 24 rotations in the octahedral rotation group $\mathcal{R}_o$ are graph isometric transformations.

Example 2: Irregular graph with six vertices. On the contrary, if the graph is constructed irregularly, its rotation group $\mathcal{R}'$ will have fewer elements than the octahedral rotation group $\mathcal{R}_o$. According to the definition, an irregular spherical graph may have different distances be-



(a) Regular Spherical Graph (b) Irregular Spherical Graph

Figure 2. Regular and irregular spherical graphs with six vertices, showing how graph regularity affects rotation-equivariance.

tween adjacent vertices or different number of neighbors for the vertices. In Fig. 2(b), we take an irregular graph with different distances between adjacent vertices for example. Given an arbitrary vertex v'_{r0} in the rotated image x'_R, there should be a vertex v'_{t0} in the original image x' that satisfies $x'_R(v'_{r0}) = x'(v'_{t0})$. Then, since the distance between two adjacent vertices differs from each other, for the neighbor vertices $v'_{r1} \sim v'_{r4}$ of the vertex v'_{r0} in x_R, there will no longer exist vertices $v'_{t1} \sim v'_{t4}$ in x' satisfying $x'_R(v'_{rn}) = x'(v'_{tn}), \forall n = 1, 2, ..., 4$.

The above two examples conclude that the vertices in a regular graph can be rotated exactly to all the other vertices to satisfy the definition of graph isometric transformation, while the vertices in an irregular graph fail to do so. In other words, for a given number of vertices, fewer graph isometric transformations will be contained in the rotation group of an irregular graph than a regular one. To accurately represent a high-resolution spherical image, the number of vertices N should be sufficiently large. A regular spherical graph constructed from the spherical polyhedron with only tens of vertices is far from enough. For a large number of finite vertices (e.g., thousands of vertices), however, it is impossible to construct a perfectly regular spherical graph. Therefore we design two quantitative measures to evaluate the degree of irregularity for a spherical graph: the variance of edge weights V_1 and the variance of neighbor numbers V_2, as

$$V_1 = \frac{1}{E}\sum_{i=1}^{E}(w_i - \mu_w)^2,\ V_2 = \frac{1}{N}\sum_{i=1}^{N}[n(v_i) - \mu_n]^2, \quad (5)$$

where $E = |\mathcal{E}|$ is the number of edges, w_i is the weight for the i-th edge, μ_w is the mean value of the edge weights; while $N = |\mathcal{V}|$ is the number of the vertices, $n(v_i)$ is the number of neighbors for each vertex v_i, and μ_n is the mean value of neighbor numbers. According to the graph regularity definition, we have $V_1 = V_2 = 0$ if the graph is regularly constructed. For irregular graphs, these two irregularity measures become larger than zero, and reveal the degree of irregularity of the constructed graph, which in turn determine the number of graph isometric transformations contained in the rotation group. Therefore, smaller values of V_1 and V_s are preferred in the spherical graph construction to preserve a higher degree of rotation equivariance of a graph convolutional layer.

3.2. Rotation-Equivariant Convolutional Layer

To reduce the degree of graph irregularity, we construct in this paper the spherical graph based on the Geodesic ICOsahedral Pixelation (GICOPix). Compared with other popular pixelation schemes, such a graph is isotropic to a large extent, where the cells are minimally distorted and almost equilateral. A fine-grained binary division can even further increase the resolution of the resultant graph. As will be seen in the experiments, GICOPix can outperform the other pixelations schemes in terms of irregularity measures, equivariance errors and invariance errors.

With GICOPix, the graph is constructed by repeatedly partitioning each equilateral triangle of a simpler geodesic icosahedron into four equilateral triangles, and then projecting the new vertices onto the sphere. All the vertices of the geodesic icosahedron become the graph vertices. Each vertex has six adjacent vertices except the twelve vertices of the original icosahedron that have five neighbors. We define the spherical graph constructed based on the original icosahedron as G_0^* with $N = 12$ vertices, and denote l as the subdivision level, i.e., the number of subdivision operation on the original icosahedron. Then, the graph constructed based on the first level ($l = 1$) geodesic icosahedron is G_1^* with $N = 42$ vertices. By induction, the graph based on the l-th level geodesic icosahedron becomes G_l^* with $N = 10 \times 2^{2l} + 2$ vertices. We show the constructed graph of GICOPix at levels 0, 2, 4 in bottom row of Fig. 4.

The original icosahedron has a symmetry rotation group called icosahedral rotation group $\mathcal{R}_i$ of order 60, which is a subgroup of $SO(3)$. The subdivision of each equilateral triangle is performed in the same fashion. Therefore, every rotation in $\mathcal{R}_i$ transforms a spherical graph G_l to itself and preserves all the relevant structure of that spherical graph, which is therefore graph isometric. Since a Chebyshev polynomial filter is equivariant to the graph isometric transformation [13], the graph convolutional layer is thus equivariant to the icosahedral rotation group $\mathcal{R}_i$. In this way, we construct a rotation-equivariant graph convolutional layer.

3.3. Rotation-Equivariant Pooling Layer

Multi-scale features with hierarchical representations of graphs need to be considered when generalizing CNNs to graphs. For this purpose, we propose a novel graph-coarsening scheme for the proposed pixelation process. The proposed pooling operator needs to be equivariant to the rotation, which is important for constructing the rotation-invariant classification architecture.

Specifically, we coarsen the l-th level spherical graph into the $(l-1)$-th level by maintaining the vertices of graph G_{l-1}^*, as illustrated in Fig. 3. We keep the hierarchical structure of the spherical graph without changing the under-

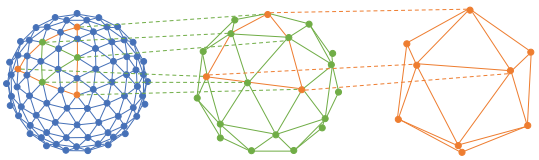


Figure 3. Illustration of the proposed rotation-equivariant pooling operator.

lying rotation group. Then we show that this can result in a rotation-equivariant pooling. Let us define $\mathcal{V}_l$ as the set of vertices of the l-th level spherical graph. Given a rotation R applied to l-th level spherical signal, we have $[L_R x](v_l) = x(R^{-1}v_l), \forall v_l \in \mathcal{V}_l$. The pooling layer P keeps a coarser spherical signal as $P(x)(v_{l-1}) = x(v_{l-1}), \forall v_{l-1} \in \mathcal{V}_{l-1}$. Rotating the l-th level spherical signal and then feeding it through the pooling layer results in $[P(L_R x)](v_{l-1}) = x((R^{-1}v)_{l-1}) = x(R^{-1}(v_{l-1})), \forall v_{l-1} \in \mathcal{V}_{l-1}$. And feeding the l-th level spherical signal through the pooling layer followed by rotating the resultant feature maps gives $[L_R P(x)](v_{l-1}) = x(R^{-1}(v_{l-1})), \forall v_{l-1} \in \mathcal{V}_{l-1}$. Thus, $[P(L_R x)](v_{l-1}) = [L_R P(x)](v_{l-1}), \forall v_{l-1} \in \mathcal{V}_{l-1}$, i.e., the pooling operation is equivariant to the rotation. Since the proposed convolutional layers are rotation equivariant at different levels, the stacks of the graph convolutional layers and the pooling layers also maintain rotation-equivariance.

In more detail, for all the feature maps $F_i^h, \forall i = 1, 2, ..., K_h$ of the h-th graph convolutional layer, we perform the same pooling operation. Assume that the input spherical graph is at level l_0. The h-th feature map is a spherical graph at level $l_0 - h$. For the graph pooling layer following the h-th graph convolutional layer, we keep all the vertices that belong to the spherical graph at the next level $l_0 - h - 1$, and the values on them are preserved.

3.4. Rotation-Invariant Transition Layer

By stacking graph convolutional layers and hierarchically pooling their results, we can fulfill the rotation-equivariance. After that, however, we would prefer to enforce the rotation invariance to performing classification tasks. In conventional CNNs, a stack of convolutional and pooling layers are followed by the fully-connected layers. However, the fully-connected layers are still spatial sensitive and not invariant to different rotations. Thus, a transition layer is required to extract the rotation-invariant features before the fully-connected layers.

To enforce rotation-invariance, a computationally efficient method [13] is used to perform gradient computation and back propagation. Specifically, we compute a set of graph-convolved signals $t_k = T_k(\tilde{L})x$ using Chebyshev polynomials $T_k(\tilde{L})$ of different order k with $k = 0, 1, ..., K$ for an input signal x. The resultant signals $t_k, k = 0, 1, ..., K$ correspond to the responses on multi-scale resolutions, all of which are equivariant to the rotation-equivariance. Then, we collect the mean μ_k and variance σ_k on each convolved feature map t_k across the vertices of the spherical graph, and output a concatenated feature vector $[\mu_0, \sigma_0, \mu_1, \sigma_1, ..., \mu_K, \sigma_K]$. The resultant features are invariant to the rotation since they are spatially agnostic to the responses of vertices in the spherical graph.

4. Experiments

In this section, we compare the equivariance errors of graph convolutional layers and the invariance errors of transition layers achieved by the proposed SGCN with the three pixelation schemes under various degrees of graph irregularity. We also evaluate the effectiveness of the proposed SGCN and compare it with the state-of-the-art methods in the spherical image classification tasks on the S-MNIST dataset and the S-CIFAR-10 dataset. To show the capability of the SGCN for real problems, we further demonstrate the performance comparison of 3D object classification tasks on ModelNet40 dataset. In addition, we conduct ablation studies on the roles of the hierarchical pooling layer and transition layer.

4.1. Degree of Irregularity

We compare the proposed GICOPix with the two other popular pixelation schemes for the degree of graph irregularity, i.e., the Generalized Sprial Set Pixelation (GSSPix) and the Hierarchical Equal Area isoLatitude Pixelation (HEALPix), in terms of the proposed two measures.

The GSSPix is an explicit construction of almost uniformly distributed points on a sphere [22]. For N points, the set cut the sphere with N horizontal planes with each latitude containing one point and the successive points having approximately the same distances. The HEALPix is commonly used for cosmological data with the properties that each pixel covers the same surface area as every other pixel. And 24 pixels at the corner of the rhombus connecting two rhombus of the base rhombic dodecahedron only have seven neighboring pixels.

We illustrate the graphs constructed based on the three pixelation schemes in Fig. 4, where the proposed GICOPix scheme results in a more regular spherical graph. We also calculate the weight variance V_1 and degree variance V_2 of the spherical graphs at three different levels $L = 0, 2, 4$. As shown in Table 1, the spherical graph based on the proposed GICOPix scheme has the smallest variance at all the levels. Especially, for the weight variance V_1, it is smaller with one order of magnitude, which suggests that the proposed GICOPix-based spherical graph is the most regular one.

4.2. Equivariance Error

To assess the proposed criterion of graph construction, we measure the equivariance error for the first spherical convolutional layer of the three implementations of the

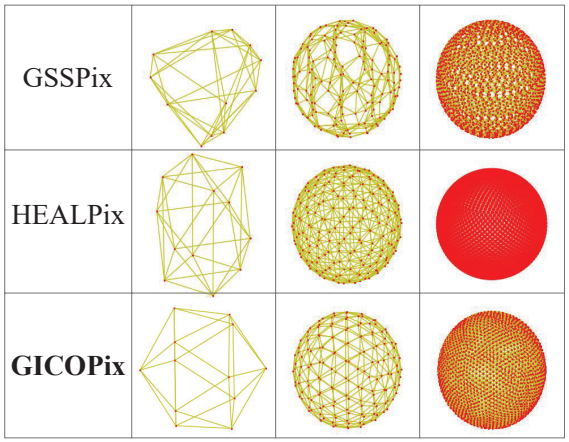


Figure 4. The spherical graph of level $L = 0, 2, 4$ with the three different pixelation schemes, i.e., GSSPix, HEALPix and the proposed GICOPix.

Table 1. Quantitative measurement of the degree of irregularity with the three different pixelation schemes. v_1 and v_2 are the variances of edge weights and vertex degrees, respectively.

Scheme	Level	# of vertices	V1(1e-2)	V2(1e-1)
GSSPix	0	12	2.867	2.500
	2	162	3.183	6.327
	4	2562	2.514	4.800
HEALPix	0	12	1.077	**0**
	2	192	1.296	1.094
	4	3072	1.465	0.078
GICOPix	0	12	**0**	**0**
	2	162	**0.149**	**0.686**
	4	2562	**0.154**	**0.047**

SGCN with three different graph construction schemes, which are denoted as the GICOPix-SGCN, GSSPix-SGCN, and HEALPix-SGCN. Their pooling layers are slightly different from each other since they depend on different pixelation schemes. Following the work of [3], we define the equivariance error as

$$\Delta = \frac{1}{n}\sum_{i=1}^{n} std(L_{R_i}\Phi(x_i) - \Phi(L_{R_i}x_i))/std(\Phi(x_i)). \tag{6}$$

We sample $n = 1000$ spherical images $x_i, i = 1, 2, ..., n$ with random 3D rotations R_i. Feeding each spherical image through the first graph convolutional layer results in 32 feature maps. By performing a graph isometric rotation on a completely regular graph, the equivariance error is expected to be zero. However, the constructed spherical graph is not ideally regular with random rotations from the continuous 3D rotation group $SO(3)$. As shown in Table 2, the proposed GICOPix-SGCN has the smallest equivariance error, and the SGCN based on other pixelation scheme has a larger equivariance error with higher graph irregularity. This indicates that the proposed principles are effective for modeling the rotation-equivariance.

Table 2. Equivariance error of the first spherical graph convolutional layer with the three pixelation schemes. The SGCN based on GICOPix has the smallest equivariance error.

Scheme	GSSPix	HEALPix	**GICOPix**
Error	0.942	0.434	**0.385**

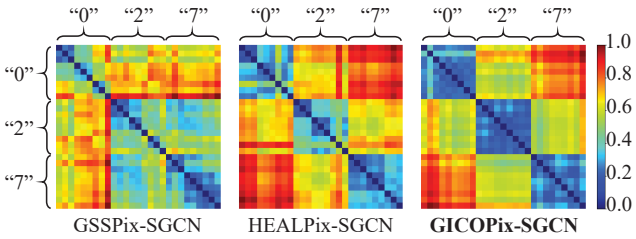


Figure 5. Illustration of the invariance error of the SGCN with the three different pixelation schemes, where the numbers besides the colorbar are the normalized Euclidean distances indicating the degree of dissimilarity of different features.

4.3. Invariance Error

We evaluate the invariance error of the transition layer to assess the ability of the proposed SGCN capturing the rotation-invariance. Specially, we select three different spherical images from the S-MNIST dataset, i.e, '0', '2', and '7', and project each image in nine different positions $\theta \in \{-1/8, 0, 1/8\}, \phi \in \{0, 1/8, 1/4\}$, totally yielding 27 spherical images. More details of spherical images creation will be introduced in the following section. By feeding these images into the SGCN, we obtain the features of the transition layer and evaluate the pairwise Euclidean distances in a $[27 \times 27]$ distance matrix. Ideally, the feature maps of different positions for the same spherical image should be identical, which means the three $[9 \times 9]$ diagonal sub-matrices of the $[27 \times 27]$ distance matrix should be zero (i.e., in blue in Fig. 5).

As illustrated in Fig. 5, the GICOPix-SGCN has more similar feature maps for different positions of the same spherical image than the GSSPix-SGCN and HEALPix-SGCN. This suggests the SGCN has smaller invariance error and thus can encode the rotation-equivariance and fulfill rotation-invariance better with a more regular pixelation scheme.

4.4. S-MNIST Classification

Dataset. The S-MNIST dataset is created by placing the digits on a plane tangent to the sphere at point (θ, ϕ) and projecting them on the spherical surfaces via the gnomonic projection [12, 5]. To evaluate the generalization performance of the proposed SGCN on the rotated images, we create two instances of this dataset: the non-rotated (NR) dataset and the rotated (R) ones. For the NR dataset, the tangent position is chosen randomly from a uniform distribution of spherical coordinates with the longitude $\theta \in$

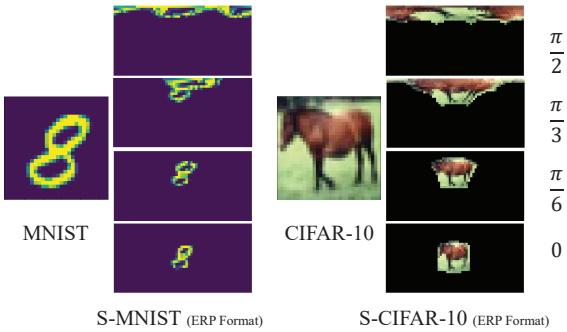


Figure 6. ERP format examples in the S-MNIST dataset and S-CIFAR-10 dataset with $\phi = \{0, \frac{\pi}{3}, \frac{\pi}{6}, \frac{\pi}{2}\}$. The distortion of the ERP format with $\phi = \frac{\pi}{2}$ is the largest.

$\{-\frac{\pi}{2}, \frac{\pi}{2}\}$, the latitude $\phi \in \{-\frac{\pi}{4}, \frac{\pi}{4}\}$. For the R dataset, $\theta \in \{-\pi, \pi\}$, $\phi \in \{-\frac{\pi}{2}, \frac{\pi}{2}\}$. We show the ERP of a spherical image from the S-MNIST dataset with $\phi = 0, \frac{\pi}{3}, \frac{\pi}{6}, \frac{\pi}{2}$ in Fig. 6.

Experiment setup. We benchmark our results against the S2CNNs [3], the GCNNs [12], the PDOs [11] and the SphereNet [5]. Except that the PDOs [11] have one Meshconv block, two ResBlocks, and an average pool layer and the transition layer of the SGCN has Chebyshev polynomials of the highest order 5, the architectures of all the models are the same. The network consists of two stacks of convolutional and pooling layers, followed by a fully-connected layer of ten neurons. The first stack has 32 filters, while the second has 64 filters. All the convolutional layer is followed by the ReLU activation. The order of the polynomial filter for the graph convolutional layer is set to 25. We input spherical signals at the level-4 resolution with 2562 pixels for GSSPix-SGCN and GICOPix-SGCN, 3072 pixels for HEALPix-SGCN. We train the models by the momentum optimizer with momentum 0.9 for 50 epochs with batch size 10. To avoid the overfitting, the batch normalization, weight decay rate $5e-4$ and dropout 0.9 are adopted. The initial learning rate is 0.02 and reduced to 0.002 after 33 epochs.

Result. The classification performances of different models on S-MNIST are compared in Table 3. We train each model on the rotated dataset and test on the rotated dataset as well (R/R). The GICOPix-SGCN outperform all the baselines except the Spherenet [5] based on the conventional CNNs. To evaluate the ability of models in encoding rotation-equivariance, we train the proposed model and Spherenet [5] on the non-rotated dataset and test on the rotated dataset (N/R). The Spherenet [5] performs much worse, while the performance of the proposed GICOPix-SGCN performs the best with only a slight decrease (Dec.) in performance compared to R/R.

We also compare the performances of the three different pixelation schemes with the same network, i.e., the GSSPix-SGCN, the HEALPix-SGCN, and the GICOPix-SGCN. In the R/R and N/R settings, the proposed GICOPix-SGCN outperforms the SGCNs based on the other two schemes with a significant performance gain, especially for the N/R setting. We attribute the success of the GICOPix-SGCN to the ability to explore rotation-equivariance. Besides, the SGCN based on a more regular spherical graph has a better performance on rotation-invariant classification for the S-MNIST dataset.

Table 3. Accuracy(%) of different models on the S-MNIST dataset. We achieve the comparable performance with less parameters in the R/R setting. In the N/R setting, the GICOPix-SGCN achieves the best performance. The performance under the "N/R" setting and "Dec" demonstrate the proposed model has stronger capability to capture rotation-equivariance.

Models	R/R	N/R	Dec.	Param.
GCNNs [12]	82.79	-	-	282K
S2CNNs [3]	88.14	-	-	149K
PDOs [11]	83.00	61.09	21.91	62 K
SphereNet [5]	**94.41**	55.18	39.22	196K
GSSPix-SGCN	74.41	43.26	31.15	58K
HEALPix-SGCN	92.36	91.41	0.95	58K
GICOPix-SGCN	93.58	**93.43**	**0.15**	58K

Table 4. Accuracy(%) of different models on the S-CIFAR-10 dataset. The GICOPix-SGCN achieves the best performance in both R/R and N/R settings. The performance under the "N/R" setting and "Dec" demonstrate the proposed model has stronger capability to capture the rotation-equivariance.

Models	R/R	N/R	Dec.	Param.
SphereNet [5]	53.90	37.18	16.72	196K
GSSPix-SGCN	47.51	38.85	8.66	58K
HEALPix-SGCN	55.08	51.90	3.18	58K
GICOPix-SGCN	**58.03**	**56.84**	**1.19**	58K

4.5. S-CIFAR-10 Classification

Dataset. The S-CIFAR-10 dataset contains more photo-realistic images than the S-MNIST dataset. We create the R and NR sets of the S-CIFAR-10 dataset in the same way as generating the S-MNIST dataset. The ERP of a spherical image from the S-CIFAR-10 dataset with $\phi = 0, \frac{\pi}{3}, \frac{\pi}{6}, \frac{\pi}{2}$ is shown in Fig. 6

Experiment setup. We adopt the SphereNet [5] as our baseline model. The network of the SGCN and implementation details are the same as the S-MNIST classification task except that the learning rate is reduced to 0.01 and 0.001.

Result. The performances of different models on the S-CIFAR-10 dataset are compared in Table 4. The proposed GICOPix-SGCN achieve the state-of-the-art performance in both R/R and N/R settings. Especially, for the N/R setting, the proposed GICOPix-SGCN has the smallest decrease in the performance. This suggests that the proposed SGCN based on a more regular spherical graph has a stronger ability to encode rotation-equivariance and fulfill rotation-invariance, and thus performs better for the rotation invariant classification on the S-CIFAR-10 dataset.

4.6. 3D Object Classification

Dataset. The ModelNet40 [28] dataset contains 40-class 3D models with 9843 training samples and 2468 testing samples. To apply the proposed SGCN in the 3D object classification task, we convert from the 3D geometries to signals on the sphere by following the method in [3, 11]. We project the 3D meshes onto a level-4 unit sphere with each mesh at the coordinate origin. First, we send a ray from the points on the sphere to the origin and record 3 channels information: the ray length from each point to the mesh and sin, cos of the surface angle. Further, we augment the signal with another 3 channels for the convex hull of the mesh, forming 6 channels of the signal in total. Following [7], for the NR dataset, the 3D objects are rotated with random azimuthal rotations. For the R dataset, the 3D objects are rotated with arbitrary rotations randomly.

Experiment Setup. We benchmark our results against the 3D model, i.e, the PointNet [18], the SubVolSup MO [19], the MVCNN 12x [25], the RICNN [31], the ClusterNet [2], and the spherical CNN model, i.e., the PDOs [11] and the SphericalCNN [7]. Compared to the network of our SGCN with two stacks of convolutional layers and plooing layers, all these baseline methods have a much more complex structure and a significantly enormous number of network parameters, such as 4 blocks with one meshconv block and three ResBlocks for the PDOs [11], and 8 spherical convolutional layers for the SphericalCNN [7]. The implementation details of the SGCN are the same as the S-MNIST classification task except that the networks are trained for 100 epochs with batch size 16. The initial learning rate is 0.01 and reduced to 0.001 after 50 epochs.

Result. The performances of different models on ModelNet40 dataset are compared in Table 5. Compared to the 3D models which have a much more complex structure, the GICOPix-SGCN achieves comparable performance in the R/R setting and N/R setting. Compared to the spehrical CNN models, the GICOPix-SGCN achieves comparable performance to the PDOs [11] and the SphericalCNN [7] in the R/R setting. The competing methods suffer a sharp drop in performance for the N/R setting with the unseen rotations presented, and the PDOs [11] perform no better than a random chance. In contrast, the GICOPix-SGCN is robust to this and still perform well. This indicates that our GICOPix-SGCN has strong applicability to real problems in 3D object classification. And the SGCN based on a more regular spherical graph has a stronger ability to explore rotation-equivariance and fulfill rotation-invariance.

4.7. Ablation Study

In Tables 3, 4 and 5, we have demonstrated the effectiveness of the proposed SGCN with a regular graph for rotation invariant classification, and the important role of the graph construction for modeling rotation-equivariance.

Table 5. Accuracy(%) of different models on the ModelNet40 dataset. The GICOPix-SGCN achieves the comparable performance to the 3D models in both R/R and N/R settings. Compared to the spherical CNN models, the GICOPix-SGCN achieves the best performance in the N/R setting. The performance under the "N/R" setting and "Dec" demonstrate the ability of the proposed model to capture rotation-equivariance.

Models	R/R	N/R	Dec.	Param.
PointNet [18]	83.6	14.7	68.9	3.5M
SubVolSup MO [19]	85.0	45.5	39.5	17M
MVCNN 12x [25]	77.6	70.1	7.5	99M
RICNN [31]	86.4	86.4	**0.0**	0.7M
ClusterNet [2]	87.1	**87.1**	**0.0**	1.4M
PDOs [11]	**89.8**	23.5	66.3	3.7M
SphericalCNN [7]	86.9	78.6	10.2	0.5M
GSSPix-SGCN	66.1	14.1	52.0	0.1M
HEALPix-SGCN	83.9	78.4	5.5	0.1M
GICOPix-SGCN	86.3	84.0	2.3	0.1M

Table 6. The effect of different components in GICOPix-SGCN on the S-MNIST and ModelNet40 dataset. "HPool" denotes the proposed hierarchical pooling scheme and "Tran" indicates the proposed transition layer.

HPool	Tran	S-MNIST		ModelNet40	
		R/R	N/R	R/R	N/R
✓	✓	93.58	**93.43**	**86.26**	**84.04**
✓		93.31	92.67	84.00	83.31
	✓	**93.91**	91.85	85.58	81.97

We also study here the impact of the two main components of the proposed SGCN, i.e., the hierarchical pooling layer and the transition layer. In Table 6, "HPool" denotes the proposed hierarchical pooling, and "Tran" denotes the proposed transition layer. We replace the proposed pooling layer by the common method as in [6] and replace the transition layer by the global average pooling. In Table 6, we can see that the transition layer has a gain of 0.76% for the S-MNIST classification and 0.73% for the ModelNet40 classification in the N/R setting. The hierarchical pooling has an additional 1.58% improvement for the S-MNIST classification and 2.07% improvement for the ModelNet40 in the N/R setting.

5. Conclusion

In this paper, we have presented the spherical graph convolutional network (SGCN) based on GICOPix to encode rotation-equivariance for spherical image analysis. We proposed a spherical graph construction criterion and constructed the spherical graph based on GICOPix with the minimum degree of irregularity. In addition, we designed the SGCN with the hierarchical pooling operator and transition layer. The experiments have demonstrated that the GICOPix-SGCN could encode the rotation-equivariance with a stronger ability to recognize the spherical images.

References

[1] John R Baumgardner and Paul O Frederickson. Icosahedral discretization of the two-sphere. *SIAM Journal on Numerical Analysis*, 22(6):1107–1115, 1985.

[2] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4994–5002, 2019.

[3] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018.

[4] Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. *arXiv preprint arXiv:1902.04615*, 2019.

[5] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision*, pages 518–533, Munich, Germany, September 2018.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, Barcelona, Spain, December 2016.

[7] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision*, pages 54–70, Munich, Germany, September 2018.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, Columbus, OH, USA, June 2014.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, Venice, Italy, October 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, USA, June 2016.

[11] Chiyu Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Niessner. Spherical CNNs on unstructured grids. In *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.

[12] Renata Khasanova and Pascal Frossard. Graph-based classification of omnidirectional images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 860–869, Venice, Italy, October 2017.

[13] Renata Khasanova and Pascal Frossard. Isometric transformation invariant graph-based deep neural network. *arXiv preprint arXiv:1808.07366*, 2018.

[14] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. SpherePHD: Applying CNNs on a spherical PolyHeDron representation of 360° images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9181–9189, Long Beach, CA, USA, June 2019.

[15] Shigang Li. Monitoring around a vehicle by a spherical image sensor. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):541–550, April 2006.

[16] Maxime Meilland, Andrew I. Comport, and Patrick Rives. A spherical robot-centered representation for urban navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5196–5201, Taipei, October 2010.

[17] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications. *Astronomy and Computing*, 27:130–146, April 2019.

[18] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[19] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, Las Vegas, NV, USA, June 2016.

[20] Guo-Jun Qi. Learning generalized transformation equivariant representations via autoencoding transformations. *arXiv preprint arXiv:1906.08628*, 2019.

[21] Guo-Jun Qi, Liheng Zhang, Chang Wen Chen, and Qi Tian. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8130–8139, 2019.

[22] Evguenii A. Rakhmanov, E. B. Saff, and Y. M. Zhou. Minimal discrete energy on the sphere. *Mathematical Research Letters*, 1(6):647–662, January 1994.

[23] Lingyan Ran, Yanning Zhang, Qilin Zhang, and Tao Yang. Convolutional neural network-based robot navigation using uncalibrated spherical images. *Sensors*, 17(6):1341, 2017.

[24] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, November 2018.

[25] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, Santiago, Chile, December 2015.

[26] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360° imagery. In *Advances*

in Neural Information Processing Systems, pages 529–539, Long Beach, CA, USA, December 2017.

[27] Guofeng Tong, Ran Liu, and Jindong Tan. 3D information retrieval in mobile robot vision based on spherical compound eye. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 1895–1900, Phuket, Thailand, December 2011.

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

[29] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3533–3541, 2019.

[30] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019.

[31] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3D point clouds deep learning. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 204–213. IEEE, 2019.

[32] Qiang Zhao, Chen Zhu, Feng Dai, Yike Ma, Guoqing Jin, and Yongdong Zhang. Distortion-aware CNNs for spherical images. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1198–1204, Stockholm, Sweden, July 2018.