SQformer: Spectral-Query transformer for hyperspectral image arbitrary-scale super-resolution

Shuguo Jiang, Nanying Li, Meng Xu, Shuyu Zhang, Sen Jia

***Abstract*—Super-resolution is vital for the quality improvement of hyperspectral images (HSIs) under the spatial and spectral resolution trade-off. However, deep learning HSI super-resolution approaches typically adopt the "one model, one scale" scheme that is inefficient in training and storing. This is difficult in maximizing orbit equipment performance and aligning multiple spatial resolution data in remote sensing. So, this paper intends to address HSI arbitrary-scale super-resolution, enabling the scaling of HSIs to arbitrary sizes using a single model. To do this end, we treat HSI arbitrary-scale super-resolution as a retrieval problem. It conceptualizes the HSI as a dictionary of pixel-wise tokens with spatial-spectral features, position information, and scale information. Its objective is to employ a set of initialized tokens related to the high-resolution (HR) HSI as queries to retrieve matched spectral features from low-resolution (LR) one, which is so-called token-based query-to-spectrum. Since these query tokens can be constructed flexibly (e.g., through random initialization), we can generate a desired number of them to reconstruct our HR HSI, thus achieving arbitrary-scale super-resolution. This process considers not only position information but also spectral features so that it can decrease spectral distortion. With the above idea, we developed a HSI arbitrary-scale super-resolution method, dubbed as Spectral-Query transformer (SQformer). Specifically, it begins by converting the LR HSI into a dictionary of LR tokens and then constructs a desired number of HR tokens. To enable flexible token construction, we design an implicit spectral token (particularly a learnable vector) and replicate it $\alpha H \times \alpha W$ times to form the HR tokens. Next, the HR and LR tokens are passed into a transformer decoder to find the most matched spectral response for the former by soft-weighting the LR tokens. Finally, the HR tokens are spatially rearranged in order, forming a HR HSI. Extensive experiments have demonstrated its effectiveness on remote sensing data. The code will be released at: *https://github.com/ShuGuoJ/SQformer.git*.**

***Index Terms*—Hyperspectral image (HSI), arbitrary-scale super-resolution.**

I. Introduction

The work is supported by the National Natural Science Foundation of China (Grant No. 62271327), the Project of Department of Education of Guangdong Province (Grant No. 2023KCXTD029), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2022A1515011290), Shenzhen Science and Technology Program (Grant No. RCJC20221008092731042, JCYJ20220818100206015, KQTD20200909113951005), Research Team Cultivation Program of Shenzhen University (Grant No. 2023JCT002). (Corresponding author: Sen Jia)

Shuguo Jiang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China, and also with the School of Computer Science, Wuhan University (e-mail: shuguoj@foxmail.com);

Nanying Li, Meng Xu, Shuyu Zhang, and Sen Jia are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China (e-mail: linanying2021@email.szu.edu.cn; m.xu@szu.edu.cn; shuyu-zhang@szu.edu.cn; senjia@szu.edu.cn).

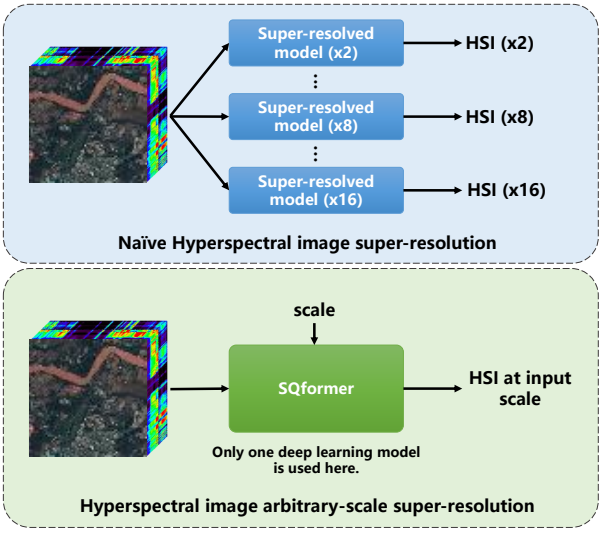

Fig. 1. Differences between naive hyperspectral image super-resolution and hyperspectral image arbitrary-scale super-resolution in deep learning. Technically, naive deep learning hyperspectral image super-resolution always regards image magnification at different scale factors as single sub-tasks and trains specific models for them. This scheme is very inefficient in training and storing. Moreover, it is unable to perform non-integer super-resolution, hard to maximize the performance of various terminals. To solve the problems, hyperspectral image arbitrary-scale super-resolution aims to treat magnification at all scales as a task and use only one model to do it.

HYPERSPECTRAL images (HSIs) [1]–[3], which contain the unique spectral curve of ground objects, have been a vital tool in earth observation [4]–[6], environmental monitoring [7], and so on. However, due to limitations imposed by the lowest signal-to-noise ratio during imaging, it already needs to compromise its spatial resolution in favor of achieving nanoscale spectral resolution. Consequent low spatial resolution usually causes mixed pixels, indistinguishable boundaries, and blurred textures, inevitably influencing related applications such as change detection [8], object recognition [9], scene interpretation [10], and classification [11], [12]. Furthermore, the low resolution (LR) of HSIs also has posed a challenge for spatially fusing them with other high-resolution (HR) modalities, such as RGB and multi-spectral images in remote sensing multi-modalities learning. Unfortunately, it is difficult to break the trade-off between spatial and spectral resolution in HSIs solely relying on hardware improvements. Therefore it is desired to enhance its spatial resolution from the aspect of algorithms in order to provide high-quality data and accurate spatial alignment for downstream tasks.

Single hyperspectral image super-resolution [13]–[15] has

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2024.3463745

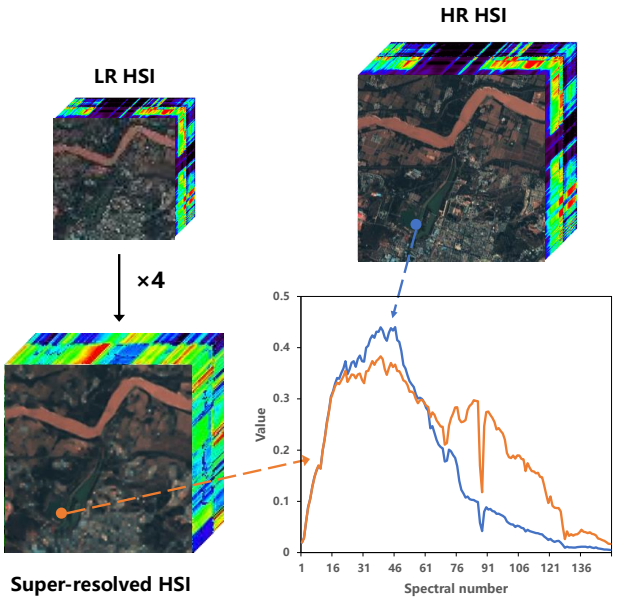

Fig. 2. Spectral distortion. Existing arbitrary-scale super-resolution methods for RGB images often encounter significant spectral distortion issues when upscaling the LR HSI by a factor of 4.

achieved great success over the last few decades, particularly with the widespread adoption and dominance of deep learning in image processing [16]–[19]. It is a feasible, efficient, and straightforward way to reconstruct HR HSIs from their LR counterparts [20]–[24]. Current methods typically adopt "one model, one scale" scheme with either a transposed convolutional layer [25] or a pixel shuffle layer [26] to learn LR-to-HR mapping for super-resolution, shown in Figure 1. These upsampling layers always require predefining an upsampling scale for parameter construction, thereby coupling spatial expansion with high-resolution feature learning for images. As a result, once models complete training, they can only upsample images to the predetermined scale factor. But, this inflexible approach is unable to freely scale up images in inference so that it is hard to meet the diverse requirements of downstream tasks. Moreover, training a single model for different scales is infeasible since there is only limited registered data and memory available to train and store models with millions of parameters in practice.

One straightforward solution to achieve HSI arbitrary-scale super-resolution is transferring RGB arbitrary-scale super-resolution methods [27]–[29] proposed recently into this field. These approaches mainly employed the implicit neural representation to learn an image function with respect to spatial coordinates. Subsequently, they utilize pixel-wise coordinates as independent variables to calculate pixel values for HR images. Owing to the continuity of spatial coordinates, it can generate arbitrary-scale images in a continuous field. However, the difference between RGB images and HSIs usually hinders their applications to achieve HSI arbitrary-scale super-resolution without spectral distortion, as shown in Figure 2. First of all, the pixel value of RGB images is three discrete channels—Red (R), Green (G), and Blue (B), while HSI's approximates a continuous spectral curve. Intuitively, regressing spectral curves by spatial coordinates is more challenging than regressing three discrete channels due to their continuity and non-linearity. Besides, even a slight change in spectral curves may lead to huge differences in the characteristics of ground objects, given their highly continuous and non-linear nature, whereas this has little influence on RGB images. So, finding an effective and efficient way to conduct arbitrary-scale super-resolution for HSIs is an urgent problem in the domain.

To address the above problems in HSI super-resolution, we attempt to treat it as a token-based query-to-spectrum process in the paper. The HSI, in the process, is conceptualized as a dictionary of pixel-wise tokens, each of which involves spatial-spectral features, position information, and scale information. To obtain a HR HSI, it would create a series of HR tokens as queries to retrieve matched spectral features from the LR dictionary. Finally, the HR tokens are rearranged spatially based on their position to compose the HR HSI. Since the HR tokens can be produced flexibly (e.g., through random initialization), we can generate a desired number of them to reconstruct our HR HSI, thus achieving arbitrary-scale super-resolution. In addition, the approach considers not only position information but also spectral features during retrieving so that it is able to increase spectral precision in the super-resolved HSI.

As a result, we designed an arbitrary-scale super-resolution method, namely SQformer, for HSIs. As shown in Figure 3, it firstly converts the LR HSI into a dictionary of LR tokens by a feature backbone, which incorporates spatial and spectral features of HSIs into tokens to ensure their visual continuity and to reduce spectral redundancy. Meanwhile, it would create a set of HR tokens as queries in the HR token construction stage. To make the token creation flexible, we design an implicit spectral token (particularly a learnable vector) and replicate it multiple times to initialize the HR tokens. In other words, each HR token is a copy of the implicit spectral token at the beginning. The implicit spectral token is learned through the super-resolution training loss, so it can find a better representation for querying. Given the time complexity of matching in a large dictionary (HSIs typically contain tens of thousands of pixels) and the local spatial relationship between HR and LR tokens, we would find a LR candidate set for each HR token in accordance with Euclidean distance ahead.

Next, the HR tokens and their candidate set are fed into a transformer decoder to search for appropriate spectral responses. The transformer decoder is composed of several cross-attention and self-attention modules, which are stacked alternatively. The cross-attention module takes as input the query token and its candidate set to soft-weight features from the latter according to their spectral and positional similarity; while the self-attention module takes as input all HR tokens from the cross-attention module to explore their non-local similarity to further register their spectral response. The non-local similarity typically exists in HSIs but has been less explored for HSI super-resolution.

To our knowledge, this is the first time to do arbitrary-scale super-resolution for HSIs with a token-based query-to-spectrum scheme and a transformer decoder. Although there is

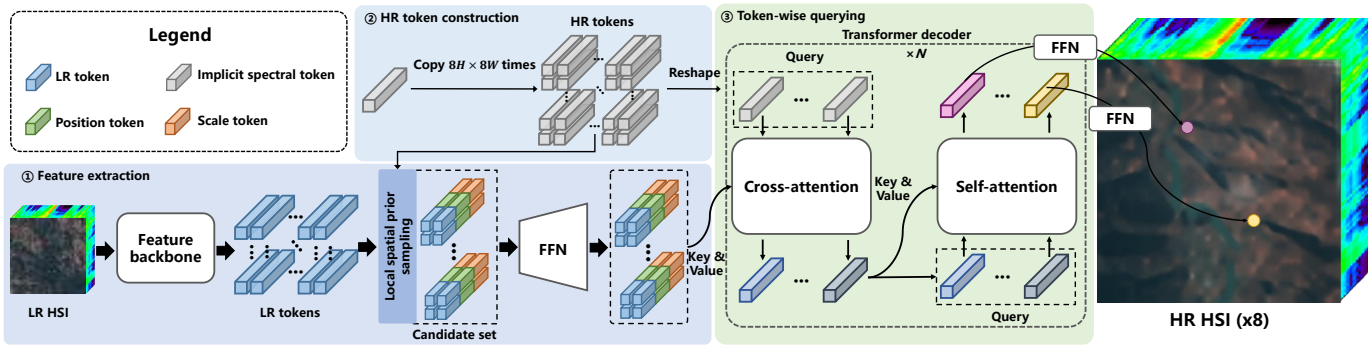

Fig. 3. The overall architecture of SQformer. It includes feature extraction, HR token construction, and token-wise querying three stages. Here takes scaling up the LR HSI by $\times 8$ as an example to demonstrate the process. The SQformer starts with converting the LR HSI into a dictionary of LR tokens through the feature backbone, embedding pixel-wise spatial and spectral features into the tokens. Concurrently, it would copy the implicit spectral token $8H \times 8W$ times to produce HR tokens. Before passed into the transformer decoder in the next stage, the HR tokens will preselect a candidate set from the LR dictionary with their local spatial prior. The selected tokens are concatenated with position and scale tokens, which is fused by a FFN to enhance its positional and scale representation. Afterward, the HR tokens and their candidate set are fed into the transformer decoder to conduct our query-to-spectrum process regarding the former as queries and the latter as keys and values. Finally, the output HR tokens are rearranged in space to form the target HR HSI.

an RGB super-resolution approach, called ITSRN [29], similar to ours, we differ in the following aspects: firstly, we use HR tokens derived from the implicit spectral tokens as queries, instead of spatial coordinates, and consider spectral features in retrieving to reduce spectral distortion; secondly, we also comprehensively consider local spatial relationships between HR and LR tokens as well as scale factors to improve model efficiency and performance; thirdly, the model incorporates non-local similarity among HR tokens to refine spectral responses during the super-resolution process for HSIs. Extensive super-resolution experiments on remote sensing datasets demonstrate that our methods can better super-resolve LR HSIs to arbitrary scales while increasing their spectral precision than other methods, as observed from both quantitative and qualitative results. An additional classification experiment on our super-resolved HSIs proves that increasing spatial resolution of HSIs can improve performance on the downstream task. The main contributions of this paper can be summarized as follows:

- This paper proposed a token-based query-to-spectrum scheme to address the challenges of HSI arbitrary-scale super-resolution. This approach converts HSI super-resolution as a spectral retrieval process from LR HSIs. The retrieval process is driven by input queries of which reconstruction is not restricted by scale factors and is theoretically infinite so that it is able to achieve arbitrary-scale magnification. Besides, it considers not only spatial distance but also spectral similarity between LR and HR tokens in matching, beneficial for spectral consistency after super-resolution. To this end, a specific model, namely SQformer, for HSI arbitrary-scale super-resolution is proposed.
- To better represent raw HSIs, an implicit spectral token is designed. Inspired by spectral similarities at the low-frequency region, we construct an implicit spectral token (essentially a learnable vector), which is used along with pixel coordinates to depict HSIs and to facilitate query construction. As the implicit spectral token learns under the supervision of training loss, it would yield a better representation for low-frequency spectral information. In addition, relative position and scale information that is crucial for super-resolution are also embedded into tokens to enhance their feature representation.
- A local spatial prior is imposed on the spectral retrieval process to accelerate its running. It is inefficient to retrieve the spectral feature for queries within LR HSIs that contain tens of thousands of pixels. Moreover, running deep learning models is computationally intensive. Considering the local spatial relationships in the mapping between HR and LR HSIs, we employ it to restrict our retrieval space, thereby reducing unnecessary computation.
- Extensive experiments on HSI datasets demonstrate our proposed method is superior to other methods. An extra classification experiment on our super-resolved HSIs proves that increasing spatial resolution of HSIs is indeed beneficial for the downstream task. To be convenient for reproduction, our code will be released at: *https://github.com/ShuGuoJ/SQformer.git*.

The remainder of the paper is organized as follows. In Section II, we will review related works regarding hyperspectral image super-resolution and arbitrary scale super-resolution. Then, a detailed elaboration of our proposed method is provided in Section III. A series of ablation and comparative experiments are conducted in Section IV. Finally, conclusions are made in Section V.

II. Related Works

A. Single Hyperspectral Image Super-resolution

Single hyperspectral image super-resolution algorithms serve as a complementary approach for hardware, assisting in enhancing the spatial resolution of HSIs. It is more efficient and straightforward than popular hyperspectral and multispectral fusion, where only LR HSIs are used. Unfortunately, the coupling of spatial and spectral characteristics in HSIs poses a challenge, how to retain spectral fidelity after spatial magnification, for HSI super-resolution. Previous research regards hyperspectral image super-resolution as a constrained

optimization problem that is solved via sparsity [20]–[22], non-local similarity [30], [31], and low rankness [32], [33]. These works are restricted by limited human knowledge, which are hard to sufficiently characterize the complex patterns in HSIs and achieve unsatisfactory HR results. Besides, they split super-resolution into several independent stages, easily incurring an under-fitting solution. End-to-end learning has been proven to be superior to multistage optimization.

In the last decade, deep learning, which is able to learn abstract representations through end-to-end training, has been introduced into this field. They all follow a standard architecture from RGB image super-resolution: feature extraction and projection from LR to HR. Their main contribution focuses on extracting features from HSIs through deep learning for reconstruction. According to the type of used networks, they can be divided into 2D CNN [34], [35], 3D CNN [36], and the mixture of 2D and 3D CNN [37], [38]. Li et al. [34] used 2D CNN combined with a spatial constraint strategy to super-resolve HSIs. The spatial constraint strategy optimizes the model by making LR HSIs generated by super-resolved HSIs close to real LR ones. Considering the 3D structure of HSIs, Mei et al. [36] replace 2D CNN with 3D CNN to simultaneously slide on the spatial and spectral dimensionality. Due to the sophisticated coupling nature of HSIs, neither of them performs well in spectral and spatial feature extraction. So, Wang et al. [37] and Li et al. [38] design hybrid networks of 2D CNN and 3D CNN for better joint feature extraction.

HSIs are typically input into models as a whole for super-resolution, resulting in intensive memory usage, which is not ideal for edge devices. Therefore, some works attempt to scale up every band individually, and Li et al. [39], as a representative of them, input each band and its two neighboring bands into models to obtain the super-resolved band one by one. To mitigate significant spectral distortion in resultant HSIs caused by separated super-resolution for each band, they also designed an enhanced back-projection method to further refine the results with spectral angle constraint. In the previous stage, acquiring HSIs in remote-sensing is hard and expensive, while a great number of RGB images is available. To alleviate the lack of HSIs, Li et al. [40] used RGB image super-resolution as an auxiliary task to pre-train deep learning models. On the other hand, Sidorov et al. [41], inspired by deep image prior [42], proposed deep hyperspectral prior to reduce dependency on a large amount of HSIs for deep learning models.

Although the quality of hyperspectral image super-resolution has been greatly improved with the powerful capability of deep learning, the earth observation and multi-modalities learning have proposed some new requirements for this domain, e.g., freely upsampling HSIs to arbitrary spatial resolution to get visual details at different levels and to be aligned with other HR data. However, current methods all fail to be competent for this demand since they need to preset a scale factor for models. Once the models finish training, they are only able to scale up HSIs with the predetermined factor. Moreover, training a separate model for each scale is infeasible and inefficient in practice. Thus, how to model degradation processes for different scales and achieve arbitrary-scale super-resolution in a single model for HSIs in remote sensing is worth considering.

B. Arbitrary-Scale Super-resolution

Until now, there are no arbitrary-scale super-resolution methods designed for HSIs, so we introduce arbitrary-scale super-resolution for RGB images as guidance here. Arbitrary-scale super-resolution in deep learning, proposed by Hu et al. [27] firstly, aims to magnify images to arbitrary scale factors only through a single deep learning model. Conventional methods, such as interpolation [43], [44], have been able to do this via a set of hand-crafted parameters. Still, they fail to achieve high-quality and photo-realistic HR images due to poor prior knowledge. Hu et al. [27] is almost the first to propose a deep learning network, MetaSR, for arbitrary-scale super-resolution. Their core idea is designing a meta learner as a parameter generator to produce a group of convolutional kernels based on pixel coordinates and scale factors. These convolutional kernels play a role in projecting pixels into any position in HR images, enabling arbitrary-scale super-resolution and outperforming conventional interpolation-based methods. This highly novel idea attracts researchers to do arbitrary-scale super-resolution with deep learning. Wang et al. [45] develop a plug-in module that includes multiple scale-aware feature adaption blocks and a scale-aware upsampling layer for asymmetric super-resolution.

Afterward, Chen et al. [28] proposed Local Implicit Image Function (LIIF) to represent images continuously, which takes pixel coordinates as inputs and outputs their corresponding RGB values. Its basic concept involves implicit neural functions, where deep learning networks are trained to learn an image function with respect to pixel coordinates, rather than conventionally learning feature extraction. As the image function implicitly encoded in neural networks is related to continuous spatial coordinates, discrete images are transformed into a continuous space. As a result, it is capable of regressing pixel values in any position of HR images. Subsequently, Yang et al. [29] developed ITSRN to super-resolve screen content images to arbitrary scales, in which pixel coordinates in LR and HR images are seen as query-key matching pairs to aggregate similar pixel values. Yet, multiple properties of pixels and non-local similarity in HR images, which can boost image quality further, are ignored in ITSRN. Currently, there are many improvements [46]–[48] based on the aforementioned works to make images freely scalable.

Their success inspires us to achieve arbitrary-scale super-resolution for HSIs to meet the increasing requirements in the HSI domain. Especially when hyperspectral imagers are becoming more lightweight, miniaturized, and cheaper, we can obtain abundant data to train deep learning models. But the gap between RGB and hyperspectral images prevents us from directly applying the above methods to the HSI domain. Owing to the coupled spatial-spectral structure, HSI super-resolution requires us to carefully maintain pixel-wise spectral consistency when performing spatial magnification. Thus, how to design an effective arbitrary-scale super-resolution architecture for HSIs is worth considering.

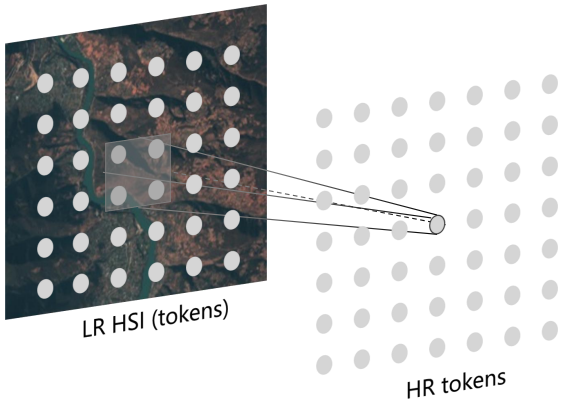

Fig. 4. Local spatial prior. After scaling down the HR tokens with $\alpha H \times \alpha W$ into the space of the LR HSI with $H \times W$, we can clearly observe that the HR token falls within the gray square region in the LR HSI and is highly correlated with the LR tokens inside.

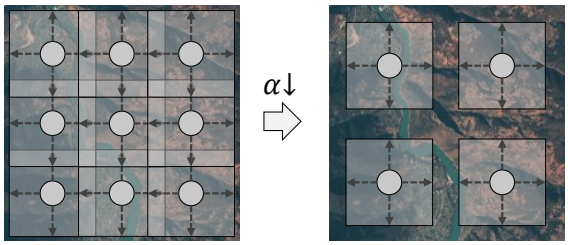

Fig. 5. Spatial unoverlap. As α decreases, the required number of tokens also declines, resulting in a sparser distribution in space, which causes spatial semantic discontinuity in HSIs.

III. Methodology

To be convenient for scaling up LR HSIs to arbitrary sizes by a single model, a token-based query-to-spectrum scheme is proposed that converts HSI super-resolution as a retrieval problem. Its core process is using a set of HR tokens as queries to retrieve matched spectral features from the LR dictionary. Benefiting from a flexible style for token construction, it can achieve HSI arbitrary-scale super-resolution by setting a desired number of HR tokens. Next, we will first describe our overall pipeline, as shown in Figure 3, and then explain our design for SQformer.

A. Overall pipeline

Figure 3 illustrates our arbitrary-scale super-resolution approach SQformer, which includes three stages: feature extraction, HR token construction, and token-wise querying. In the feature extraction stage, a LR HSI $\mathbf{I} \in \mathbb{R}^{H \times W \times B}$ is fed into a feature backbone to embed its spatial and spectral features into tokens, forming a dictionary $\mathbf{Z}$ of LR tokens in the end. Each element in the dictionary is a pixel-wise token, so $\mathbf{Z} = \{\boldsymbol{z}_{0,0}, \ldots, \boldsymbol{z}_{H,W}\}$. Here H, W, and B denote the height, width, and band number of the LR HSI. In the HR token construction stage, it would copy the implicit spectral token $\boldsymbol{p} \in \mathbb{R}^c$ $\alpha H \times \alpha W$ times to produce a series of HR tokens $\mathbf{H} = \{\boldsymbol{h}_{0,0}, \ldots, \boldsymbol{h}_{\alpha H, \alpha W}\}$ as queries for each position in the HR HSI. Note that c is channel dimension and α denotes an upsampling factor. To be convenient, we set the HR size is α times of the LR size here. As query can be constructed freely in intuition, the HR size actually can be arbitrary.

Querying in a large dictionary, where HSIs generally include tens of thousands of pixels, has high time complexity, but there exist local spatial relationships between LR and HR tokens that HR tokens are highly related to neighboring LR tokens in space (displayed in Figure 4). Consequently, we employ this local spatial prior knowledge to find a candidate set of LR tokens for each HR token to restrict its matching space. Specifically, the HR token would sample K spatially nearest LR tokens $\mathbf{z}_{i,j}$ as its candidate set $\mathbf{U}_{i,j}$, based on Euclidean distance. K is typically set as 4, so $\mathbf{U}_{i,j} = [\boldsymbol{z}_{\lfloor i/\alpha \rfloor, \lfloor j/\alpha \rfloor}, \boldsymbol{z}_{\lfloor i/\alpha \rfloor, \lceil j/\alpha \rceil}, \boldsymbol{z}_{\lceil i/\alpha \rceil, \lfloor j/\alpha \rfloor}, \boldsymbol{z}_{\lceil i/\alpha \rceil, \lceil j/\alpha \rceil}]$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote rounding real numbers up and down. In order to enhance the position and scale representation of candidate tokens, a position token and a scale token are concatenated behind them. The produced hybrid tokens are further fused by a feed-forward network (FFN) that is composed of two fully connected layers and a non-linear activation function lying in their midst.

Afterwards, the HR tokens $\boldsymbol{h}_{i,j}$ and corresponding candidate sets $\mathbf{U}_{i,j}$ then are transmitted into a transformer decoder in the token-wise querying stage. The transformer decoder stacks cross-attention and self-attention modules by turns. Its cross-attention module takes as input the HR tokens and their candidate sets to search for matched spectra, while the self-attention module takes as input all HR tokens $[\boldsymbol{h}_{0,0}, \ldots, \boldsymbol{h}_{\alpha H, \alpha W}]$ to capture their intrinsic non-local similarity. The exploration of non-local similarity here would further improve spectral consistency for super-resolved HSIs. In the end, all HR tokens are aligned with spectral dimension B by a FFN, composing the target HR HSI.

B. Feature extraction

To increase spatial coherence and decrease spectral redundancy, a feature backbone $f_\beta(\cdot)$ is used here to embed spatial-spectral features into tokens, shown as follows:

$$\mathbf{Z} = f_\beta(\mathbf{I}) \tag{1}$$

Then, the hyperspectral cube $\mathbf{I}$ is converted into a dictionary of LR tokens $\mathbf{Z}$.

We use an existing deep learning model [49]–[52] as our feature backbone since our super-resolution scheme can be plug-and-play into any model. We remove all downsampling and pooling operations from the feature backbone $f_\beta(\cdot)$ to keep the spatial size of feature maps identical to input size. Although the above operations are widely used in vision models to decrease running memory and computation, our objective here is to obtain pixel-wise tokens for LR HSIs.

C. Token design and local spatial prior

In the subsection, we will detail how to design our implicit spectral token, position token, and scale token. These tokens play crucial roles in constructing HR tokens, enhancing positional representation, and capturing scale changes. An

TABLE I
QUANTITATIVE ABLATION STUDY OF SQFORMER ON THE GF5 DATA SET. HERE, -P/-S REFERS TO REMOVING THE POSITION TOKEN AND SCALE TOKEN FROM LR TOKENS RESPECTIVELY, WHILE +L REFERS TO ADDING LAYER NORMALIZATION TO SQFORMER.

	In-distribution			Out-of-distribution		
	×2	×3	×4	×6	×8	×16
SQformer (-P)	35.06	31.95	31.00	29.47	28.63	27.02
SQformer (-S)	40.44	35.55	33.77	31.11	29.77	27.43
SQformer (+L)	40.23	35.40	33.65	31.05	29.74	27.46
SQformer	**40.54**	**35.65**	**33.86**	**31.21**	**29.86**	**27.52**

TABLE II
QUANTITATIVE COMPARISON BETWEEN SQFORMER WITH SIN-COS [53] AND OUR PROPOSED POSITION TOKEN ON THE GF5 DATA SET.

	In-distribution			Out-of-distribution		
	×2	×3	×4	×6	×8	×16
sin-cos [53]	37.06	34.01	33.31	30.87	29.20	25.12
relative pos.	**40.54**	**35.65**	**33.86**	**31.21**	**29.86**	**27.52**

TABLE III
QUANTITATIVE ABLATION STUDY ON K NEAREST NEIGHBORS.

K	In-distribution			Out-of-distribution		
	×2	×3	×4	×6	×8	×16
36	40.46	35.56	33.77	31.13	29.80	27.49
16	40.53	35.62	33.84	31.18	29.84	27.48
4	**40.54**	**35.65**	**33.86**	**31.21**	**29.86**	**27.52**

introduction to the local spatial prior employed in the approach is also provided here

Implicit spectral token. To learn a better representation, the implicit spectral token $\boldsymbol{p}$ is designed as a learnable vector that is updated with model weights by the training loss. As a result, the learned token would include the intrinsic characteristic concerning spectra. During constructing HR tokens, it would be copied $\alpha H \times \alpha W$ times to initialize them.

Local spatial prior. Figure 4 illustrates that, after scaling LR and HR tokens into the same space, the HR token falls into a local region of the LR HSI and is highly related to LR tokens in this region. Thus, we employ this local spatial prior to restrict the matching space of HR tokens. Concretely, we adopt the Euclidean distance between HR and LR tokens to find K nearest LR tokens for every HR token. The sampled LR tokens serve as its candidate set for matching.

Position token. Despite restricting the matched space, the distance of these candidate tokens to the HR one is not identical, and the closer a candidate token is to the HR one, the greater its importance. Moreover, the transformer is insensitive to their spatial relationship by nature. Thus, we have designed a position token to indicate the spatial relationship between candidate and HR tokens. The position token must be flexible and adaptive to different spatial sizes since the input and output sizes in arbitrary-scale super-resolution vary in a large range. As a result, we assign relative coordinates of the HR and candidate tokens in terms of the x-axis (height) and y-axis (width) to the position tokens. Then, it is attached to the candidate token, where $\boldsymbol{z}_{\lfloor i/\alpha \rfloor, \lfloor j/\alpha \rfloor}$ is taken as an example:

$$\widetilde{\boldsymbol{z}}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor} = [\boldsymbol{z}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor}, \Delta x, \Delta y] \tag{2}$$

Here Δx and Δy present the relative position of $\boldsymbol{z}_{\lfloor i/\alpha \rfloor, \lfloor j/\alpha \rfloor}$ with respect to $\boldsymbol{h}_{i,j}$ on two axes. And they are normalized in $[-2, 2]$ to avoid extreme values.

Owing to the relative position token, it does not need other extra operations, such as interpolation, to extend its indicated range when encountering larger spatial-size HSIs. This usually happens on absolute position encoding, where a set of position tokens is predetermined, such as the sin-cos function. Meanwhile, no trainable parameters are introduced here to increase the training load.

Scale token. The spatial arrangement of HR tokens in relation to LR tokens changes with different upsampling scales, as shown in Figure 5. With a decrease of α, the HR tokens will distribute more sparsely, causing their candidate tokens does not overlap. This would incur visual block artifacts in our super-resolved HSIs. Thus, we design a scale token $[1/\alpha]$ and also attach it behind candidate tokens in order to avoid the above issue, which is shown as follows:

$$\widehat{\boldsymbol{z}}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor} = [\widetilde{\boldsymbol{z}}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor}, 1/\alpha] \tag{3}$$

Then, the result tokens $\widehat{\boldsymbol{z}}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor}$ is transmitted into a FFN to fuse the spatial-spectral feature, relative position representation, and scale change for better representation. The operation of FFN is presented as follows:

$$\boldsymbol{u}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor} = \mathbf{W}_2(\sigma(\mathbf{W}_1 \widehat{\boldsymbol{z}}_{\lfloor \frac{i}{\alpha} \rfloor, \lfloor \frac{j}{\alpha} \rfloor})) \tag{4}$$

where $\mathbf{W}_1 \in \mathcal{R}^{d_{\boldsymbol{u}} \times (c+3)}$ and $\mathbf{W}_2 \in \mathcal{R}^{d_{\boldsymbol{u}} \times d_{\boldsymbol{u}}}$ are trainable parameters and $\sigma(\cdot)$ is the rectified linear unit (ReLU). $d_{\boldsymbol{u}}$ is channel dimension of $\boldsymbol{u}$. Afterward, the candidate token is represented as $\boldsymbol{u}_{\lfloor i/\alpha \rfloor, \lfloor j/\alpha \rfloor}$ for a clear distinction, and the set of nearest neighbors is transformed into $\widehat{\mathbf{U}}_{i,j} = [\boldsymbol{u}_{\lfloor i/\alpha \rfloor, \lfloor j/\alpha \rfloor}, \boldsymbol{u}_{\lfloor i/\alpha \rfloor, \lceil j/\alpha \rceil}, \boldsymbol{u}_{\lceil i/\alpha \rceil, \lfloor j/\alpha \rfloor}, \boldsymbol{u}_{\lceil i/\alpha \rceil, \lceil j/\alpha \rceil}]$ as well.

D. Token-wise querying

A transformer decoder consisting of N attention blocks is placed here to carry out the query-to-spectrum process. Every attention block includes a cross-attention module to match an appropriate spectrum from candidate sets for HR tokens and a self-attention module to explore non-local similarity among HR tokens. Each of them is also followed by a FFN to enhance channel features. Specifically, the cross-attention takes the HR token $\boldsymbol{h}^l_{i,j}$ as query and its LR candidate $\widehat{\mathbf{U}}_{i,j}$ as key to do match by the attention mechanism, which is shown as follows:

$$\begin{aligned} \widehat{\boldsymbol{h}}^l_{i,j} &= Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\ &= softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{\mathbf{K}}}})\mathbf{V} \end{aligned} \tag{5}$$

where $\mathbf{Q} = \boldsymbol{h}^{l-1}_{i,j}\widehat{\mathbf{W}}^l_{\mathbf{Q}}$, $\mathbf{K} = \widehat{\mathbf{U}}_{i,j}\widehat{\mathbf{W}}^l_{\mathbf{K}}$, $\mathbf{V} = \widehat{\mathbf{U}}_{i,j}\widehat{\mathbf{W}}^l_{\mathbf{V}}$. Here, l denotes the layer number and $\boldsymbol{h}^0_{i,j} = \boldsymbol{h}_{i,j}$ when l is 0. Besides, $\widehat{\mathbf{W}}^l_{\mathbf{Q}} \in \mathcal{R}^{c \times d_{\mathbf{Q}}}$, $\widehat{\mathbf{W}}^l_{\mathbf{K}} \in \mathcal{R}^{d_{\boldsymbol{u}} \times d_{\mathbf{K}}}$, $\widehat{\mathbf{W}}^l_{\mathbf{V}} \in \mathcal{R}^{d_{\boldsymbol{u}} \times d_{\mathbf{V}}}$ are three linear mappings concerning $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ respectively. The $d_{\mathbf{Q}}$, $d_{\mathbf{K}}$, $d_{\mathbf{V}}$ are the channel dimension of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ respectively. Then $\widehat{\mathbf{H}}^l = [\widehat{\boldsymbol{h}}^l_{0,0}, \ldots, \widehat{\boldsymbol{h}}^l_{\alpha H, \alpha W}]$ is got.

The $\widehat{\mathbf{H}}^l \in \mathcal{R}^{(\alpha H \times \alpha W) \times d_{\mathbf{V}}}$ is passed into the self-attention module to aggregates similar HR token features by global attention due to the non-similarity in HSIs, that is

$$\mathbf{H}^l = Attention(\widehat{\mathbf{H}}^l\mathbf{W}^l_{\mathbf{Q}}, \widehat{\mathbf{H}}^l\mathbf{W}^l_{\mathbf{K}}, \widehat{\mathbf{H}}^l\mathbf{W}^l_{\mathbf{V}}) \tag{6}$$

(a) Cosine similarity (b) Euclidean distance (c) GT

Fig. 6. Qualitative comparison between two nearest neighbor selections on the GF5 data set.

Here, $\mathbf{W}_{\mathbf{Q}}^{l} \in \mathcal{R}^{d_{\mathbf{V}} \times d_{\mathbf{Q}}}$, $\mathbf{W}_{\mathbf{K}}^{l} \in \mathcal{R}^{d_{\mathbf{V}} \times d_{\mathbf{K}}}$, and $\mathbf{W}_{\mathbf{V}}^{l} \in \mathcal{R}^{d_{\mathbf{V}} \times d_{\mathbf{V}}}$. Considering the position-invariance of patterns, e.g., objects belonging to the same class but lying in different positions should share similar features as well, positional encoding is not added here. At the end of the transformer decoder, a FFN is used to align the feature dimension of HR tokens to B so as to compose the target HR HSI $\mathbf{H}^{N} \in \mathcal{R}^{\alpha H \times \alpha W \times B}$.

IV. Experiments

A. Data sets and experimental setup

HSI Super-resolution data sets. We choose two HSI data sets, Gaofen5 (GF5) and Chikusei, as benchmarks to evaluate super-resolved results by peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and spectral angle mapping (SAM). GF5 is obtained by a visible short-wave infrared advanced hyperspectral imager (AHSI) with 30 m spatial resolution mounted on the Gaofen5 satellite. The HSIs in GF5 include 330 bands (150 visible and near-infrared bands and 180 short-wave infrared bands) ranging from 0.4 to 2.5 μm. Note that only the first 150 bands are used for super-resolution. In our experiments, 1540 images serve as the training set, and 145 images compose the testing set. Another data set, Chikusei, is collected by the Headwall Hyperspec-VNIP-C imager over agricultural and urban areas. The whole image with the size of 2517×2335 and 2.5 m spatial resolution includes 128 bands from 0.363 to 1.018 μm. In the experiments, 81 patches with the size of 256×256 are clipped from the top and left part following [55]. Among them, 64 patches are used to train, and 17 patches remain to test.

Feature backbone. As the proposed approach can be plug-and-play into any deep learning models, we conduct HSI arbitrary-scale super-resolution experiments on two CNN-based architectures, EDSR-baseline [49] and RDN [50], and two transformer-based architectures, SwinIR [51] and Restormer [52], respectively. The chosen models serve as the extractor f_{β} to encode spatial-spectral features in LR HSIs. They all have removed downsampling and pooling operations to keep the spatial size of input and output unchanged.

Parameter setting. The N in the transformer decoder is set as 3. Meanwhile, each self-attention layer consists of 96 hidden units, 4 heads, and 4 times intermediate dimensionality. Note that layer normalization is removed here since we find it is harmful to super-resolution empirically.

Optimization. For training, we follow the prior work [49] and randomly crop 48×48 patches from HSIs as inputs to networks. At the same time, scale factors for each patch are sampled from the uniform distribution $\mathcal{U}(1, 4)$, and HR counterparts are seen as the ground-truths. The model is updated by the Adam optimizer with an initial learning rate $2e{-}4$ for 1000 epochs in total, which decays by 0.5 at $[500, 700, 900, 950]$th epochs, while the configurations of MetaSR and LIIF follow [28]. As our goal is to perform arbitrary scale super-resolution for single HSIs, we evaluate the model at scale factors in training distribution $\times 1 - \times 4$ and out of the distribution: $\times 6 - \times 16$. It is worth noting that scale factors beyond the distribution are much larger than training ones.

Compared methods. We compare our methods with other arbitrary-scale super-resolution methods, such as MetaSR [27], LIIF [28], and ITSRN [29], and some single-scale super-resolution methods, such as EDSR-baseline [49], RDN [50], MCNet [56], SSPRS [54], GDD [57], EUNet [58], and ERCSR [38]. The single-scale super-resolution methods can only upsample HSIs to a fixed scale.

We set two benchmarks in the following comparative experiments. One is a comparison to arbitrary-scale super-resolution methods, while another is a comparison to single-scale super-resolution methods. The former focuses on the comprehensive performance in in-distribution $\times 2 - \times 4$ and out-of-distribution scale factors $\times 6 - \times 16$, but the latter is more concerned about the one-side performance in specific scale factors $[\times 2, \times 4]$.

B. Ablation study

Subsequently, a series of ablation experiments are conducted to evaluate the effectiveness of SQformer in terms of the position token, scale token, layer normalization, and candidate token sampling. Here, all ablation experiments employ RDN as the extractor to process HSIs and report their PSNR.

Position token. The position token with relative positional information (relative pos.) about the HR and LR tokens is attached behind candidate tokens, making the transformer decoder perceive their spatial relationship. By comparing SQformer and SQformer (-P) in Table I, we can observe that SQformer without position tokens has poor performance, demonstrating positional information is important for super-resolution. This is because HR tokens can also decide which candidate is more important by their position relationships. On the other hand, the candidate tokens and HR tokens both are reshaped as sequences before inputting into the transformer decoder, losing their 2D spatial structure. So, it is necessary to introduce additional position indications to maintain the 2D spatial structure implicitly. In Table II, we also make a comparison with sin-cos, which is widely applied in natural language processing [53] and image classification [59] to represent the absolute position of tokens. The experimental result shows SQformer with the absolute position indicator, sin-cos [53], still performs poorly, especially at $\times 16$. The absolute positional encoding needs interpolation operation to extend its encoding range when encountering larger-size HSIs, decreasing the fidelity of position description. The success of our position token validates our analysis in Section III-D that position encoding for super-resolution should be extensible since its size of inputs and outputs varies in a large range.

Scale token. To make SQformer perceive scale change as well, we append scale tokens to candidate tokens. By

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2024.3463745

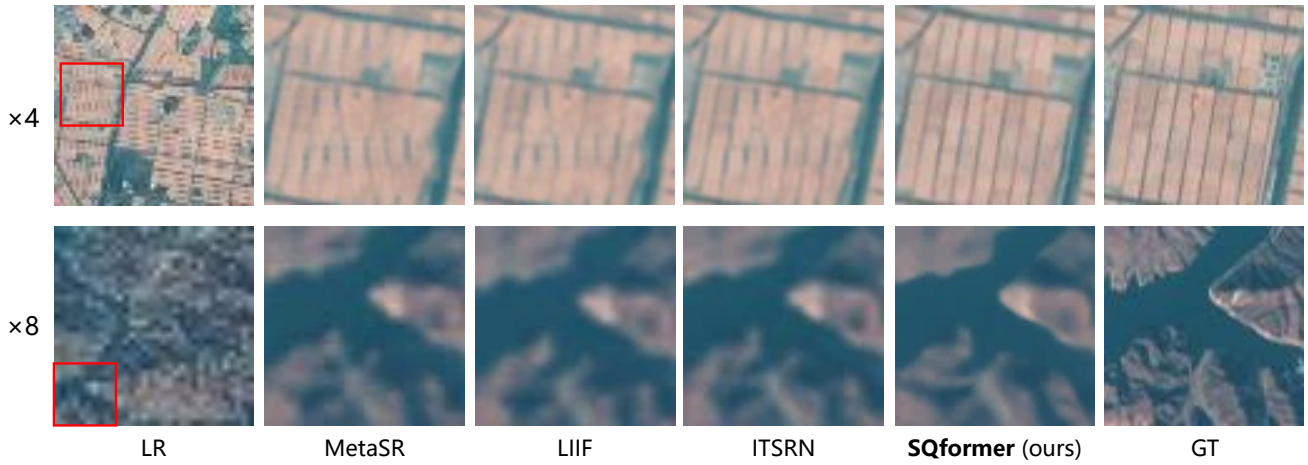

Fig. 7. Qualitative comparison of false-color images to other arbitrary scale super-resolution methods on the GF5 data set. The false-color images are composed of the 19th (blue), 29th (green), and 61th (red) bands. Here RDN is used as the feature backbone for all methods, and GT refers to HR false-color. Unlike others, SQformer can better reconstruct loss information on boundaries at the first row ($\times 4$) and has more apparent outlines at the second row ($\times 8$).

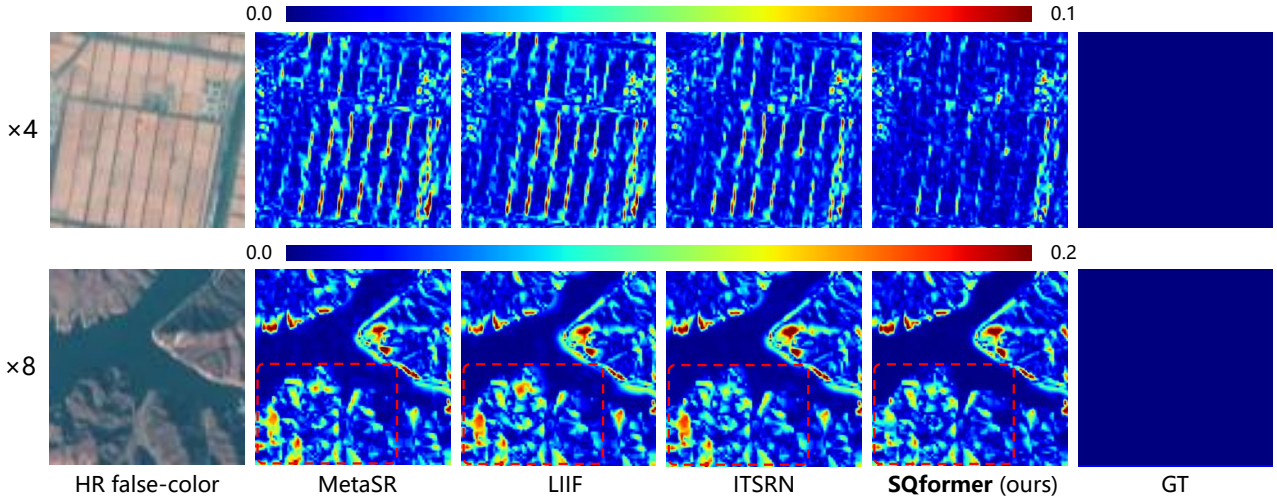

Fig. 8. Qualitative comparison of mean absolute error visualizations on the GF5 data set. Here RDN is used as the feature backbone for all methods, and GT refers to a zero matrix. The bluer the color, the closer to HR HSIs the super-resolution results.

comparing SQformer and SQformer (-S) in Table I, it is observed that adding scale tokens is beneficial to further improve super-resolution performance. On the one hand, scale tokens provide the relative ratio of LR to HR HSIs for the model. It is beneficial for alleviating image distortion when HR tokens distribute more sparsely as α decreases. On the other hand, it also avoids the sub-optimal problem, e.g., $\times 2$ images is the sub-solution of $\times 4$ ones.

Layer normalization. Although many recent models adopt layer normalization to unify the distribution of neural units, we find it may not be suitable for SQformer empirically. In Table I, it is evident that SQformer (+L) works more badly than SQformer. The reason may be LR and HR HSIs do not lie in the same distribution but layer normalization assumes they are.

Candidate sampling. We will first sample K nearest neighbors for each HR token as its candidate set and input them into the transformer decoder. The sampling is conducted based on their spatial Euclidean distance to restrict the retrieval space. In Table III, we increase K from 4 to 36 and report their super-resolution performance. It is observed that increasing K gradually has a negative impact on SQformer since more and more distant and irrelevant LR tokens are considered. As a result, $K = 4$ is optimal. In addition, we also replace Euclidean distance regarding spatial coordinates with Cosine similarity about features. Figure 6 shows SQformer adopting Cosine similarity works very poorly for super-resolution. It would sample identical LR tokens for HR ones since they all are initialized by the implicit spectral token and have the same feature at the beginning, causing SQformer collapse.

C. Comparison to arbitrary-scale super-resolution methods

This subsection reports comparisons to arbitrary-scale super-resolution methods (MetaSR [27], LIIF [28], ITSRN [29]) on 4 different feature backbones (EDSR-baseline [49], RDN [50], SwinIR [51], and Restormer [52]), respectively. The combination method is denoted as 'a-b', where 'a' represents the feature backbone name and 'b' represents the

TABLE IV
QUANTITATIVE COMPARISON TO METHODS FOR ARBITRARY SCALE SUPER-RESOLUTION ON THE GF5 DATA SET. THEY ARE EVALUATED BY PSRN (DB), SSIM, AND SAM (DEGREE°). THE BEST RESULT IS BOLDED. * REPRESENTS THAT THE MODEL IS MODIFIED WITH THE LIIF DECODER TO ACCOMMODATE ARBITRARY-SCALE SUPER-RESOLUTION.

Method	In-distribution									Out-of-distribution								
	×2			×3			×4			×6			×8			×16		
	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓
Bicubic [44]	38.27	0.975	1.81	33.60	0.935	2.99	31.92	0.909	3.61	29.82	0.870	4.54	28.70	0.850	5.12	26.70	0.824	6.27
ERCSR* [38]	37.56	0.946	1.71	33.66	0.876	2.38	32.15	0.828	2.79	30.22	0.761	3.41	29.17	0.724	3.82	27.24	0.676	4.69
SSPRS* [54]	38.65	0.955	1.35	34.25	0.887	2.05	32.58	0.840	2.50	30.48	0.770	3.17	29.37	0.732	3.60	27.34	0.678	4.52
Restormer-MetaSR [27]	38.57	0.977	1.39	34.17	0.941	2.07	32.52	0.917	2.55	30.50	0.880	3.22	29.41	0.860	3.66	**27.43**	0.829	4.61
Restormer-LIIF [28]	39.39	0.981	1.27	34.79	0.949	1.93	33.02	0.927	2.38	30.74	0.888	3.08	29.53	0.865	3.53	27.41	0.832	4.48
Restormer-ITSRN [29]	38.74	0.978	1.34	33.96	0.939	2.13	32.60	0.919	2.51	30.47	0.882	3.19	29.45	0.862	3.58	**27.43**	0.831	4.48
Restormer-SQformer (**ours**)	**39.78**	**0.982**	**1.22**	**35.05**	**0.952**	**1.87**	**33.33**	**0.932**	**2.31**	**30.88**	**0.892**	**3.03**	**29.62**	**0.868**	**3.48**	**27.43**	**0.832**	**4.45**
SwinIR-MetaSR [27]	38.72	0.978	1.36	34.22	0.941	2.05	32.55	0.918	2.54	30.45	0.879	3.37	29.25	0.855	4.06	27.07	0.815	5.62
SwinIR-LIIF [28]	40.09	0.983	1.19	35.26	0.954	1.83	33.51	0.934	2.27	30.95	0.893	3.00	29.66	0.868	3.47	27.40	0.832	4.45
SwinIR-ITSRN [29]	38.84	0.978	1.32	34.00	0.940	2.12	32.62	0.919	2.49	30.49	0.882	3.17	29.46	0.862	3.57	**27.47**	0.832	4.47
SwinIR-SQformer (**ours**)	**40.17**	**0.984**	**1.17**	**35.34**	**0.955**	**1.80**	**33.59**	**0.936**	**2.23**	**31.02**	**0.894**	**2.96**	**29.72**	**0.870**	**3.43**	**27.47**	**0.833**	**4.40**
EDSR-baseline-MetaSR [27]	38.63	0.977	1.38	34.18	0.941	2.08	32.53	0.917	2.56	30.47	0.880	3.24	29.38	0.859	3.68	27.40	0.829	4.64
EDSR-baseline-LIIF [28]	38.61	0.977	1.39	34.26	0.942	2.07	32.58	0.918	2.53	30.51	0.882	3.18	29.39	0.861	3.61	27.36	0.832	4.51
EDSR-baseline-ITSRN [29]	38.93	0.979	1.30	34.36	0.943	2.03	32.68	0.920	2.48	30.60	0.883	3.14	29.48	0.862	3.56	**27.46**	0.832	4.46
EDSR-baseline-SQformer (**ours**)	**39.81**	**0.983**	**1.20**	**35.01**	**0.952**	**1.87**	**33.23**	**0.930**	**2.32**	**30.82**	**0.890**	**3.03**	**29.59**	**0.867**	**3.49**	**27.46**	**0.833**	**4.43**
RDN-MetaSR [27]	39.86	0.983	1.24	35.00	0.952	1.92	33.27	0.931	2.38	30.79	0.889	3.15	29.57	0.864	3.64	27.43	0.829	4.69
RDN-LIIF [28]	39.78	0.982	1.23	35.03	0.952	1.91	33.22	0.930	2.35	30.81	0.889	3.06	29.59	0.866	3.51	27.44	0.833	4.45
RDN-ITSRN [29]	40.11	0.984	1.17	34.70	0.951	1.98	33.47	0.934	2.30	30.94	0.893	3.03	29.68	0.868	3.49	27.44	0.832	4.48
RDN-SQformer (**ours**)	**40.54**	**0.985**	**1.12**	**35.65**	**0.958**	**1.73**	**33.86**	**0.939**	**2.16**	**31.21**	**0.899**	**2.88**	**29.86**	**0.873**	**3.36**	**27.52**	**0.834**	**4.37**

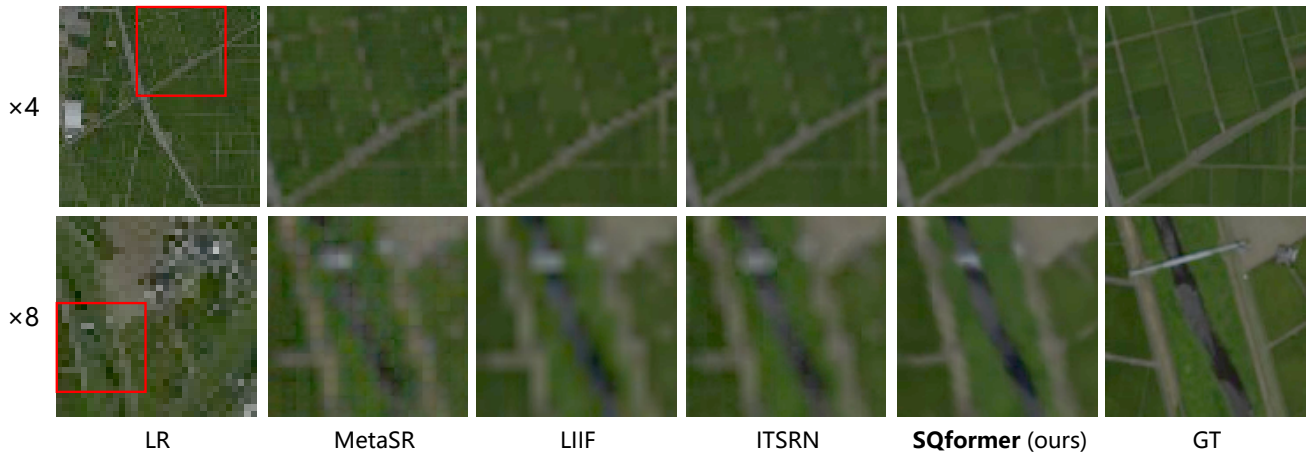

Fig. 9. Qualitative comparison of false-color images to other arbitrary scale super-resolution methods on the Chikusei data set. The false-color images are composed of the 20th (blue), 35th (green), and 45th (red) bands. Here RDN is used as the feature backbone for all methods, and GT refers to HR false-color. For SQformer, the roads across fields at the first row (×4) are clearer, and the shape of land covers at the second row (×8) is more apparent than others.

arbitrary-scale super-resolution method name. Taking "EDSR-baseline" and "MetaSR" as an example, it is dubbed as "EDSR-baseline-MetaSR". Next, we will showcase experimental results on GF5 and Chikusei data sets from the quantitative and qualitative perspectives.

Quantitative results. Table IV shows our quantitative comparisons to other methods on in-distribution ($\times 1 - \times 4$) and out-of-distributed ($\times 6 - \times 16$) scale factors. On most scale settings, SQformer outperforms other arbitrary-scale super-resolution methods in terms of visual and spectral metrics, indicating our method can better preserve spectral consistency when upsampling HSIs to arbitrary sizes. Even at an extreme scale factor of $\times 16$, it still demonstrates comparable results to other methods. Although ITSRN is similar to ours, it works poorly for HSI reconstruction owing to its ignorance of the characteristics of HSIs during super-resolution. In summary, SQformer works better than current methods at in-distribution and out-of-distribution scale factors on HSI super-resolution.

From the perspective of the architecture of feature extraction, CNN-based backbones are more suitable for SQformer, especially for RDN which contains dense connections to aggregate features from low to high levels. Contrary to ours, recently popular LIIF prefers transformer-based backbones, but it is still inferior to SQformer. The hierarchical data structure has been extensively adopted in image classification and segmentation to capture multi-scale features efficiently and effectively, while few works adopt it in super-resolution. Here, we use the newest U-shape network, Restormer [52], as the extractor to perform super-resolution and the experimental results show it is competitive with flat architectures like EDSR-baseline backbones. The outstanding performance of our method across 4 different feature backbones suggests its superior compatibility

Table V shows comparison results on Chikusei which is also extensively applied in evaluating remote sensing HSI super-resolution. The experimental results show a similar conclusion that SQformer surpasses others in most cases of in-distribution and out-of-distribution scale factors. Moreover, SQformer with CNN-based extractors can achieve better performance in remote sensing HSI arbitrary scale super-resolution.

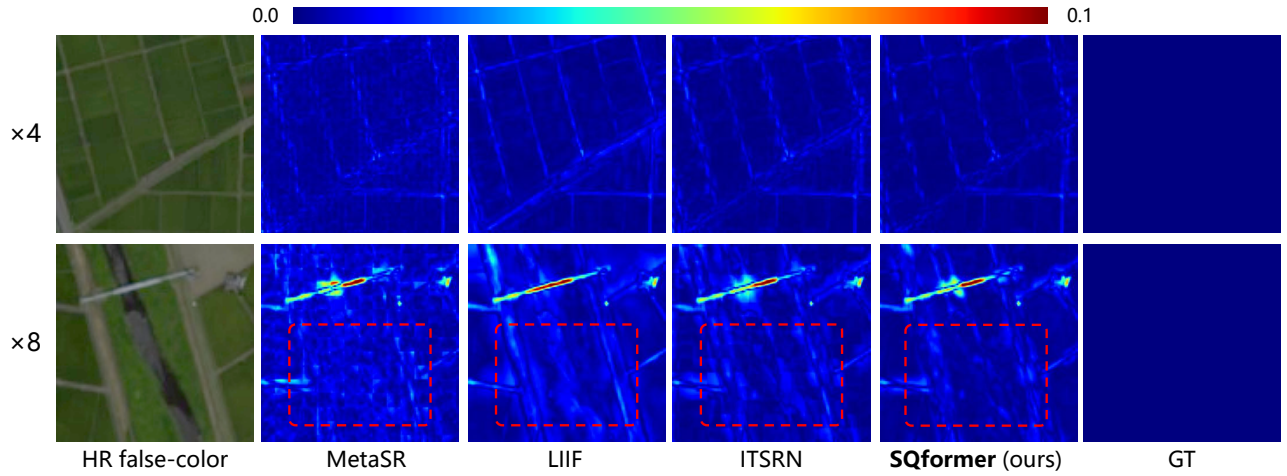

Fig. 10. Qualitative comparison of mean absolute error visualizations on the Chikusei data set. Here RDN is used as the feature backbone for all methods, and GT refers to a zero matrix. The bluer the color, the closer to HR HSIs the super-resolution results.

TABLE V
QUANTITATIVE COMPARISON TO METHODS FOR ARBITRARY SCALE SUPER-RESOLUTION ON THE CHIKUSEI DATA SET. THEY ARE EVALUATED BY PSRN (DB), SSIM, AND SAM (DEGREE°). THE BEST RESULT IS BOLDED.

Method	In-distribution									Out-of-distribution								
	×2			×3			×4			×6			×8			×16		
	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓
Restormer-MetaSR [27]	40.09	0.967	3.64	38.00	0.966	3.44	36.03	0.955	3.88	33.91	0.931	5.01	32.39	0.905	6.13	30.05	0.848	8.74
Restormer-LIIF [28]	41.61	0.987	2.86	38.08	0.973	3.49	36.12	0.958	4.01	34.01	0.936	4.87	32.55	0.918	5.56	**30.32**	0.888	7.12
Restormer-ITSRN [29]	41.24	0.985	2.96	37.14	0.965	4.36	35.68	0.953	4.35	33.75	0.931	5.25	32.40	0.914	6.02	30.08	0.880	8.08
Restormer-SQformer (ours)	**42.11**	**0.989**	**2.75**	**38.48**	**0.975**	**3.26**	**36.37**	**0.961**	**3.76**	**34.18**	**0.939**	**4.61**	**32.65**	**0.921**	**5.31**	**30.32**	**0.890**	**6.91**
SwinIR-MetaSR [27]	41.72	0.987	3.08	37.96	0.971	3.72	35.78	0.954	4.55	33.51	0.926	6.07	32.01	0.903	7.30	29.54	0.855	10.24
SwinIR-LIIF [28]	41.73	0.988	2.79	38.12	0.973	3.38	36.18	0.959	3.87	34.07	0.937	4.71	32.59	0.920	5.37	**30.37**	**0.891**	6.87
SwinIR-ITSRN [29]	41.12	0.985	2.87	36.85	0.963	4.53	35.59	0.951	4.39	33.71	0.930	5.32	32.37	0.913	6.08	30.10	0.882	7.96
SwinIR-SQformer (ours)	**42.16**	**0.989**	**2.65**	**38.44**	**0.975**	**3.18**	**36.35**	**0.961**	**3.71**	**34.16**	**0.939**	**4.55**	**32.64**	**0.921**	**5.24**	30.28	**0.891**	**6.84**
EDSR-baseline-MetaSR [27]	40.75	0.983	3.68	37.23	0.965	4.58	35.18	0.946	5.45	33.05	0.916	6.85	31.63	0.892	8.11	29.21	0.842	11.06
EDSR-baseline-LIIF [28]	39.28	0.977	3.93	36.60	0.958	4.56	35.04	0.943	5.03	33.35	0.922	5.72	32.20	0.908	6.31	30.23	0.884	7.56
EDSR-baseline-ITSRN [29]	42.18	0.989	**2.39**	37.44	0.967	3.55	36.04	0.957	3.82	33.96	0.934	4.75	32.59	0.918	5.49	**30.32**	0.887	7.25
EDSR-baseline-SQformer (ours)	**42.65**	**0.990**	2.43	**38.64**	**0.976**	**2.98**	**36.52**	**0.962**	**3.51**	**34.20**	**0.940**	**4.36**	**32.69**	**0.922**	**5.09**	30.31	**0.891**	**6.74**
RDN-MetaSR [27]	41.73	0.987	2.98	37.88	0.970	3.77	35.75	0.953	4.58	33.54	0.926	5.91	32.09	0.904	7.07	29.72	0.859	9.75
RDN-LIIF [28]	39.94	0.980	3.96	37.15	0.963	4.57	35.50	0.949	5.00	33.62	0.927	5.71	32.38	0.911	6.32	30.27	0.883	7.62
RDN-ITSRN [29]	42.12	0.988	2.41	37.43	0.967	3.81	36.01	0.956	3.89	33.92	0.933	4.84	32.56	0.917	5.59	30.27	0.886	7.44
RDN-SQformer (ours)	**42.85**	**0.990**	**2.47**	**38.91**	**0.977**	**2.99**	**36.74**	**0.964**	**3.48**	**34.38**	**0.941**	**4.32**	**32.83**	**0.924**	**5.05**	**30.44**	**0.893**	**6.70**

TABLE VI
QUANTITATIVE COMPARISON TO METHODS FOR SINGLE-SCALE SUPER-RESOLUTION ON THE GF5 DATA SET. THEY ARE EVALUATED BY PSRN (DB), SSIM, AND SAM (DEGREE°). THE BEST RESULT IS BOLDED.

	Metrics	Bicubic	EDSR-baseline	RDN	MCNet	SSPRS	GDD	EUNet	ERCSR	SQformer (ours)
×2	PSNR↑	38.27	38.93	38.90	38.55	39.87	36.61	38.91	39.05	**40.54**
	SSIM↑	0.975	0.979	0.979	0.955	0.966	0.947	0.958	0.959	**0.985**
	SAM↓	1.81	1.33	1.14	1.33	1.17	2.39	1.29	1.27	**1.12**
×4	PSNR↑	31.92	32.80	33.01	32.45	33.01	33.02	32.09	32.67	**33.86**
	SSIM↑	0.909	0.922	0.931	0.835	0.853	0.910	0.823	0.842	**0.939**
	SAM↓	3.61	2.48	2.26	2.53	2.35	3.26	2.63	2.48	**2.16**

Qualitative results. We illustrate the qualitative comparison of false-color images and mean absolute error (MAE) between SQformer and compared SOTA (ITSRN, LIIF, and MetaSR) on the GF5 and Chikusei data sets in Figure 7–10 respectively. For an in-distribution scale factor $\times 4$, our SQformer achieves a more pleasing result than other comparative methods from false-color images at the first row of Figure 7 and 9. SQformer reconstructs details around borders more accurately and has clear visual perception. Furthermore, the maps of MAE at the first row of Figure 8 and 10 also demonstrate that our result is closer to the ground truth. Even for the training scale factor $\times 4$, other methods fail to reconstruct lost spatial textures and details, especially around borders. The HSI downsampled $\times 8$ has suffered from bad spatial degradation shown at the second row of Figure 7 and 9, where a large part of spatial textures and details has got lost. Nonetheless, SQformer can still reconstruct the shapes and borders of objects better, achieving a more impressive and clear result than others. In this case, MetaSR presents discontinuity, causing a "block" effect (zooming in for clear observation), and LIIF loses high-frequency information, leading to a blurred result. Also, ITSRN has a blurred result since only spatial relationships between LR and HR HSIs are considered in reconstruction. Their MAE maps at the second row of Figure 8 and 10 also

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2024.3463745

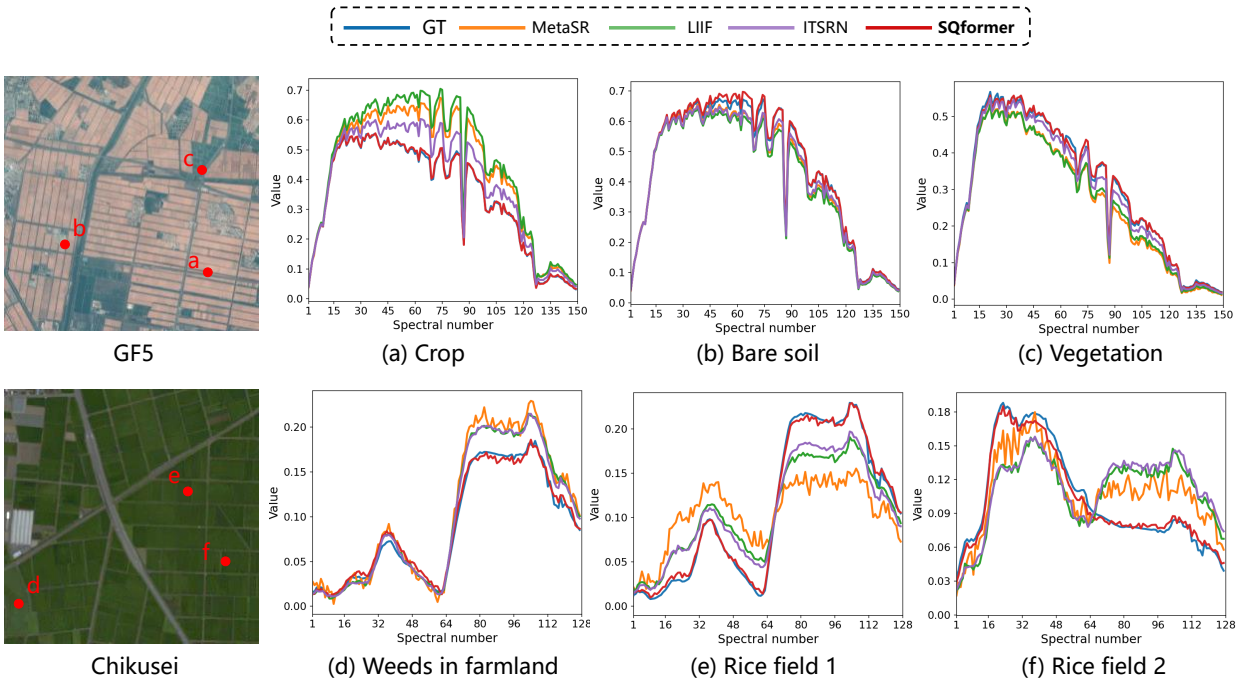

Fig. 11. Qualitative comparison of spectral curves of reconstructed HSIs on the GF5 and Chikusei data set at a, b, c, d, e, and f.

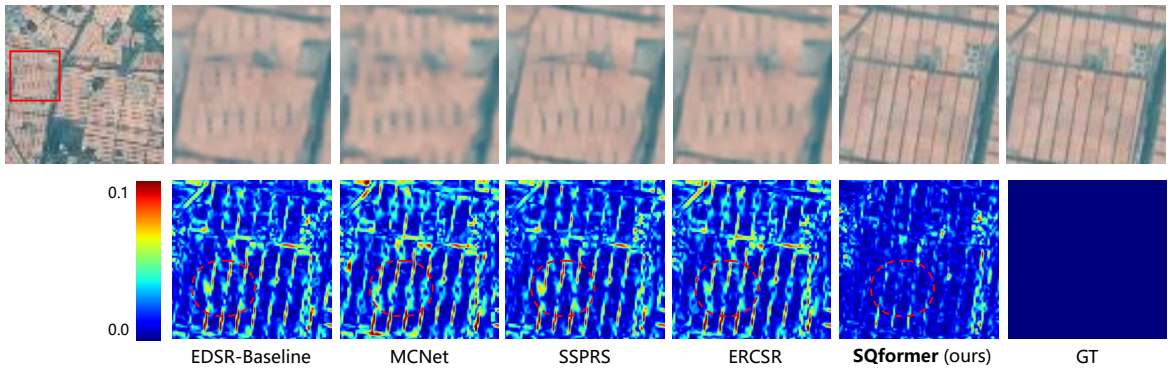

Fig. 12. Qualitative comparison to single-scale super-resolution methods at the $\times 4$ scale factor. The first row illustrates their false-color image composed by the 19th (blue), 29th (green), and 61th (red) bands, while the second is their mean absolute error image. Here, GT refers to the HR false-color image. SQformer can better reconstruct boundary information, as shown at the first row ($\times 4$), and is more approximate to the GT, shown at the second row.

show consistent results.

Besides, we also show the spectral curves of super-resolved HSIs on GF5 and Chikusei data sets in Figure 11. It reveals three common spectral distortion cases, intensity differences in (a), (b), and (c), tendency differences in (d), (e), and (f), as well as smoothness differences in (c), (e), and (f). Although comparative methods in (a), (b), and (c) are consistent with the ground truth in terms of trends, their partial spectral intensity is very different. In (d), (e), and (f), the tendency of compared methods is distinct from the ground truth. In addition to intensities and tendencies, the smoothness of reconstructed spectra is another typical visual criterion. Our reconstructed spectrum is more smooth than others in (c), (e), and (f). All in all, spectral curves generated by ours are closer to the ground truth, demonstrating the effectiveness of SQformer in reconstructing HSIs with high-dimensional spectra.

Efficiency analysis. Table VII quantitatively shows the efficiency of SQformer and other comparative methods in terms of the number of parameters (Params.), peak memory consumption (Mem.), running time, and floating point operations (FLOPs). They are calculated when given 30000 query tokens at one time, and FLOPs is got by the third-party library, THOP. Obviously, our model is much smaller than ITSRN in terms of the number of parameters but a bit bigger than LIIF. While our method may require more time for HSI reconstruction, it's important to note that the objective

TABLE VII
EFFICIENCY IN TERMS OF MODEL SIZE, PEAK MEMORY, RUNNING TIME, AND FLOPS.

Queries	Method	Params.	Mem.(GB)	Time (ms)	FLOPs
30000	MetaSR	23.51M	11435	17.01	684.92G
	LIIF	1.69M	2169	9.95	67.37G
	ITSRN	23.71M	22229	41.59	2705.12G
	SQformer (ours)	1.90M	28975	35.21	45.71G

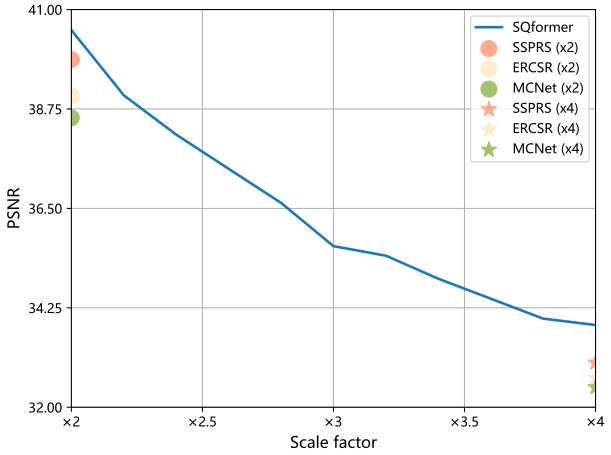

Fig. 13. Ability to integer and non-integer HSI super-resolution. SQformer sampled $[2, 3, 4]$ as integer scale factors and $[2.2, 2.4, 2.8, 3.2, 3.4, 3.8]$ as non-integer scale factors for HSI super-resolution, while comparative single-scale methods magnified HSIs at $[2, 4]$. Ideally, once more scale factors are sampled, our PSNR metrics within the range from $\times 2$ to $\times 4$ is nearly a curve.

TABLE VIII
LAND COVER CLASSES WITH NUMBER OF SAMPLES PER CLASS FOR THE YRE DATA SET.

No.	Class	Amount
C1	Building	533
C2	River	5,376
C3	Salt marsh	4,995
C4	Shallow sea	17,550
C5	Deep sea	18,677
C6	Intertidal saltwater marsh	2,343
C7	Tidal flat	1,792
C8	Pond	1,787
C9	Sorghum	646
C10	Corn	1,509
C11	Lotus root	2,719
C12	Aquaculture	8,019
C13	Rice	5,508
C14	Tamarix chinensis	1,220
C15	Freshwater herbaceous marsh	1,417
C16	Suaeda salsa	874
C17	Spartina alterniflora	580
C18	Reed	1,970
C19	Floodplain	347
C20	Locust	75
	Total	77,937

of HSI super-resolution is to enhance its quality to improve downstream tasks, rather than for display on terminals. So, we can tolerate its high latency.

D. Comparison to single-scale super-resolution methods

Apart from comparing with arbitrary-scale super-resolution methods, we also make a comparison to some single-scale super-resolution methods, such as EDSR-baseline [49], RDN [50], MCNet [56], SSPRS [54], GDD [57], EUNet [58], and ERCSR [38]. They experiment on the GF5 data set at scales ($\times 2$ and $\times 4$). These single-scale super-resolution methods are only able to upsample HSIs to a predetermined scale factor and are trained on the corresponding scale data set.

Quantitative results. Table VI reports our comparison to single-scale methods, where SQformer has performed best at $[\times 2, \times 4]$. Benefiting from the arbitrary-scale nature, SQformer should have learned to incorporate cross-scale features. These features may contribute to improving the quality of super-resolution. The lowest SAM of SQformer suggests it excels in providing accuracy spectra for super-resolved images.

Qualitative results. The super-resolution results and mean absolute error images are displayed in Figure 12. We can find that the false-color image produced by SQformer shows clearer boundaries and outlines. Its mean absolute error image is also more approximate to the ground-truth, particularly in the red circle region. Both quantitative and qualitative comparisons above indicate our excellent ability in HSI super-resolution.

Ability to non-integer HSI super-resolution. The advantage of our approach is non-integer super-resolution in comparison to single-scale HSI super-resolution methods. In order to demonstrate our non-integer super-resolution, we tested the proposed method at nine sampled scale factors $[2, 2.2, 2.4, 2.8, 3, 3.2, 3.4, 3.8, 4]$ and plotted its PSNR curve, shown in Figure 13. Single-scale super-resolution methods can only perform integer super-resolution and must train a specific model for each scale factor, which would waste a lot of training resources and storage resources. However, our model is one-shot training and is able to conduct HSI super-resolution at any scale, including integer and non-integer. Consequently, SQformer is more efficient in training and storing than single-scale super-resolution methods to some extent.

E. Improvement to HSI classification task

The HSI classification task is chosen as a baseline to evaluate the quality of super-resolved HSIs produced by our method. The Yellow River Estuary (YRE), which is imaged by the Gaofen 5 satellite over the Yellow River Estuary field, is used as a benchmark data set. It includes a HSI with 1400×1400 size and has labeled 20 ground-object categories. The detailed class information is displayed in Table VIII. We evaluate classification performance with overall accuracy (OA), where the higher OA, the better the performance.

Figure 14 shows classification results made by a logistic regression classifier on the super-resolved HSI at four scale factors ($\times 1$, $\times 2$, $\times 3$, $\times 4$). Here, $\times 1$ refers to the original HSI. It is clear that classification accuracy has been improved progressively as spatial resolution increases. This is because high spatial resolution HSIs with clearer shapes and boundaries are beneficial for extracting ground-object features. The experimental result here proves that super-resolving LR HSI to get detailed spatial information can improve downstream tasks. All in all, utilizing fine spatial and spectral information in high spatial resolution HSIs enables a more precise interpretation of the Earth's surface.

V. CONCLUSION

This paper proposed a new method SQformer by converting HSI super-resolution as a token-based query-to-spectrum process to achieve HSI arbitrary-scale super-resolution. Extensive experiments have been carried out on the GF5 and Chikusei HSI datasets, where SQformer shows better performance than either arbitrary-scale or single-scale super-resolution methods.

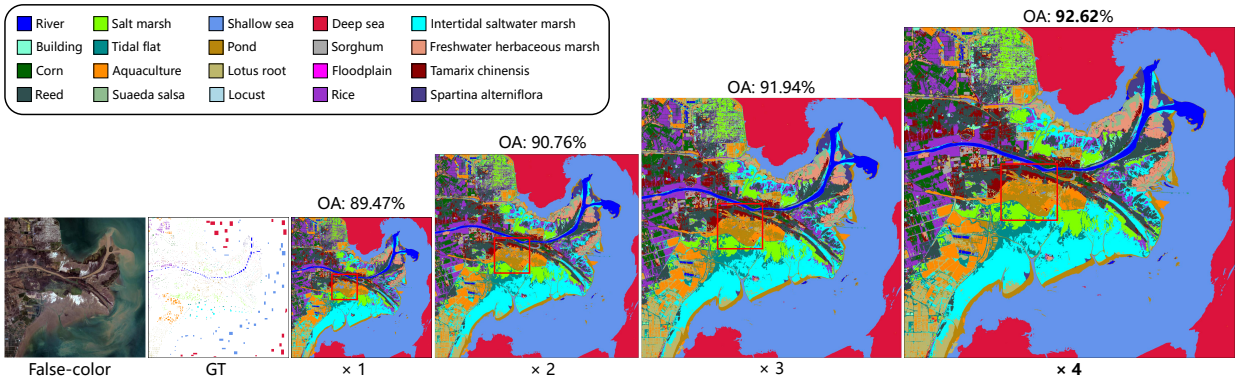

Fig. 14. Qualitative and quantitative classification results of the HSI captured over the Yellow River Estuary by the Gaofen5 satellite at $\times 1$, $\times 2$, $\times 3$, and $\times 4$. The top-left part is the legend of visualization results.

The quantitative and qualitative results, in terms of spectrum, prove its ability to precisely reconstruct pixels' spectra during spatial super-resolution. In addition to comparative experiments, a series of ablation studies have effectively demonstrated our module design as well. We have employed our super-resolved HSIs at different scales for classification, in which the higher spatial resolution of HSI results in increased classification accuracy for detailed spatial information.

The new design here for HSI arbitrary-scale super-resolution also comes with high latency. However, it is noted that HSI super-resolution intends to provide high-quality HSI for downstream tasks, instead of displaying on the terminal. As a result, the method's high latency is acceptable.

REFERENCES

[1] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, 2021.

[2] S. Li, W. Song, L. Fang, Y. Chen, and J. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, 2019.

[3] N. Audebert, B. L. Saux, and S. Lefèvre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, 2019.

[4] A. Bannari, A. Pacheco, K. Staenz, H. McNairn, and K. Omari, "Estimating and mapping crop residues cover on agricultural lands using hyperspectral and IKONOS data," *Remote Sens. Environ.*, vol. 104, no. 4, pp. 447–459, 2006.

[5] M. Teke, H. Deveci, O. Haliloğlu, S. Gürbüz, and U. Sakarya, "A short survey of hyperspectral remote sensing applications in agriculture," in *IEEE RAST.*, 2013, pp. 171–176.

[6] S. Sudharsan, R. Hemalatha, and S. Radha, "A survey on hyperspectral imaging for mineral exploration using machine learning algorithms," in *IEEE WiSPNET.*, 2019, pp. 206–212.

[7] C. Jänicke, A. Okujeni, S. Cooper, M. Clark, P. Hostert, and S. van der Linden, "Brightness gradient-corrected hyperspectral image mosaics for fractional vegetation cover mapping in northern California," *Remote Sens. Lett.*, vol. 11, no. 1, pp. 1–10, 2020.

[8] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, 2019.

[9] A. Mohammadzadeh, A. Tavakoli, and M. V. Zoej, "Road extraction based on fuzzy logic and mathematical morphology from pan-sharpened ikonos images," *Photogramm. Rec.*, vol. 21, no. 113, pp. 44–60, 2006.

[10] F. Laporterie-Déjean, H. de Boissezon, G. Flouzat, and M. Lefèvre-Fonollosa, "Thematic and statistical evaluations of five panchromatic/multispectral fusion methods on simulated pleiades-hr images," *Inform. Fusion*, vol. 6, no. 3, pp. 193–212, 2005.

[11] S. Jia, S. Jiang, Z. Lin, M. Xu, W. Sun, Q. Huang, J. Zhu, and X. Jia, "A semisupervised siamese network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.

[12] S. Jia, S. Jiang, S. Zhang, M. Xu, and X. Jia, "Graph-in-graph convolutional network for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2022.

[13] L. Gao, J. Li, K. Zheng, and X. Jia, "Enhanced autoencoders with attention-embedded degradation learning for unsupervised hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–17, 2023.

[14] D. Hong, J. Yao, C. Li, D. Meng, N. Yokoya, and J. Chanussot, "Decoupled-and-coupled networks: Self-supervised hyperspectral image super-resolution with subpixel fusion," *IEEE Trans. Geosci. Remote Sens.*, 2023.

[15] J. Li, K. Zheng, L. Gao, L. Ni, M. Huang, and J. Chanussot, "Model-informed multi-stage unsupervised network for hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, 2024.

[16] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution by coupled spectral unmixing," in *Int. Conf. Comput. Vis.*, 2015, pp. 3586–3594.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[19] G. Huang, Z. Liu, L. V. D. Maaten, and K. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.

[20] N. Akhtar, F. Shafait, and A. Mian, "Bayesian sparse representation for hyperspectral image super resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3631–3640.

[21] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3034–3047, 2019.

[22] X. Han, B. Shi, and Y. Zheng, "Self-similarity constrained sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5625–5637, 2018.

[23] H. Wang, C. Wang, and Y. Yuan, "Neighbor spectra maintenance and context affinity enhancement for single hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, 2024.

[24] J. Li, K. Zheng, Z. Li, L. Gao, and X. Jia, "X-shaped interactive autoencoders with cross-modality mutual learning for unsupervised hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, 2023.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[26] W. Shi, J. Caballero, F. Huszár, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1874–1883.

[27] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-SR: A magnification-arbitrary network for super-resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1575–1584.

[28] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 8628–8638.

[29] J. Yang, S. Shen, H. Yue, and K. Li, "Implicit transformer network for screen content image continuous super-resolution," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 13 304–13 315, 2021.

[30] R. Dian, L. Fang, and S. Li, "Hyperspectral image super-resolution via non-local sparse tensor factorization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5344–5353.

[31] R. Dian, S. Li, and L. Fang, "Learning a low tensor-train rank representation for hyperspectral image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2672–2683, 2019.

[32] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5135–5146, 2019.

[33] M. Vicinanza, R. Restaino, G. Vivone, M. D. Mura, and J. Chanussot, "A pansharpening method based on the sparse representation of injected details," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 180–184, 2014.

[34] Y. Li, W. Xie, and H. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," *Pattern Recognition*, vol. 63, pp. 371–383, 2017.

[35] J. Hu, X. Jia, Y. Li, G. He, and M. Zhao, "Hyperspectral image super-resolution via intrafusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7459–7471, 2020.

[36] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, "Hyperspectral image spatial super-resolution via 3d full convolutional neural network," *Remote Sens.*, vol. 9, no. 11, p. 1139, 2017.

[37] Q. Wang, Q. Li, and X. Li, "Hyperspectral image superresolution using spectrum and feature context," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11 276–11 285, 2020.

[38] Q. Li, Q. Wang, and X. Li, "Exploring the relationship between 2d/3d convolution for hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8693–8703, 2021.

[39] Q. Li, Y. Yuan, X. Jia, and Q. Wang, "Dual-stage approach toward hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 31, pp. 7252–7263, 2022.

[40] K. Li, D. Dai, and L. V. Gool, "Hyperspectral image super-resolution with RGB image super-resolution as an auxiliary task," in *WACV*, 2022, pp. 3193–3202.

[41] O. Sidorov and J. H. Yngve, "Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution," in *Int. Conf. Comput. Vis. Workshops*, 2019, pp. 3844–3851.

[42] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 9446–9454.

[43] P. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, no. 2, pp. 201–204, 1981.

[44] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, 1981.

[45] L. Wang, Y. Wang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning a single network for scale-arbitrary super-resolution," in *Int. Conf. Comput. Vis.*, 2021, pp. 4801–4810.

[46] J. Lee and K. Jin, "Local texture estimator for implicit representation function," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 1929–1938.

[47] S. Yang, M. Ding, Y. Wu, Z. Li, and J. Zhang, "Implicit neural representation for cooperative low-light image enhancement," in *Int. Conf. Comput. Vis.*, 2023, pp. 12 918–12 927.

[48] C. Vasconcelos, C. Oztireli, M. Matthews, M. Hashemi, K. Swersky, and A. Tagliasacchi, "Cuf: Continuous upsampling filters," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 9999–10 008.

[49] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2017, pp. 136–144.

[50] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2472–2481.

[51] J. Liang, J. Cao, G. Sun, K. Zhang, L. V. Gool, and R. Timofte, "SwinIR: Image restoration using swin transformer," in *Int. Conf. Comput. Vis.*, 2021, pp. 1833–1844.

[52] S. Zamir, A. Arora, S. Khan, M. Hayat, F. Khan, and M. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5728–5739.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inform. Process. Syst.*, vol. 30, pp. 1–15, 2017.

[54] J. Jiang, H. Sun, X. Liu, and J. Ma, "Learning spatial-spectral prior for super-resolution of hyperspectral imagery," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1082–1096, 2020.

[55] Y. Zheng, J. Li, Y. Li, J. Guo, X. Wu, and J. Chanussot, "Hyperspectral pansharpening using deep prior and dual attention residual network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8059–8076, 2020.

[56] Q. Li, Q. Wang, and X. Li, "Mixed 2D/3D convolutional network for hyperspectral image super-resolution," *Remote Sens.*, vol. 12, no. 10, p. 1660, 2020.

[57] T. Uezato, D. Hong, N. Yokoya, and W. He, "Guided deep decoder: Unsupervised image pair fusion," in *Eur. Conf. Comput. Vis.*, 2020, pp. 87–102.

[58] D. Liu, J. Li, Q. Yuan, L. Zheng, J. He, S. Zhao, and Y. Xiao, "An efficient unfolding network with disentangled spatial-spectral representation for hyperspectral image super-resolution," *Inform. Fusion*, vol. 94, pp. 92–111, 2023.

[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. Learn. Represent.*, 2020, pp. 1–21.

Shuguo Jiang received the B.E. degree from Xiamen University of Technology, Xiamen, China, in 2020 and the M.E. degree from Shenzhen University, Shenzhen, China, in 2023. He is currently pursuing his Ph.D. degree with the School of Computer Science, Wuhan University, Wuhan, China.

His research interests include remote sensing and hyperspectral image processing.

Nanying Li received the B.E. degrees in automation and M.E. degrees in information and communication engineering from the Hunan Institute of Science and Technology, Yueyang, China, in 2017 and 2021, respectively. She is currently pursuing the Ph.D. degree in computer science and technology with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Her research interests include hyperspectral image classification, anomaly detection, and image segmentation.

Meng Xu (Member, IEEE) received the B.S. and M.E. degrees in electrical engineering from the Ocean University of China, Qingdao, China, in 2011 and 2013, respectively, and the Ph.D. degree from the University of New South Wales, Canberra, ACT, Australia, in 2017. She is currently an Associate Research Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Her research interests include cloud removal and remote sensing image processing.

Shuyu Zhang received the B.E. and Ph.D. degrees from the College of Earth Sciences, Zhejiang University, Hangzhou, China, in 2015 and 2020, respectively. She is a Post-Doctoral Researcher with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Her research interests include hyperspectral image classification and deep learning.

Sen Jia (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively. Since 2008, he has been with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, where he is currently a Full Professor.

His research interests include hyperspectral image processing, signal and image processing, and machine learning.