Pattern Recognition Letters 202 (2026) 44–49

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Geometric algebra transformer with spectral-spatial-volumetric feature extraction for hyperspectral image classification

Jacopo Fabi, Claudio Schiavella *, Irene Amerini

Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome, Rome, 00185, Italy

ARTICLE INFO

Editor: Dr. Y. Liu

Keywords:
Hyperspectral image classification
Geometric algebra
Transformer

ABSTRACT

Current approaches to hyperspectral imaging classification have achieved important results, mostly depending on convolution-based preprocessing that adds complexity to the approach. At the same time, these methods often do not directly integrate all the data's spatial, spectral, and volumetric dependencies at once. This article introduces Hyperspectral Geometric Algebra Transformer (H-GATr), an approach that addresses these limitations by using a Geometric Algebra (GA) architecture to integrate hyperspectral information into multivector representations, without additional trainable parameters. Thus, exploiting the intrinsic geometric structure of the data is possible to obtain an efficient integration of all hyperspectral sample dependencies. Experiments carried out on various benchmark datasets demonstrate that H-GATr is capable of achieving performance comparable to and even superior to current models, offering a compact solution for remote sensing applications.

1. Introduction

Hyperspectral Imaging (HSI) enables the acquisition of full spectral information for each pixel of a scene, making it a valuable tool in remote sensing and geoscience. HSI has been used to monitor vegetation health [1], map mineral deposits [2], detect land cover changes [3], and assess environmental hazards [4]. The Hyperspectral Image Classification task consists of assigning a class label to each pixel in the image. Typically, to classify a pixel, a spatial window centered on that pixel is used in order to exploit both spectral and spatial dimensions [5].

Convolutional Neural Networks (CNNs), which have achieved important results in the literature [6–8], have explored different ways to simultaneously process these two dimensions. A straightforward approach involves the use of 3D convolutional filters, as in [9], where such filters are first applied along the spatial dimension and then along the spectral one, also employing residual connections to improve network depth. In contrast, the method proposed in [10] uses two separate blocks that extract spectral and spatial information in parallel through learnable morphological operations.

Transformers have also demonstrated their validity thanks to their attention mechanism. For instance, SAT-Net [11] first reduces the number of spectral bands through a spectral attention mechanism, and then captures both spatial and spectral relationships among patch-level representations using self-attention. Similarly, the HSI Transformer (HiT) [12] first extracts spatial and spectral features using parallel 3D convolutions, then applies a convolution-based feature mixing module to encode features along height, width, and spectral dimensions, while Transformer layers capture global relationships across patches. Thanks to their ability to capture both spatial and spectral dependencies, Transformer-based architectures currently achieve state-of-the-art performance in hyperspectral image classification [13,14]. However, these models cannot directly process raw hyperspectral data, which require preprocessing to generate suitable tokens. A common approach is to employ convolution based techniques to produce these tokens, enriching them with spectral and spatial information. While effective, this strategy increases the network's complexity and may overlook information that is already present in the data [15,16].

For this reason, we chose to explore the use of Geometric Algebra (GA), a mathematical framework that represents geometric entities as multivectors, simultaneously combining scalar, vector, and higher-order components. GA has proven highly effective in deep learning applications involving geometrically structured data [17]. GA-based architectures have already been applied to Hyperspectral Image Classification. For instance, the GA-CNN model [18] performs convolutions on multivectors generated from the spectral dimension. However, this approach presents two main limitations: the multivectors are created only from spectral information, neglecting spatial features, and the convolutions mainly capture spatial relationships, without fully exploiting spectral dependencies. Another relevant GA-based model, GASSF-NET [19], integrates LiDAR and hyperspectral data, but at the cost of increased network complexity.

* Corresponding author.
E-mail addresses: fabi.1809860@studenti.uniroma1.it (J. Fabi), schiavella@diag.uniroma1.it (C. Schiavella), amerini@diag.uniroma1.it (I. Amerini).

https://doi.org/10.1016/j.patrec.2026.02.001
Received 29 October 2025; Received in revised form 10 January 2026; Accepted 2 February 2026
Available online 4 February 2026

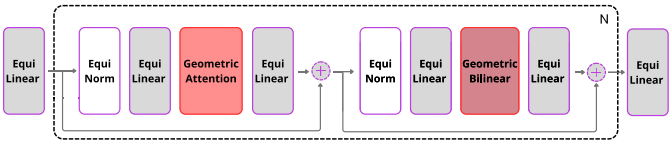

Fig. 1. Overview of the geometric algebra transformer.

Hyperspectral image classification poses unique challenges due to the high dimensionality of the spectral bands and the complex spatial-spectral relationships. Traditional approaches sometimes fail to fully capture interactions across spectral and spatial domains or to exploit relational information among local patches. To overcome these limitations, we designed the Hyperspectral Geometric Algebra Transformer (H-GATr), which expands the Geometric Algebra Transformer (GATr) architecture [20] to hyperspectral data. H-GATr introduces a multivector format to represent spatial, spectral, and volumetric features, which allows to embed hyperspectral patches directly into GA space and perform relational reasoning without extra parameters.

2. Preliminary

2.1. Geometric algebra

Geometric Algebra (GA), also known as Clifford Algebra, is a mathematical framework that can represent geometric objects and operations [21]. Its fundamental elements, called *multivectors*, encode magnitude, direction, and orientation across multiple dimensions. In general a multivectors can be seen as a sum of different geometric elements with different dimensionality (grade): scalars (grade 0), vectors e_i (grade 1), bivectors $e_i e_j$(grade 2) and pseudoscalar $e_1 \dots e_d$ (grade d).

In order to represent volumes, the GATr model uses the projective geometric algebra $\mathbb{G}_{3,0,1}$, where the number 3 indicates the count of basis vectors e_i satisfying $e_i^2 = 1$, and 1 indicates the number of basis vectors e_i satisfying $e_i^2 = 0$. A multivector x in this algebra can be written as shown in Eq. (1). It has $2^4 = 16$ components, corresponding to all possible products that can be formed from the four basis vectors e_0, e_1, e_2, e_3:

$$x = x_s + x_0 e_0 + x_1 e_1 + \cdots + x_{01} e_0 e_1 + \cdots + x_{0123} e_0 e_1 e_2 e_3 \tag{1}$$

The *Geometric Product* is the multiplication operation between multivectors in GA. For two vectors a and b, it is defined as in Eq. (2), combining the inner and wedge products:

$$ab = a \cdot b + a \wedge b, \tag{2}$$

where $a \cdot b$ is the standard inner (dot) product, representing the symmetric part, and $a \wedge b$ is the wedge (outer) product, representing the antisymmetric part.

This operation combines magnitude alignment (through the inner product) and oriented area (through the wedge product) into a single algebraic object, allowing multivectors to capture geometric relationships in a unified framework.

2.2. Geometric algebra transformer

The Geometric Algebra Transformer adopts an encoder-like architecture and is composed of N stacked blocks, as illustrated in Fig. 1. An initial equi-linear layer adjusts the hidden dimension, the number of channels expected for each input multivector, similarly to the embedding dimension in conventional Transformer models.

EquiLinear layers The equilinear layers apply the following trasformation:

$$\phi(x) = \sum_{k=0}^{d+1} w_k \langle x \rangle_k + \sum_{k=0}^{d} v_k e_0 \langle x \rangle_k \tag{3}$$

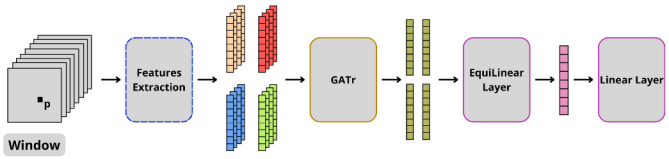

Fig. 2. Overview of the hyperspectral geometric algebra transformer.

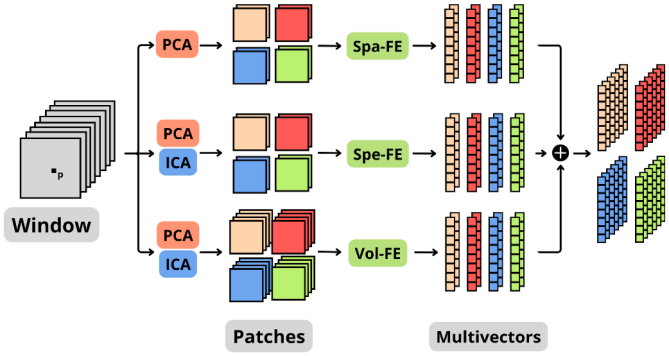

Fig. 3. Window-to-multivector feature transformation: The spectral bands are reduced by PCA, the window is split into patches for feature extraction, and the results are concatenated into the final multivector representation.

where x is the input multivector, $\langle x \rangle_k$ denotes the blade projection of the multivector, which sets all non-grade-k elements to zero. Here $w \in \mathbb{R}^{d+2}$, $v \in \mathbb{R}^{d+1}$, with $d = 3$.

Geometric bilinears layer A standard linear layer alone cannot build an expressive network. For this reason, the bilinear layer produces new feature representations through the geometric product and the join, the operation that identifies the smallest subspace containing both multivectors.

The layer takes as input two hidden states x and y, and a multivector z, computed as the mean of the network's input embeddings By concatenating the two operations, the layer produces:

$$\text{Bilinear}(x, y; z) = \text{Concatenate}(xy, \text{EquiJoin}(x, y; z)) \tag{4}$$

with $\text{EquiJoin} = z_{0123}(x^* \wedge y^*)^*$.

Geometric attention layer It extends the mechanism of traditional transformers to the multivector domain. Given queries q, key k and value v as multivectors, with n_i tokens and n_c channels, the equivariant multivector attention is defined as

$$Attention(q, k, v)_{i'c'} = \sum_i Softmax_i\left(\frac{\sum_c \langle q_{i'c}, k_{ic} \rangle}{\sqrt{8n_c}}\right) \tag{5}$$

Here i, i' denote token indices, while c and c' denote channel indices. The operator $\langle \cdot, \cdot \rangle$ represents the inner product.

3. Method

We propose the Hyperspectral GATr (H-GATr) which consists of three main components, as can be seen from the architecture's overview (Fig. 2). The first is the features extraction, which converts the input into multivectors. The second is the GATr which processes the multivectors and extracts relational features. Finally, the EquiLinear Layer and a Linear Layer map the output of GATr to the logits corresponding to the prediction classes.

3.1. Multivector features extraction

For each pixel x to classify, we consider a window $\mathcal{W}$ of size $H \times W \times D$, where D denotes the number of bands, and H and W are divisible by 4. The window is centered on the pixel at position $(\frac{H}{2}, \frac{W}{2})$. Our goal is to extract geometric features from this window based on the

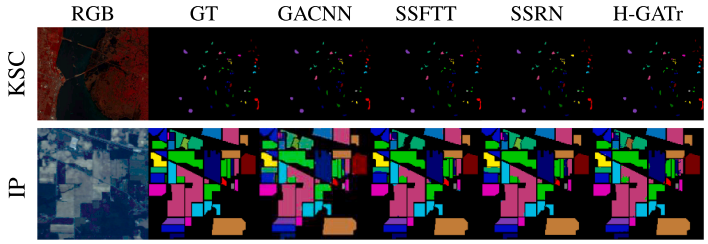

Fig. 4. Dataset samples of KSC and IP datasets and predictions from GACNN, SSFTT, SSRN, and H-GATr models.

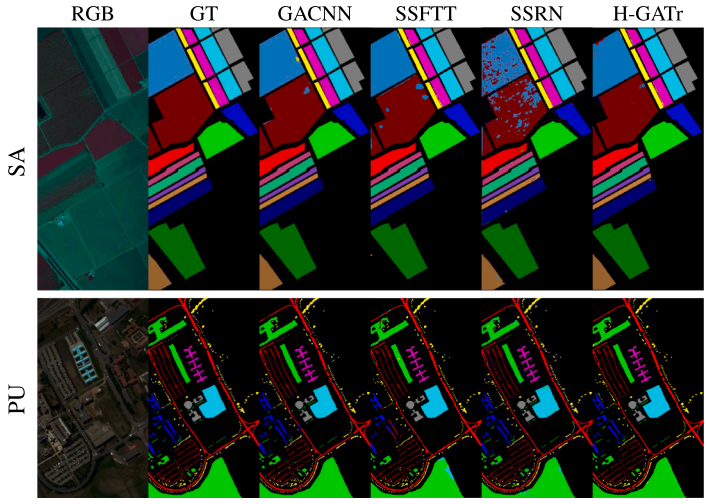

Fig. 5. Dataset samples of SA and PU datasets and predictions from GACNN, SSFTT, SSRN, and H-GATr models.

arrangement of the data, without relying on trainable weights. Since this window represents a three-dimensional data cube, it is crucial to preserve the information along these dimensions within the multivector representation. Consequently, first two types of features are extracted: the spatial features (Spa), which are obtained by independently processing each spectral band, and the spectral features (Spe), which are derived by independently analyzing the spectrum of each pixel. To complete the representation, we further extracted volumetric (Vol) features capable of describing the joint relationships between the spatial and spectral dimensions of the hyperspectral cube. To achieve this, we design the multivector feature extractor (see Fig. 3).

First three distinct representations of the window are generated: $\mathcal{W}^{\text{Spa}}$, $\mathcal{W}^{\text{Spe}}$, and $\mathcal{W}^{\text{Vol}}$. The first is obtained reducing the number of bands with the principal component analysis (PCA). The other two are generated applying PCA, followed by the independent component analysis (ICA). Using a different number of reduction channels D^{Spa}, D^{Spe}, and D^{Vol}, three representations with different values can be obtained. ICA is not applied to the spatial features, as it would make the values statistically independent, whereas spatial correlations need to be preserved.

The three windows $\mathcal{W}^{\text{Spa}}$, $\mathcal{W}^{\text{Spe}}$, and $\mathcal{W}^{\text{Vol}}$ are then divided into 4×4 patches. From these patches, are extracted three different features:

- **Spatial Feature Extractor (Spa-FE)**: each patch $\mathcal{P}^{\text{Spa}}$ has D^{Spa} bands with a size of 4×4. The 16 values of each band are assigned to the multivector in a left-to-right, top-to-bottom order. Since a multivector is generated for each band, this operation produces a total of D^{Spa} multivectors.
- **Spectral Feature Extractor (Spe-FE)**: a multivector is created from 16 pixels located at the same spatial position (i, j), but belonging to different spectral bands. Specifically, for a given position (i, j), a multivector is formed by taking, in order, the pixels at that position from the first 16 spectral bands.

  If the total number of spectral bands D^{Spe} is greater than 16, then the pixels in the subsequent bands (from the 17^{th} to the 32^{th}, and so on) form additional multivectors.

  Let D^{Spe} be divisible by 16; therefore, for each position (i, j), a total of $K = \frac{D^{\text{Spe}}}{16}$ multivectors are generated. Given that the patch has a size of 4×4, the total number of multivectors created is $K \cdot 16 = D^{\text{Spe}}$.
- **Volumetric Feature Extractor (Vol-FE)**: for every group of four consecutive spectral bands, two $2 \times 2 \times 2$ kernels are applied independently over the volumetric patches $\mathcal{P}^{\text{Vol}}$. One kernel performs max pooling, while the other performs min pooling. Since each patch has a spatial size of 4×4 and is processed over a group of four bands, both pooling operators produce 8 scalar values.

  These values are then combined through an interleaving operation, in which max and min pooled responses are alternated to form a single multivector. This interleaving preserves the correspondence between extreme responses computed at the same spatial locations and spectral group, enabling a compact representation of both spatial and spectral information within a single multivector.

  Therefore, the total number of multivectors obtained by this operation is $\frac{D^{\text{Vol}}}{4}$.

Each patch $\mathcal{P}_{ij}$ is finally represented by the concatenation of the three different sets of features:

$$\mathcal{F}_{ij} = \text{Concat}(F_{ij}^{\text{Spa}}, F_{ij}^{\text{Spe}}, F_{ij}^{\text{Vol}}) \tag{6}$$

In this way, the total number of multivectors for each patch is $D^{\text{Spe}} + D^{\text{Spa}} + \frac{D^{\text{Vol}}}{4}$.

3.2. Classification head

After creating a set of multivector features $\mathcal{F}_{ij}$ for each patch $\mathcal{P}_{ij}$, these sets are passed through GATr, where the attention mechanism operates across the patches:

$$\mathcal{M} = \text{GATr}(\mathcal{F}) \in \mathbb{R}^{C_{out} \times n \times 16} \tag{7}$$

where C_{out} denotes the number of output channels, n the number of patches, and 16 the dimension of each multivector.

With $C_{\text{out}} = 1$, the reduction is applied only along the patch dimension using an EquiLinear layer (Section 2.2), which reduces the number of multivectors from n to 1. Then, a linear layer maps the multivector components to the class logits for cross-entropy loss. The overall output of the network is:

$$y = \text{Linear}(\text{EquiLinear}(\mathcal{M})) \in \mathbb{R}^{c} \tag{8}$$

where c denotes the number of classes.

4. Experiments

4.1. Datasets

The experiments were conducted on four benchmark hyperspectral datasets designed for land cover classification: Indian Pines (IP), Salinas Scene (SA), Pavia University (PU), and Kennedy Space Center (KSC).

The **IP** and **SA** datasets were collected by the AVIRIS sensor over Indiana and Salinas Valley (US), respectively. IP has 145×145 pixels and 224 spectral bands (200 used after water absorption removal), covering mainly agricultural areas with 16 classes and 10,249 labeled pixels. SA contains 512×217 pixels and 224 bands (200 used), including crops, bare soil, and vineyards, with 54,129 labeled pixels in 16 classes.

The **PU** dataset was acquired by the ROSIS-03 sensor over the University of Pavia (IT), with 610×610 pixels, 103 spectral bands (430-860 nm), 42,776 labeled pixels, and 9 classes.

Table 1
Hyperparameter configurations for each dataset. The *Spa/Spe/Vol* column refers to the number of channels after PCA reduction for each representation, while *Hidden dim* column refers to H-GATr hidden dimension.

Dataset	Window size	Spa/Spe/Vol	Hidden dim	Heads
IP	16 × 16	16/16/64	24	8
KSC	16 × 16	16/16/64	8	8
SA	16 × 16	16/16/64	16	8
PU	16 × 16	32/32/64	24	8

Table 2
Classification result on the KSC dataset. The classes C1, …, C13, correspond respectively to classes 1, …, 13 of the KSC dataset (e.g., C1 = Scrub). The colors refer to those used in the ground truth of the KSC dataset.

Method	GACCN[18]	SSFTT[16]	SSRN[22]	H-GATr
OA (%)	97.14±1.38	99.59±0.24	99.78±0.64	**99.85**±0.14
AA (%)	96.13±1.71	99.21±0.50	98.36±1.30	**99.73**±0.24
κ (%)	96.82±1.53	99.55±0.27	99.22±0.71	**99.83**±0.15
C1	98.48±1.10	99.72±0.59	99.92±0.14	**99.98**±0.05
C2	95.68±7.50	99.04±1.88	97.79±4.82	**99.95**±0.15
C3	93.99±5.12	99.61±0.66	99.47±1.22	**100.0**±0.00
C4	86.82±6.07	**98.99**±1.61	97.97±3.18	98.42±3.06
C5	91.65±6.06	**99.38**±1.32	88.60±12.50	98.61±2.35
C6	97.33±2.26	99.95±0.15	**100.0**±0.00	99.57±0.60
C7	98.27±2.08	94.47±6.10	95.06±7.29	**100.0**±0.00
C8	96.47±7.42	99.36±1.69	99.94±0.18	**99.97**±0.08
C9	99.24±0.77	**99.98**±0.07	99.97±0.08	99.93±0.14
C10	98.01±1.37	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00
C11	97.81±1.59	99.26±1.25	**100.0**±0.00	**100.0**±0.00
C12	96.00±1.94	99.96±0.09	**100.0**±0.00	**100.0**±0.00
C13	99.93±0.10	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00

The **KSC** dataset was collected by the AVIRIS sensor over Florida (US), with 512 × 614 pixels, 176 spectral bands (18 m/pixel), and 13 classes (5,211 labeled pixels).

Each dataset includes one hyperspectral image and its corresponding ground-truth map. Since no official split is provided, 10% of labeled pixels were used for training on IP and KSC, while 100 pixels per class were selected for SA and PU.

4.2. Experiments configurations

The hyperparameters to be chosen concern both the creation of the multivectors and the GATr architecture. The final layers, on the other hand, have dimensions that depend solely on the number of classes and the number of patches.

For multivector creation, a 16 × 16 window was used for all four datasets (Table 1). The window size directly affects the number of extracted patches, over which the attention mechanism operates, and consequently influences the model's number of parameters and computational cost. A 16 × 16 window captures local spatial information while limiting the inclusion of distant pixels, providing a good trade-off by enabling effective patch-level attention without excessively increasing the model complexity.

PCA was applied to reduce the number of bands for the three representations of each window, so that the number of multivectors generated would be the same across the three different feature types. Accordingly, 16/16/64 was chosen for the Spatial (Spa), Spectral (Spe), and Volumetric (Vol) representations, respectively. Only for PU the values 32/32/64 were selected. In the PU dataset, many classes are arranged in narrow regions that extend across the image. For this reason, prediction based on similarity with neighboring pixels is less effective than in other datasets, where classes are distributed in blocks. Consequently, it was considered preferable to use a larger number of features.

Table 3
Classification result on the SA dataset. The classes C1, …, C16, correspond respectively to classes 1, …, 16 of the SA dataset (e.g., C1 = Brocoli green weeds 1). The colors refer to those used in the ground truth of the SA dataset.

Method	GACCN[18]	SSFTT[16]	SSRN[22]	H-GATr
OA (%)	99.21±0.25	99.07±0.35	96.20±0.37	**99.31**±0.12
AA (%)	99.54±0.13	99.55±0.21	98.53±0.14	**99.67**±0.08
κ (%)	99.11±0.28	98.95±0.39	95.72±0.42	**99.23**±0.13
C1	99.92±0.14	**100.0**±0.00	99.66±0.70	**100.0**±0.00
C2	99.93±0.13	99.94±0.16	**100.0**±0.00	**100.0**±0.00
C3	99.98±0.03	**100.0**±0.00	**100.0**±0.00	**100.0**±0.00
C4	98.87±0.53	99.82±0.27	**99.95**±0.08	99.76±0.23
C5	**99.25**±0.63	99.20±0.54	99.08±0.82	99.15±0.25
C6	99.95±0.06	99.98±0.05	99.97±0.05	**100.0**±0.00
C7	99.89±0.23	99.98±0.04	99.91±0.14	**99.99**±0.01
C8	**98.36**±0.56	97.08±1.38	88.14±2.94	98.19±0.23
C9	99.93±0.12	99.91±0.23	99.95±0.05	**100.0**±0.00
C10	99.20±0.92	98.69±2.44	98.73±0.97	**99.55**±0.40
C11	99.93±0.08	99.99±0.04	99.40±1.10	**100.0**±0.00
C12	99.91±0.10	**99.97**±0.06	99.80±0.46	99.95±0.10
C13	99.94±0.10	**99.96**±0.13	99.80±0.44	99.89±0.22
C14	99.76±0.35	99.79±0.36	99.86±0.19	**99.95**±0.09
C15	97.86±0.96	**98.87**±1.07	92.3±4.51	98.37±0.65
C16	99.97±0.08	99.58±0.40	**100.0**±0.00	99.98±0.03

Table 4
Classification result on the PU dataset. The classes C1, …, C9, correspond respectively to classes 1, …, 9 of the PU dataset (e.g., C1 = Asphalt). The colors refer to those used in the ground truth of the PU dataset.

Method	GACCN[18]	SSFTT[16]	SSRN[22]	H-GATr
OA (%)	98.36±0.30	98.28±0.69	99.48±0.30	**99.56**±0.16
AA (%)	97.77±0.36	98.30±0.32	**99.55**±0.13	99.53±0.13
κ (%)	97.81±0.40	97.71±0.91	99.29±0.40	**99.41**±0.21
C1	97.16±1.17	97.70±1.76	99.44±0.41	**99.55**±0.28
C2	99.49±0.28	98.60±1.26	99.47±0.62	**99.69**±0.25
C3	97.52±1.62	96.88±2.19	**99.55**±0.45	99.30±0.46
C4	96.97±1.06	97.29±1.25	**98.70**±0.60	98.10±0.96
C5	99.61±0.44	99.70±0.36	99.98±0.06	**100.0**±0.00
C6	99.21±0.71	99.33±0.52	**99.99**±0.05	99.97±0.06
C7	98.25±1.64	99.55±0.57	99.99±0.08	**99.99**±0.03
C8	95.24±0.99	96.79±2.41	**99.99**±0.03	99.30±0.37
C9	96.46±1.37	98.83±0.73	99.71±0.55	**99.87**±0.12

For the GATr hyperparameters, a different hidden dimension was chosen for the different dataset (Table 1). The high value for IP can be attributed to the complexity of the dataset, while the low value for KSC reflects the small sample size. An high hidden dimension was also chosen for the PU dataset due to its elevated number of features.

To improve performance, data augmentation techniques were applied on the fly during training. These included vertical and horizontal flips, as well as random rotations of 90°, 180°, or 270° of the window. Dropout was also applied within GATr and in the final EquiLinear layer.

4.3. Experimental results

The proposed H-GATr was evaluated against several baseline models, including the GA-CNN [18], the Spectral-Spatial Feature Tokenizer Transformer (SSFTT) [16], and the Spectral-Spatial Residual Network (SSRN) [22].

The evaluation metrics used were Overall Accuracy (OA), Average Accuracy (AA), and Cohen's Kappa coefficient (κ). To ensure that the obtained results do not solely depend on the random data split, ten training runs are performed, each time selecting different pixels to compose the training and test datasets. Accordingly, the results are presented as mean ± variance.

Table 5
Classification result on the IP dataset. The classes C1, …, C16, correspond respectively to classes 1, …, 16 of the IP dataset (e.g., C1 = Alfalfa). The colors refer to those used in the ground truth of the IP dataset.

Method	GACCN[18]	SSFTT[16]	SSRN[22]	H-GATr
OA (%)	97.53±0.41	98.21±0.39	**98.60**±0.39	97.63±0.48
AA (%)	95.90±1.18	95.60±1.59	96.64±2.01	**97.04**±0.97
κ (%)	97.18±0.31	97.96±0.45	**98.41**±0.45	97.29±0.55
C1	92.71±4.23	96.05±3.77	91.81±13.32	**97.81**±2.55
C2	96.32±1.19	98.23±0.72	**98.45**±1.48	96.05±1.14
C3	97.17±1.03	98.00±1.18	**99.18**±0.53	97.77±1.03
C4	97.57±1.83	95.08±4.57	97.85±2.15	**98.95**±1.65
C5	96.96±1.43	96.25±3.40	**97.79**±2.39	93.89±6.03
C6	97.23±0.85	**99.07**±0.86	98.61±1.36	98.84±1.22
C7	**96.75**±5.01	80.42±21.52	93.52±10.02	88.77±7.77
C8	99.72±0.42	99.53±0.72	**100.0**±0.00	**100.0**±0.00
C9	82.31±10.48	84.37±16.47	83.33±18.22	**98.75**±3.95
C10	97.09±0.80	97.30±1.48	**97.78**±0.84	94.92±1.90
C11	98.24±0.54	**99.12**±0.67	99.01±0.49	98.63±0.50
C12	95.56±1.11	96.34±2.41	**97.29**±1.36	96.10±1.82
C13	98.28±1.61	99.94±0.19	98.79±1.54	**100.0**±0.00
C14	99.26±0.46	99.53±0.41	**99.69**±0.42	99.52±0.32
C15	97.77±1.40	97.48±1.91	98.31±1.79	**98.57**±1.41
C16	91.42±4.26	92.93±4.17	**95.67**±5.73	94.08±4.61

Table 6
Number of learnable parameters (in thousands).

Dataset	GACNN [18]	SSRN [22]	SSFTT [16]	H-GATr
IP	22.5	750.0	153.2	98.9
SA	22.5	735.9	153.2	106.0
PU	16.8	651.2	153.2	18.4
KSC	132.0	397.2	153.2	50.4

All experiments were conducted on a NVIDIA T4 GPU, using a learning rate of 0.001 and a batch size of 8.

Tables 2–5 report the OAs, AAs, κ coefficients, and the classification accuracies of all classes for HSI classification, while the Figs. 4–5 show the prediction maps of all baseline models. The legends are provided in the corresponding Tables 2–5.

Overall, H-GATr achieves the best or comparable performance across all evaluated datasets. In particular, it attains the highest OA and κ on the KSC, PU, and SA datasets. H-GATr also achieves the highest AA on the KSC, PU, and IP datasets, while on the SA dataset SSRN slightly outperforms it in terms of AA by 0.02. On the IP dataset, H-GATr achieves lower OA (-0.97) and κ (-1.12) compared to SSRN. The most notable improvements of H-GATr are observed in terms of AA on the KSC (+0.52) and IP (+0.40) datasets. These gains are mainly reflected in higher classification accuracies for minority classes, namely class C7 in KSC and class C9 in IP. In addition, H-GATr exhibits lower or comparable standard deviations across most datasets and evaluation metrics, indicating stable performance across runs.

This low standard deviations observed across most datasets indicate that H-GATr captures class-specific characteristics consistently across all classes and samples.

The model is also able to generalize effectively. For instance, it achieves high accuracy on the Grapes class (C8) in the Salinas dataset, as shown in Fig. 5, despite the significantly larger number of test samples compared to training samples. It also performs well on classes with very few training samples, such as C1 (42 samples) and C9 (20 samples) in the IP dataset.

H-GATr leverages spatial information to assign neighboring samples to the same class. When neighboring classes are spectrally distinct or have a sufficient number of samples (as in the PU and SA datasets), the model accurately distinguishes between them. On the IP dataset, OA and κ are slightly lower than SSRN, which can be explained by the combination of high spectral similarity and low sample counts for several classes. In particular, the spatial proximity of similar classes can make relational aggregation more challenging, slightly reducing performance for minority or closely spaced classes. Nevertheless, H-GATr still attains higher average accuracy and relatively low variance, reflecting its ability to handle minority and challenging classes effectively (Fig. 4).

The effect of H-GATr's ability to extract multivector features without relying on additional trainable weights is evident in Table 6. Although H-GATr has slightly more parameters than GACCN, it consistently outperforms it across all metrics and datasets, showing a more efficient use of its weights. H-GATr also reduces the number of parameters by over 85% compared to SSRN, and by between 30% (on the SA dataset) and over 82% (on the PU dataset) compared to SSFTT, offering an advantage in memory-limited scenarios while maintaining high accuracy.

4.4. Ablation study

We performed an ablation study on the features extracted from the images, namely spatial (Spa), spectral (Spe), and volumetric (Vol) features (Table 7). The goal of this study was twofold: first, to determine whether any single feature type is more important than the others, and second, to assess whether combining all three features leads to actual performance improvements compared to using each feature individually or in partial combinations.

The ablation study shows that the spatial feature is the most impactful. When considered individually, it achieves higher performance than the other two features. Moreover, the combination spatial and spectral features attains results that are very close, in absolute terms, to those obtained using all three features. In particular, in terms of OA, the variation with respect to the full feature combination ranges between -0.37% and +0.07%.

The volumetric feature, on the other hand, exhibits the lowest performance when used alone. This behavior is expected, as it provides a more compact representation of the input data due to pooling, which inherently reduces the amount of available fine-grained information.

Nevertheless, the contribution of the volumetric feature when combined with the others is limited when it is paired with a single feature: it

Table 7
Ablation study evaluating the contribution of spatial (Spa), spectral (Spe), and volumetric (Vol) features on the four hyperspectral datasets.

Features			KSC			SA			PU			IP		
Spa	Spe	Vol	OA(%)	AA(%)	k(%)	OA(%)	AA(%)	k(%)	OA(%)	AA(%)	k(%)	OA(%)	AA(%)	k(%)
✓			99.43	98.82	99.36	98.86	99.49	98.73	99.43	99.32	99.24	97.35	97.04	97.02
	✓		99.44	98.80	99.37	97.85	99.16	97.60	98.59	98.44	98.12	96.52	96.04	96.03
		✓	99.04	98.32	98.93	97.19	98.20	96.87	92.91	92.55	90.58	95.19	94.77	94.52
✓	✓		99.54	98.95	99.49	98.94	99.56	98.82	99.40	99.34	99.20	**97.70**	**97.34**	**97.38**
✓		✓	99.53	99.16	99.48	98.77	99.46	98.62	99.15	99.23	98.86	97.48	96.90	97.13
	✓	✓	99.37	98.88	99.30	98.50	99.34	98.33	98.69	98.65	98.25	96.93	96.82	96.51
✓	✓	✓	**99.85**	**99.73**	**99.83**	**99.31**	**99.67**	**99.23**	**99.56**	**99.53**	**99.41**	97.63	97.04	97.29

improves the performance of the spatial feature alone only on the KSC dataset, and that of the spectral feature alone only on the PU and IP datasets. In contrast, when the volumetric feature is jointly combined with both spatial and spectral features, it consistently contributes to improved performance. Overall, the combination of all three features yields the best results in most cases.

The only dataset for which the spatial and spectral combination yields better results is IP. This behavior can be attributed to the relatively small size of the IP dataset, which contains a limited number of training samples for some classes. In this setting, the additional abstraction introduced by the volumetric representation may require a larger amount of training data to be effectively exploited and generalized by the model.

In conclusion, the combination of all three features is retained in the study, as it provides the best results on three out of four datasets, thus representing the most robust configuration overall.

5. Conclusion

In this letter, we introduced H-GATr, a GA-based framework for hyperspectral image classification that replaces convolutional preprocessing with a direct multivector embedding of spatial, spectral, and volumetric information. By leveraging the expressiveness of GA, H-GATr captures relational dependencies among patches without additional trainable filters, while maintaining state-of-the-art accuracy across four benchmarks. This highlight the potential of GATr for modeling geometric information beyond classical primitives.

A potential direction for future work is to explore the use of GATr for integrating additional data modalities beyond hyperspectral images, such as LiDAR, SAR, or multispectral images. This is possible because GATr can handle scalar inputs in parallel channels alongside multivector features, while the multispectral images and other 3D modalities could be encoded as multivectors following an approach similar to that used in this work.

CRediT authorship contribution statement

Jacopo Fabi: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Claudio Schiavella:** Writing – review & editing, Visualization, Validation, Supervision, Conceptualization; **Irene Amerini:** Writing – review & editing, Validation, Supervision, Project administration.

Data availability

The data and the code used in this research are publicly available. The datasets can be obtained from publicly accessible repositories at the following link: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes. The source code is available at the following link: https://github.com/jfabi88/HGATr

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] R. Hernández-Clemente, A. Hornero, M. Mottus, et al., Early diagnosis of vegetation health from high-resolution hyperspectral and thermal imagery: lessons learned from empirical relationships and radiative transfer modelling, Curr. For. Rep. 5 (2019) 169–183. https://doi.org/10.1007/s40725-019-00096-1

[2] S. Peyghambari, Y. Zhang, Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: an updated review, J. Appl. Remote Sens. 15 (3) (2021) 031501. https://doi.org/10.1117/1.JRS.15.031501

[3] G. Tejasree, L. Agilandeeswari, Land use/land cover (LULC) classification using deep-LSTM for hyperspectral images, Egypt. J. Remote Sens. Space Sci. 27 (1) (2024) 52–68. https://doi.org/10.1016/j.ejrs.2024.01.004

[4] V. Kumar, R.S. Singh, M. Rambabu, Y. Dua, Deep learning for hyperspectral image classification: a survey, Comput. Sci. Rev. 53 (2024) 100658. https://doi.org/10.1016/j.cosrev.2024.100658

[5] M. Ahmad, S. Distefano, A.M. Khan, M. Mazzara, C. Li, H. Li, J. Aryal, Y. Ding, G. Vivone, D. Hong, A comprehensive survey for hyperspectral image classification: the evolution from conventional to transformers and mamba models, Neurocomputing 644 (2025). https://doi.org/10.1016/j.neucom.2025.130428

[6] X. Yang, Y. Ye, X. Li, R.Y.K. Lau, X. Zhang, X. Huang, Hyperspectral image classification with deep learning models, IEEE Trans. Geosci. Remote Sens. 56 (9) (2018) 5408–5423. https://doi.org/10.1109/TGRS.2018.2815613

[7] L.D. Medus, M. Saban, J.V. Francés-Víllora, M. Bataller-Mompeán, A.R.-M. noz, Hyperspectral image classification using CNN: application to industrial food packaging, Food Contr. 125 (2021). https://doi.org/10.1016/j.foodcont.2021.107962

[8] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, G. Bai, HSI-CNN: a novel convolution neural network for hyperspectral image, in: 2018 International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, China, 2018, pp. 464–469. https://doi.org/10.1109/ICALIP.2018.8455251

[9] M. He, B. Li, H. Chen, Multi-scale 3D deep convolutional neural network for hyperspectral image classification, in: Proc. IEEE Int. Conf. Image Process, IEEE Int. Conf. Image ess, 2017, pp. 3904–3908. https://doi.org/10.1109/ICIP.2017.8297014

[10] S.K. Roy, R. Mondal, M.E. Paoletti, J.M. Haut, A. Plaza, Morphological convolutional neural networks for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14 (2021) 8689–8702. https://doi.org/10.1109/JSTARS.2021.3088228

[11] Q. Hong, X. Zhong, W. Chen, Z. Zhang, B. Li, H. Sun, T. Yang, C. Tan, SATNet: A Spatial Attention Based Network for Hyperspectral Image Classification, 14, 2022. Remote Sens, https://doi.org/10.3390/rs14225902

[12] X. Yang, W. Cao, Y. Lu, Y. Zhou, Hyperspectral image transformer classification networks, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–15. https://doi.org/10.1109/TGRS.2022.3171551

[13] S.K. Roy, A. Deria, C. Shah, J.M. Haut, Q. Du, A. Plaza, Spectral-spatial morphological attention transformer for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–15. https://doi.org/10.1109/TGRS.2023.3242346

[14] Y. Peng, J. Ren, J. Wang, M. Shi, Spectral-Swin Transformer with Spatial Feature Extraction Enhancement for Hyperspectral Image Classification, 2023. Remote Sens. 15, https://doi.org/10.3390/rs15102696

[15] S. Mei, C. Song, M. Ma, F. Xu, Hyperspectral image classification using group-Aware hierarchical transformer, IEEE Trans. Geosci. Remote Sens. 60 (2022) 5539014. https://doi.org/10.1109/TGRS.2022.3207933

[16] L. Sun, G. Zhao, Y. Zheng, Z. Wu, Spectral-spatial feature tokenization transformer for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–14. https://doi.org/10.1109/TGRS.2022.3144158

[17] D. Ruhe, J.K. Gupta, S.D. Keninck, M. Welling, J. Brandstetter, Geometric Clifford Algebra Networks, 202 of *40th Int. Conf*, PMLR, 2023. https://proceedings.mlr.press/v202/ruhe23a.html.

[18] Y. Li, Y. Wang, R. Wang, Y. Wang, K. Wang, X. Wang, W. Cao, W. Xiang, GA-CNN: convolutional neural network based on geometric algebra for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–14. https://doi.org/10.1109/TGRS.2022.3212682

[19] R. Wang, X. Ye, Y. Huang, M. Ju, W. Xiang, GASSF-Net: Geometric Algebra based Spectral-Spatial Hierarchical Fusion Network for Hyperspectral and LiDAR Image Classification, 16, 2024. Remote Sens, https://doi.org/10.3390/rs16203825

[20] J. Brehmer, P.D. Haan, S. Behrends, T.S. Cohen, Geometric algebra transformer, in: Proc. 37th Int. Conf, 37th Int. Conf, 2023, pp. 35472–35496. https://arxiv.org/abs/2305.18415.

[21] C. Doran, A. Lasenby, Geometric Algebra for Physicists, Cambridge University Press, 2003.

[22] Z. Zhong, J. Li, Z. Luo, M. Chapman, Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework, IEEE Trans. Geosci. Remote Sens. 56 (2018) 847–858. https://doi.org/10.1109/TGRS.2017.2755542