HI-MAFE: Hyperspectral Image Multi-Agent Deep Reinforcement Learning Feature Extraction

Jin Sun, Renjie Ji, Xue Wang, Kun Tan, *Senior Member, IEEE*, and Yong Mei

Abstract—Hyperspectral image (HSI) feature extraction plays a crucial role in reducing the redundancy and correlation among spectral bands while preserving the essential information. Knowledge-driven feature extraction methods, such as spectral indices (SIs), leverage the interaction mechanisms between electromagnetic waves and materials to enhance the characteristic attributes of ground objects through band operations. These methods offer key advantages, including strong physical interpretability, simple construction, and robust scene reusability. However, most of the existing SIs still rely on expert knowledge tailored to specific scenarios, leading to inherent limitations, such as subjectivity, high time consumption, and implementation complexity. In this article, to address these challenges, we propose a HSI multi-agent deep reinforcement learning (MADRL) feature extraction (HI-MAFE) algorithm, aiming to alleviate the burden of manual SIs design by human experts. HI-MAFE employs a heuristic "generation-selection" strategy to simulate the decision-making process of domain experts, with specifically designed deep reinforcement learning (DRL) models for both the generation and selection steps. To accelerate exploration in a high-dimensional action space, the model incorporates a MADRL framework. The experimental results demonstrate the effectiveness and superiority of the proposed algorithm for HSI classification. The proposed HI-MAFE framework leverages DRL to autonomously generate meaningful environmental interpretation from spectral data, thereby reducing the reliance on manually designed SIs. This research can inspire future work in SIs construction and complement the limitations of data-driven approaches.

Index Terms—Automatic feature engineering, hyperspectral remote sensing, multi-agent deep reinforcement learning (MADRL), spectral indices (SIs).

I. INTRODUCTION

SINCE the 20th century, hyperspectral remote sensing technology has emerged as a transformative advancement in remote sensing science, distinguished by its ability to capture hundreds of continuous spectral bands for each image pixel [1], [2]. This capability delivers highly detailed spectral information, enabling the widespread adoption of the technology in various domains such as mineral identification, precision agriculture, military reconnaissance, and soil property estimation [3], [4], [5], [6]. Despite its strengths, hyperspectral remote sensing faces challenges due to the high dimensionality of the data. The redundancy and correlation among spectral bands can trigger the "Hughes" phenomenon [7]. To address this issue, dimensionality reduction processing is essential. By reducing the feature space while preserving the critical information, dimensionality reduction techniques can ensure the effective analysis of hyperspectral imagery [8].

Dimensionality reduction in hyperspectral remote sensing imagery is typically achieved through two primary approaches: feature selection and feature extraction [9]. Feature selection involves identifying and retaining the most relevant spectral bands while eliminating those deemed less informative [10], [11], [12], [13], [14], [15]. Although these methods successfully preserve the physical properties of the original bands, they often fail to account for the complex spectral interactions, leading to potential information loss in feature representation.

In contrast, feature extraction employs mathematical transformations to project high-dimensional data into a lower-dimensional space, generating a new set of features. Unlike feature selection, which eliminates specific bands, the goal of feature extraction is to summarize the information while suppressing the less relevant information. Traditional feature extraction techniques, such as principal component analysis (PCAs) [16], linear discriminant analysis (LDAs) [17], and tensor robust PCAs (TRPCAs) [18] are widely employed for this purpose. However, a key limitation of these methods is their inability to retain the physical meaning of the original features, potentially disrupting their inherent structure during the transformation process.

In recent years, with the exponential increases in computational power, artificial intelligence (AI) has emerged as one of the most important and disruptive technologies. Deep learning leverages multilayer nonlinear transformation networks to autonomously extract higher-order abstract features, offering an innovative solution for hyperspectral feature extraction [19], [20], [21]. Despite the proliferation of task-specific intelligent

interpretation algorithms that have made progress in particular application scenarios, deep feature extraction still faces several challenges, as shown in Fig. 1(a).

1) *Lack of Interpretability [22]:* the existing deep learning models often function as "closed boxes," making it difficult to link the extracted high-level abstract features to the underlying physical mechanisms.
2) *Limited Scene Reusability [23]:* deep learning models exhibit weak transferability across spatial domains, land-cover types, temporal phases, spatial resolutions, and data modalities. As a result, different tasks often require the development of scene-specific interpretation models.
3) *Rely on Complex Architectures:* feature extraction typically relies on complex deep learning architectures, designing, optimizing, and fine-tuning these intricate architectures often requires extensive experimentation, increasing the development time and effort.

Compared to the aforementioned feature extraction methods, spectral indices (SIs), which are also known as knowledge-based feature extraction, emphasize the attributes of objects through operations such as band ratios, based on the interaction mechanisms between electromagnetic waves and materials. Currently, hundreds of divers SIs have been proposed for various application fields, including agricultural management [24], fire detection [25], urban planning [26], and ecological environment assessment [27], with this number continually increasing. To address the growing diversity and demand for standardized SIs, the "awesome SIs" (ASI) project provides a machine-readable, extensible catalog linked to a Python library, which enables efficient and consistent application of SIs in Earth system research [28]. Among these SIs, the most renowned and widely used spectral index is the normalized difference vegetation index (NDVI) [29], which is typically employed to assess the greenness of vegetation both spatially and temporally. Other commonly used SIs include the leaf area index (LAI) [30] and the remote sensing ecological index (RSEI) [31]. SIs offer several advantages, as shown in Fig. 1(b).

1) *Strong Physical Interpretability [8]:* the design of SIs is typically based on the physical relationships between the spectral absorption of surface features and biochemical parameters.
2) *Robust Scene Reusability [32]:* SIs rely only on operations involving specific spectral bands, without the need for complex model training or large amounts of labeled data, which enhance their applicability across different scenarios.
3) *Simplicity In Construction [32]:* SIs are characterized by their simplicity in construction, typically involving straightforward arithmetic operations between reflectance values of a few specific spectral bands. This inherent simplicity makes them easy to understand, calculate, and implement across various platforms, making SIs valuable for rapid applications. At present, the complementarity and the necessity for synergistic application between knowledge and data in remote sensing image interpretation have become increasingly evident.

Yan et al. [33] developed a self-supervised learning method that incorporates remote sensing indices for unsupervised change detection, which significantly enhanced detection accuracy. Tian et al. [34] proposed a knowledge-guided deep learning framework that integrates knowledge and neural networks from both feature-level and network-level perspectives, resulting in substantial improvement in wheat yield estimation accuracy.

Although SIs possess the aforementioned advantages, most of the existing SIs still rely on expert judgment tailored to specific scenarios, rendering them highly subjective, time-consuming, and challenging to implement, etc. In recent years, automated feature extraction (AutoFE) methods, which can automatically generate effective features without human intervention, have been widely adopted in the processing of tabular data (also known as structured data). AutoFE formalizes feature construction as the application of transformations to the original features, with the objective of enhancing the performance of the predictive model by extracting new, informative features from the base data. Numerous automated feature generation methods have been proposed for feature engineering in tabular datasets. For example, Chen et al. [35] introduced a neural architecture for automated feature engineering, termed neural feature search (NFS). In NFS, multiple recurrent neural networks (RNNs) serve as controllers to generate transformation rules for the corresponding original features. Li et al. [36] presented feature set data-driven search (FETCH), which incorporates deep reinforcement learning (DRL) into AutoFE by updating the policy function using reinforcement learning techniques. However, FETCH employs a single-agent DRL framework, wherein one agent is responsible for determining the transformation rules for all the features. More recently, Zhang et al. [37] introduced an "expansion–reduction" framework for feature generation. In the expansion phase, various transformations are applied to the raw features to generate a larger set of candidate features; in the reduction phase, features deemed irrelevant or redundant are pruned. Liu et al. [38] proposed a visible-hidden hybrid automatic feature engineering (VHAFE) to take into account the interactions of features in the evolution process.

In this article, to address the aforementioned challenges, a novel AutoFE method named HI-MAFE is proposed to automatically design physically meaningful feature sets in conjunction with the specific scene requirements, thereby reducing the dependence on expert experience. HI-MAFE employs a "generation-selection" heuristic strategy to simulate the decision-making process of human experts. Separate Markov decision process (MDP) models are designed for the generation and selection steps. In the generation step, the feature generation task is formalized as a fully cooperative multi-agent DRL (MADRL) problem. Specifically, N agents are constructed, each responsible for generating features from different spectral bands to map the input raw features into a set of candidate features. In the selection step, a feature selection strategy is devised; through iterative screening of the candidate feature set, the optimal feature subset is determined. In addition, the proximal policy optimization (PPO) algorithm

and the asynchronous advantage actor-critic (A3C) algorithm are designed to maximize the expected cumulative reward of the MDP, thereby iteratively optimizing the feature extraction strategy. Finally, spatial features are incorporated to further improve the performance of the hyperspectral remote sensing information processing. The main contributions of this article are summarized as follows.

1) A hyperspectral image (HSI) automatic feature extraction method based on MADRL is proposed, aiming to alleviate the burden of manual SI design by human experts.

2) A heuristic "generation-selection" strategy is employed to mimic the decision-making process of human experts and design specialized DRL frameworks tailored to both the feature generation and selection steps. HI-MAFE incorporates a multi-agent DRL algorithm to accelerate the exploration process in a high-dimensional action space.

3) The experimental results demonstrate that the proposed method outperforms the state-of-the-art feature selection and AutoFE approaches in both feature extraction efficiency and performance.

II. BACKGROUND OF DRL

The objective of DRL is to train an agent on how to act in complex, uncertain environmental states, with the aim of maximizing the expected cumulative reward [39]. A reinforcement learning task can be conceptualized as an MDP, which can be expressed as a tuple (S, A, R, P, and γ). At time step t, the environment is assumed to be in state $s_t \in S$, where S is the state space. After the agent observes state s_t, it chooses an action $a_t \in A$ according to its policy π, where A is an optional action space, and the policy $\pi : S \rightarrow p(A = a|S)$ is a mapping from the state to a probability distribution over actions. The environment outputs the new state $s_{t+1} \sim P(\cdot|s_t, a_t)$ according to the state transition probability function $P : S \times A \rightarrow \Delta(S)$ and returns a scalar reward value $r(s_t, a_t, s_{t+1}) \in R$, where R depicts the distribution of the immediate reward for the chosen action. $\gamma \in [0, 1]$ is a discount factor, where lower values place more emphasis on immediate rewards. In the case of an MDP, the goal of the agent is to identify the optimal policy $\pi : S \rightarrow A$, a mapping of the state space S to the action space A, that maximizes the long-term cumulative discounted rewards [40]

$$J = \mathbb{E}_{\pi, s_0} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right) | a_t = \pi\left(\cdot|s_t\right) \right]. \quad (1)$$

Based on the DRL optimization objective defined by the above formula, under a given policy π, the value function $V^\pi(s)$ can be formally defined. Its mathematical expression can be given as follows:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right) | a_t \sim \pi\left(\cdot|s_t\right), s_0 = s \right]. \quad (2)$$

Given action a, the Q value can be defined as

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right) | a_t \sim \pi\left(\cdot|s_t\right) \right.$$

$$s_0 = s, a_0 = a]. \quad (3)$$

Given the state transition probability distribution $p(s_{t+1}|s_t, a_t)$ and the reward matrix $r(s_t, a_t)$, based on the Bellman equation [41], the following equation holds for all states s_t at any time step t :

$$V^\pi(s_t) = \sum_{a \in A(s_t)} \pi(a|s_t) \sum_{s' \in S} p\left(s' | s_t, a\right) \left[r(s_t, a) + \gamma^{V^\pi}\left(s'\right)\right] \quad (4)$$

where s' represents the successor state of s. By selecting actions that maximize the reward, the optimal state and optimal policy can be obtained

$$V^{\pi^*}(s_t) = \max_a \sum_{s'} p\left(s' | s_t, a\right) \left[r(s_t, a) + \gamma^{V^\pi}\left(s'\right)\right]. \quad (5)$$

Similarly, the Q value for each state–action pair can be computed using the following equation:

$$Q^{\pi^*}(s_t, a_t) = \sum_{s'} p\left(s' | s_t, a_t\right) \left[r(s_t, a_t) + \gamma^{\max}_{a'} Q^{\pi^*}\left(s', a'\right)\right]. \quad (6)$$

By directly learning $Q^{\pi^*}(s_t, a_t)$, the optimal policy π^* can be obtained. This class of methods is commonly referred to as value-based approaches. However, in real-world scenarios, the state transition probability $p(s' | s_t, a_t)$ is typically unknown. To address this issue, a common practice is to learn state values or Q-values through sampling, where the agent collects the state, action, and reward samples by interacting with the environment. Nevertheless, this approach struggles to handle tasks with extremely large state and action spaces. Leveraging a deep neural network allows for a better approximation of the optimal state–action function, enabling more effective learning and generalization in high-dimensional state spaces. Given a deep neural network, the objective of the DRL algorithm can be reformulated as follows:

$$J(\theta) = \mathbb{E}_{a \sim \pi_\theta(\cdot | s), s \sim \rho_{\pi_\theta}} \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t; \theta). \quad (7)$$

In policy-based methods, the agent directly learns a policy to determine the probability of selecting an action given a state. Both the value-based and policy-based methods rely on sampling-based learning to optimize the parameter θ, in order to maximize the objective function $J(\theta)$.

In MADRL, a system can be represented by a six-tuple $(N, S, A, R, P, \text{and } \gamma)$, where N denotes the number of agents, S represents the state space, and $A = \{A_1, \ldots, A_N\}$ is the set of action spaces for all the agents. P denotes the state transition function, R is the reward function, and $O = \{O_1, \ldots, O_N\}$ represents the observation spaces of all the agents. a denotes the joint action vector of all the agents, and a–i denotes the set of actions taken by all the agents except agent i. In addition, τ represents the observation–action history of agent i, while τ refers to the observation–action histories of all the agents. Furthermore, T, S, and A denote the observation–action space, the state space, and the action space, respectively. In a fully observable cooperative multi-agent environment, each agent i observes the global state s_t at time step t and takes an action a_i^t according to its local stochastic policy π_i. The agent then receives a reward r_i^t. In a fully cooperative setting, all
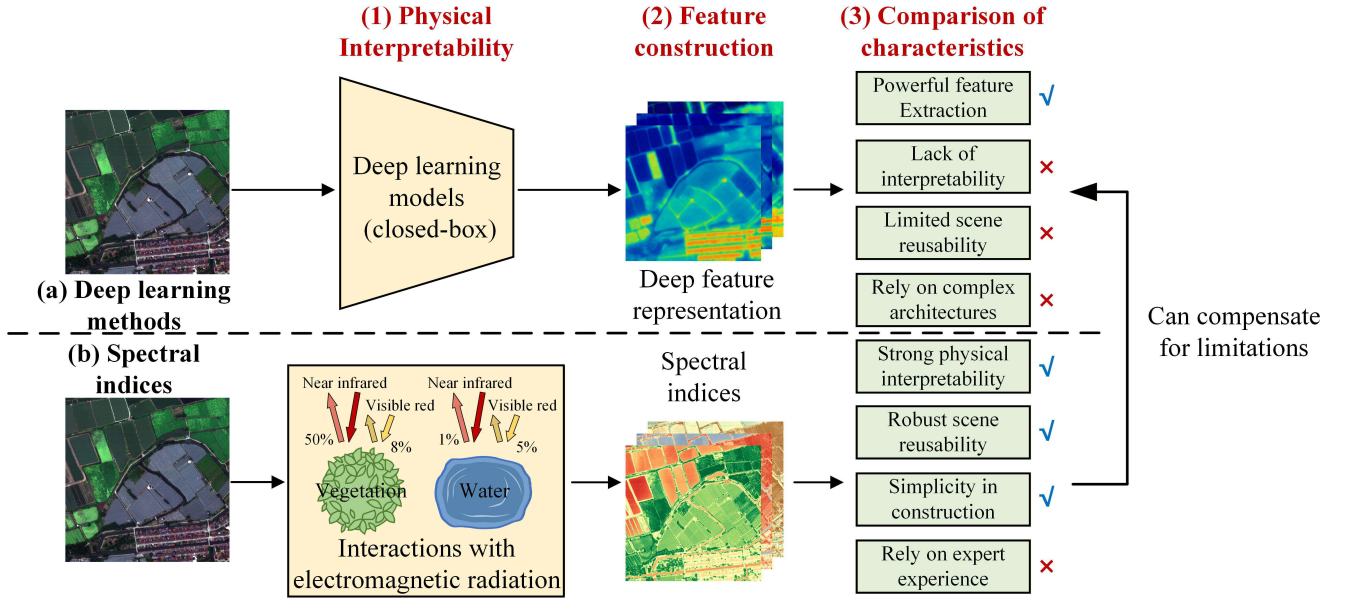
Fig. 1. Illustration of the inherent issues arising from deep learning methods and SIs. (a) Deep learning methods. (b) Spectral indices.

the agents share the same reward at each time step, that is, $r_1^t = \cdots = r_N^t = r^t$. However, if the global state is not fully observable, each agent can only access its local observation.

If agents are homogeneous, parameter sharing can enable more efficient training of their policies. In parameter-sharing training, all the agents can simultaneously leverage the experiences of the other agents, allowing them to learn diverse behaviors.

Taking the policy gradient version of single-agent-based trust region policy optimization (TRPO) with parameter sharing as an example, during the iterative process of the algorithm, decentralized policies are used to sample trajectories from each agent, and advantage values are computed to maximize the following objective:

$$L(\theta) = \mathbb{E}_{o \sim \rho_{\theta_k}, a \sim \pi_{\theta_k}} \left[\frac{\pi_\theta(a \mid o, m)}{\pi_{\theta_k}(a \mid o, m)} A_{\theta_k}(o, m, a) \right]. \quad (8)$$

In this equation, m represents the agent index, and the optimization results are utilized to compute the policy parameter updates.

III. PROPOSED METHOD

HI-MAFE employs a heuristic strategy of "generation-selection." An overview of the proposed HI-MAFE framework is provided in Fig. 2. Let D be a hyperspectral dataset, represented as $D = (X, Y)$, where $X = \{x_1, x_2, \ldots, x_d\}, X \in \mathbb{R}^{n \times d}$ is a matrix of original hyperspectral data, with n rows (instances) and d columns (features), and Y is a vector of the corresponding label values. The action sets A_g and A_s represent the sets of actions for the generation and selection steps, respectively. Given a classification algorithm L with fixed hyperparameters and a cross-validation metric E, in the generation step, the objective of HI-MAFE is as illustrated in (9). Each original feature is associated with an agent that interacts with the environment iteratively. Each agent applies

transformations based on the policy network to generate a new set of candidate features, which are subsequently input into the evaluation model. The evaluation model assesses these features and computes corresponding rewards, which are then relayed back to the agents for updating the policy network. This process is repeated iteratively until either a predefined threshold is reached or convergence is achieved

$$A_g = \underset{A_g}{\operatorname{argmax}} E\left(L\left(A_g(X), Y\right)\right). \quad (9)$$

In the selection step, the objective of HI-MAFE is illustrated in (10). The feature selection task is modeled as a sequential decision-making problem, where the agent determines which feature to select at each time step until an optimal subset of features is identified

$$A_s = \underset{A_s}{\operatorname{argmax}} E\left(L\left(A_s\left(A_g(X)\right), Y\right)\right). \quad (10)$$

A. Generation Step

In the generation step, each agent focuses solely on exploring the optimal transformation rules for its corresponding original feature, effectively addressing the issue of feature explosion. Specifically, the multi-agent feature extraction DRL is further modeled as an MDP, which includes agents, action space, state, state transition, and reward.

1) *Agents:* Compared with structured data, hyperspectral data are characterized by high dimensionality and strong redundancy. During the feature generation process in high-dimensional data, the number of features increases exponentially with the feature order, resulting in a rapid expansion of the feature space, which leads to reduced computational efficiency and greater difficulty in model training; therefore, traditional methods are prone to the problem of feature explosion. To address this issue, HI-MAFE introduces a multi-agent feature extraction mechanism. Each original feature is associated with an
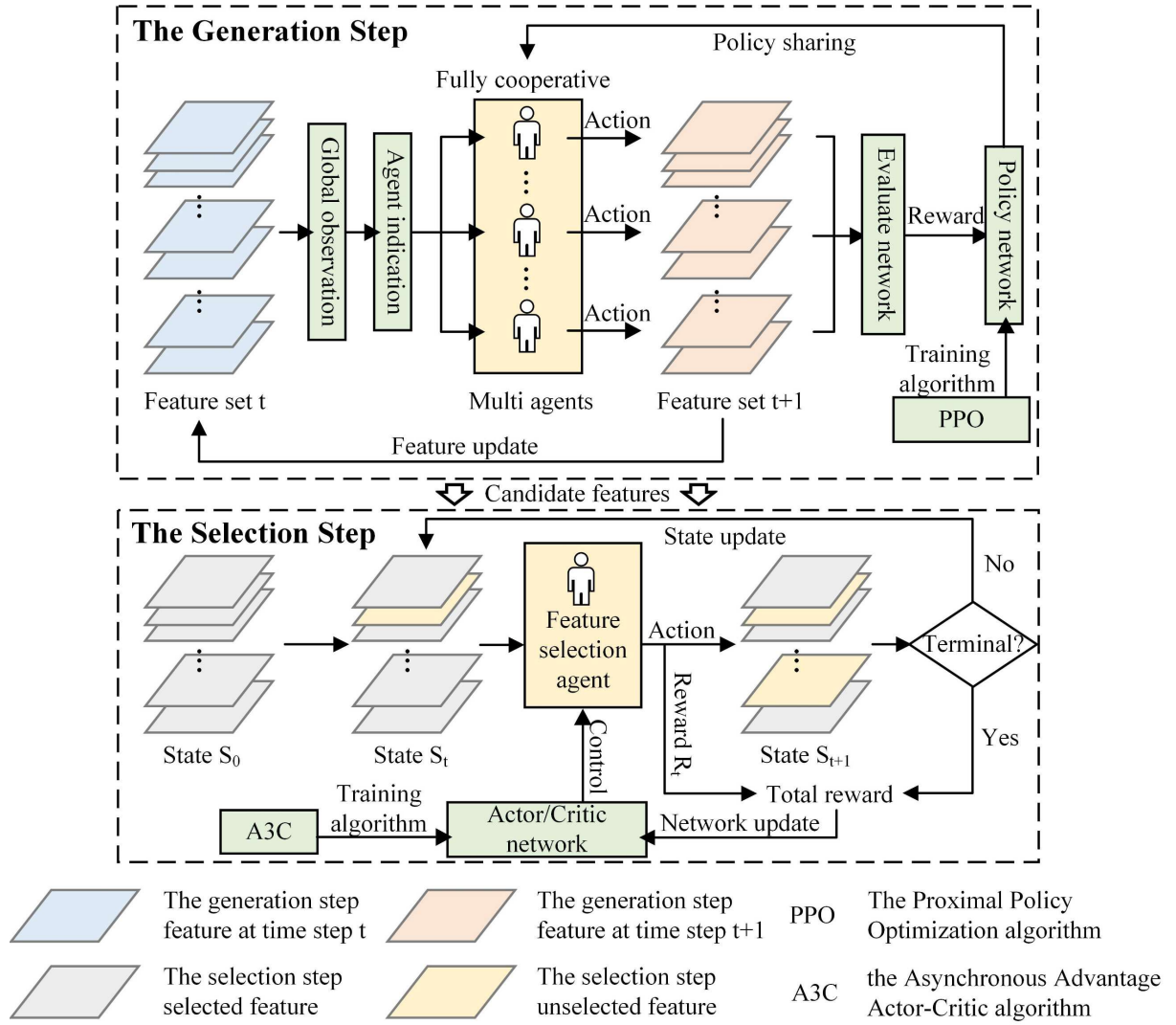
Fig. 2. Overview of the HI-MAFE framework. In the selection step, white indicates unselected features and pink indicates selected features.

individual agent, with the total number of agents equal to the total number of spectral features. All agents share a single policy network, which guides the agents' actions based on the designed policy framework.

2) *Action Space:* In HI-MAFE, operator transformations are employed to generate new features. Based on the number of features involved, operators are categorized as unary operators (logarithm, square, square root, and reciprocal) and binary operators (addition, subtraction, multiplication, and division). In addition, the action set also includes the option of taking no action on the features. Let the number of original features be n_0, the number of binary operators be n_{binary}, the number of unary operators be n_{unary}, and the maximum number of steps be K. The size of the feature space at time step $t(1 \leq t \leq K)$ can be recursively expressed as

$$n_t = n_{t-1}\left(n_{t-1} \cdot n_{\text{binary}} + n_{\text{unary}} + 1\right). \qquad (11)$$

Under a single-agent architecture, the feature space grows exponentially with increasing feature order. A single agent must explore all potential features

combinations simultaneously, resulting in high exploration costs. To mitigate this "feature explosion" problem, we incorporate an efficient search strategy such that each original feature explores only its most promising transformation rules. More specifically, we allocate an independent agent to each original feature, enabling multiple agents to learn their respective optimal transformation strategies in parallel. This ensures adequate exploration of the discriminative feature space. Under this mechanism, the exploration space of each agent can be simplified as

$$n_t = n_{t-1} \cdot n_{\text{binary}} + n_{\text{unary}} + 1. \qquad (12)$$

HI-MAFE can add other operators, depending on specific application requirements. The actions taken by all the agents at time step t can be represented as

$$A_t = \{a_{1,t}, \ldots, a_{n,t}\}. \qquad (13)$$

3) *State:* At time step t, $X_t = \{x_{1,t}, \ldots, x_{n,t}\}$ denotes the set of features generated by all the agents, where $x_{i,t} = a_{i,t-1}(x_{i,t-1})$ represents the new feature generated
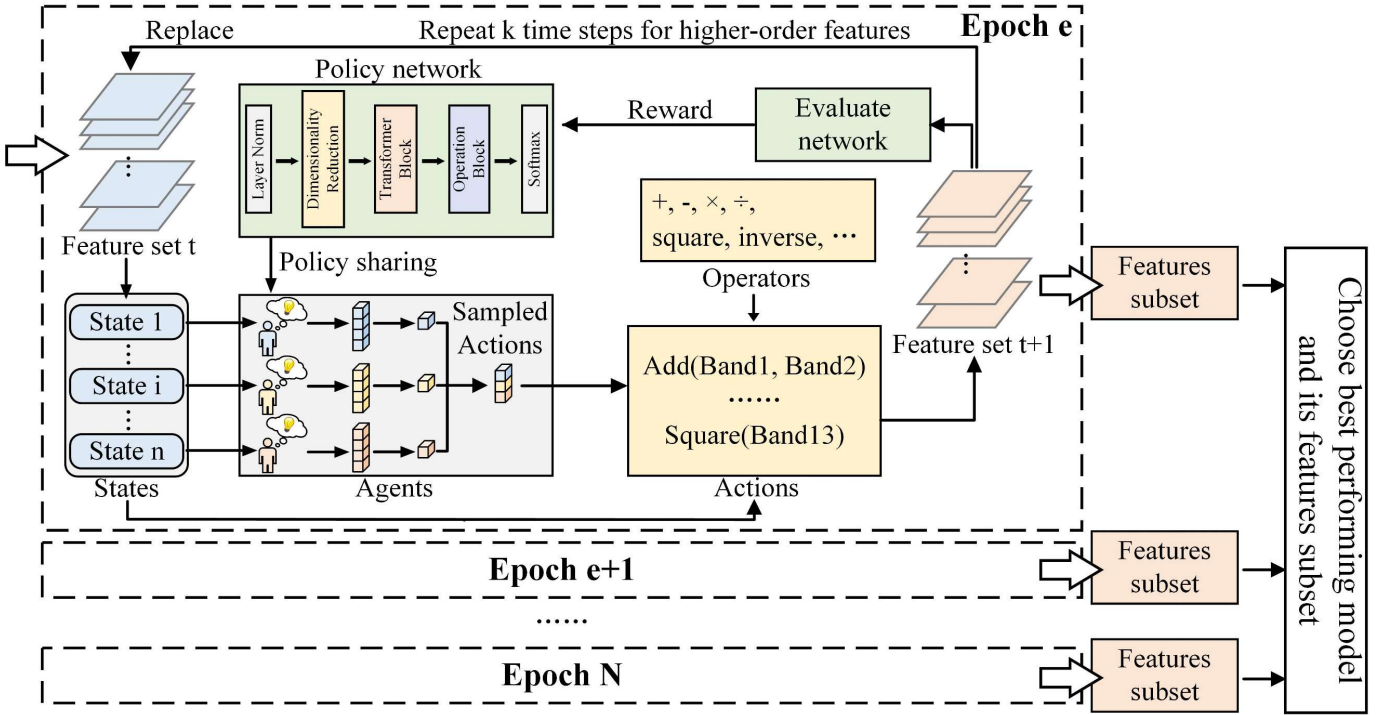
Fig. 3. Architecture of the generation step.

by applying the action $a_{i,t-1}$ to the original feature $x_{i,t-1}$. Each agent has access to the observations and action information of all the other agents. To ensure that the agents can learn distinct policies, an agent-specific indicator signal is added to the states, which is called "agent indication." The state of agent n_i at time step t is represented as

$$s_{i,t} = \{X_t, Y, x_{i,t}\}. \tag{14}$$

4) *State Transition:* After all the agents take an action, the state will be updated based on the collective actions taken.

5) *Reward:* The average performance from k-fold cross-validation is utilized as the reward. To ensure training stability, the model is penalized based on suboptimal results, as defined by the reward function in (15) [36], where $\bar{E}_t$ represents the average result from the k-fold cross-validation at time step t, and $E_{t,k}$ represents the evaluation result from the kth fold at step t. In the fully cooperative multi-agent reinforcement learning setting, the reward is evenly distributed among all the agents

$$R_t = \bar{E}_t + E_{t,\text{diff}} \tag{15}$$

$$E_{t,\text{diff}} = \sum_k \min\left(0, E_{t,k} - \bar{E}_{t-1}\right). \tag{16}$$

In hyperspectral datasets containing hundreds of features, the dimensionality of the action space significantly affects the learning efficiency, leading to instability during model training and poor convergence. PPO is a robust reinforcement learning algorithm that ensures stable training in complex environments with large action spaces, and has been successfully applied in various domains for training reinforcement learning agents [42]. In the proposed framework, the PPO algorithm is employed for the feature generation task. This approach allows the framework to sample multiple action plans in each epoch, which is then allocated to multiple threads for evaluation. Subsequently, the policy gradient is computed to update the network, thereby reducing the sample complexity. The PPO objective function can be mathematically expressed as follows [42]:

$$L^{\text{CLIP}}(\theta) = E_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{17}$$

where $r_t(\theta) = (\pi_\theta(a_t|s_t))/(\pi_{\theta_{\text{old}}}(a_t|s_t))$ represents the probability ratio of the new policy π_θ to the old policy $\pi_{\theta_{\text{old}}}$. The term $\hat{A}_t$ denotes the estimated advantage at time t, which indicates how much better the action taken is, compared to the average action. The clipping mechanism, governed by the hyperparameter ϵ, serves to limit the degree of policy updates, thereby preventing excessive changes that could destabilize the learning.

Moreover, a policy network is designed to guide the agents in action selection, as illustrated in Fig. 3. All the agents are enabled to share a common policy network, which effectively reduces the computational burden and memory requirements. This approach minimizes the complexity associated with maintaining individual policy networks for each agent, while simultaneously enhancing the coordination and cooperation among the agents. Leveraging a shared policy network yields a more cohesive learning process, enabling the agents to better align their actions in pursuit of collective objectives. The policy network comprises the following key components: 1) a layer normalization layer to stabilize the training of the neural

network; 2) a dimensionality reduction block that reduces the feature vector to a fixed length, enabling the network to handle varying sizes of feature sets while decreasing the network complexity; 3) a transformer block that learns the complex relationships among the different features; and 4) an operation block followed by a softmax layer that maps the relevant information of features to corresponding action probabilities.

Algorithm 1 Generation Step

Input: Raw feature set $X_0 = \{x_{1,0}, \ldots, x_{n,0}\}$, policy network π, metric E, preselected machine learning model L, label Y
    Parameter: Total epochs N, maximum steps K, number of workers W
Output: Action A and new feature set $\hat{X}$
1: **while** epoch $e \in [0, N]$ **do**
2:     **while** worker index $w \in [0, W]$ **do**
3:         **while** step $t \in [0, k]$ **do**
4:         $X = X_t$
5:         Update $S_t = \{s_{1,t}, \ldots, s_{n,t}\}$ via (14).
6:         Perform actions $A_t = \{a_{1,t}, \ldots, a_{n,t}\}$ according to the policy $\pi(A_t|S_t)$
7:         $X_{t+1} = \{a_{1,t}(x_{1,t}), \ldots, a_{n,t}(x_{n,t})\}$
8:         Evaluate X_{t+1} by $E(L(X_{t+1}, Y))$
9:         $t = t + 1$
10:        **end while**
11:        $w = w + 1$
12:     **end while**
13:     Calculate the reward R via (15)
14:     Carry out a gradient descent step via (17)
15:     Update the policy network π
16:     $e = e + 1$
17: **end while**

The overall training algorithm of the generation step is described in Algorithm 1.

B. Selection Step

After the generation step, a new set of candidate features $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_n\}$ is obtained. The objective of the selection step is to further identify the most effective features from this candidate set, shifting the model's focus from "diversity exploration" to "optimal decision-making." The selection step can be conceptualized as a sequential decision-making problem for an agent, where the agent must determine which features to select at different time steps until an optimal feature combination is identified. This problem is formulated as an MDP, encompassing the action space, state, state transition, and reward.

1) *Action Space:* In our case, the action of the agent is to choose a feature at each time step. The policy of the agent determines which feature to choose. The size of the agent's action space equals to the number of candidate features.
2) *State:* $S_t = \{s_1, \ldots, s_n\}$ represents the current feature selection state at time step t, where $s_i = \{0, 1\}, 1 \le i \le n$. $s_i = 1$ indicates that the ith feature has been selected, while $s_i = 0$ represents that it has not been selected.

Algorithm 2 Selection Step

Input: Raw feature set $\hat{X}$, metric E, pre-selected machine learning model L, label Y
    Parameter: Total epochs N, maximum number of features M, number of parallel environments W
Output: new feature set $\tilde{X}$
1: Initialize all the parameters θ of the global network
2: **for** $w = 1$ to W **do**
3:     Initialize the parameters θ_w of the w-th local network
4: **end for**
5: Perform the following steps asynchronously for all local networks, taking the w-th local network as an example
6: **while** epoch $e \in [0, N]$ **do**
7:     $\theta_w = \theta$, initialize S
8:     **while** $t \in [0, M]$ **do**
9:       Perform action a according to the policy network
10:       Calculate S_{t+1} via (18).
11:       Calculate the reward r_t via (19).
12:     **end while**
13:     $\theta = \theta_w, R = r_M$
14:     **for** $i \in [M - 1, 0]$ **do**
15:       $R \leftarrow r_i + \gamma R$
16:       Accumulate gradients of θ'_a: $d\theta_a \leftarrow d\theta_a + \nabla_{\theta'_a} log\pi(a_i|s_i; \theta'_a)(R - V(s_i; \theta'_v))$
17:       Accumulate gradients of θ'_v: $d\theta_v \leftarrow d\theta_v + \partial(R - V(s_i; \theta'_v))^2/\partial\theta'_v$
18:     **end for**
19:     Perform asynchronous update of θ
20: **end while**

---

3) *State Transition:* At time step t, assuming that the agent selects the ith feature, donated as $a_t = i$, if the ith feature has not been selected, the next state will be updated. Otherwise, the state remains unchanged. If the maximum number of selectable features has been reached, the action terminates. The state transition function is defined as follows:

$$S_{t+1} = \begin{cases} S_t, & \text{if } s_{a_t} = 1 \\ S_t + a_t, & \text{if } s_{a_t} = 0 \\ \text{Terminal}, & \text{if } \text{sum}(S_t) = \text{featurenum}. \end{cases} \quad (18)$$

4) *Reward:* At time step t, the number of selected features reaches the maximum limit, resulting in a new feature set $\tilde{X}$. The chosen subset of features is evaluated using k-fold cross-validation. To prevent the agents from falling into a cycle of repetitive selections, a penalty factor α and a reward factor β are incorporated into the reward function. The reward function is defined as follows:

$$r_t = \begin{cases} \alpha, & \text{if } s_{a_t} = 1 \\ \beta, & \text{if } s_{a_t} = 0 \\ -\left(\bar{E}\left(\tilde{X}\right) - \bar{E}\left(\hat{X}\right)\right), & \text{if } \text{sum}(S_t) = \text{featurenum}. \end{cases} \quad (19)$$

In the selection step, the A3C algorithm is used to solve the feature selection problem. One of the core strengths of A3C is its asynchronous architecture, which allows multiple workers
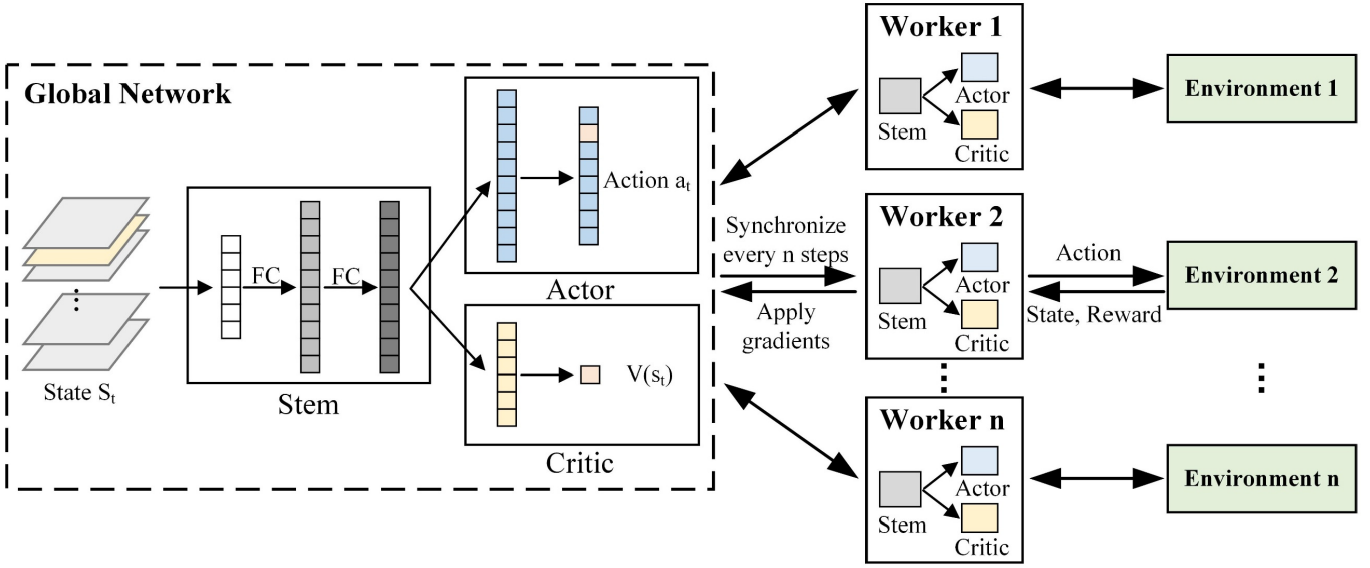
Fig. 4. Architecture of the selection step.

to explore the environment simultaneously. This parallelism accelerates the learning process. The actor updates this policy based on the feedback received from the critic. The critic's role is to minimize the mean squared error between the predicted value and the actual return. A3C utilizes an advantage function, which can be defined as

$$A(s,a) = Q(s,a) - V(s) \tag{20}$$

where $Q(s,a)$ is the action–value function representing the expected return for taking action a in state s.

The gradient of the objective function for the policy network can be updated as follows:

$$\nabla J(\theta) = E_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla \log \pi(a_t|s_t;\theta) A_t(s_t, a_t) \right]. \tag{21}$$

The selection step consists of a stem network, an actor network, and a critic network, as shown in Fig. 4. The architectures of the networks are all 1-D fully connected networks.

The overall training algorithm of the selection step is described in Algorithm 2.

C. Spatial Feature Extraction

Compared with structured data, another challenge of hyperspectral data is that it contains both spatial and spectral information. Relying solely on spectral information often proves insufficient for achieving an optimal classification accuracy. To enhance the classification results, spatial information is integrated with spectral information.

Mathematical morphology is one of the widely utilized techniques for extracting spatial features [43]. Given an input image X and a structuring element B, the mathematical formulation of erosion is defined as shown in (22). $\ominus$ denotes the erosion operation

$$E(X) = X \ominus B = \{z|(B)_Z \subseteq X\}. \tag{22}$$

The dilation operation can be mathematically expressed as shown in (23). $\oplus$ denotes the dilation operation

$$E(X) = X \oplus B = \{z|(B)_Z \cap X \neq 0\}. \tag{23}$$

Opening combines erosion followed by dilation. The opening operation is defined as

$$O(X) = (X \ominus B) \oplus B. \tag{24}$$

Closing involves dilation followed by erosion. The closing operation is defined as

$$O(X) = (X \oplus B) \ominus B. \tag{25}$$

The gray-level co-occurrence matrix (GLCM) is one of the most widely used techniques for computing second-order texture measures. The definitions of some of the important features extracted from the GLCM are provided in Table I.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the experimental results obtained on three public HSI datasets are presented to validate the effectiveness of the proposed method. In Section IV-A, the three HSI datasets are introduced. Sections IV-B and IV-C describe the experimental setup and the classification results, respectively. Sections IV-D–IV-F discuss the final extracted spectral features, the sensitivity to the number of selected features, and the algorithm's efficiency, respectively. Finally, in Section IV-G, the ablation studies performed to verify the contribution of the different components in the HI-MAFE framework are described.

A. Data Description

In the experiments, we utilized three HSI datasets: Pavia University, Houston 2013, and QingPu. A detailed description of each dataset is provided below.

1) *Pavia University:* The Pavia University dataset was captured by the reflective optics system imaging spectrometer (ROSIS) over the University of Pavia in

TABLE I
DEFINITIONS OF THE FEATURES EXTRACTED FROM THE GLCM

GLCM feature	Formula
Mean	$\mu = \sum_i \sum_j i \cdot P(i,j)$
Variance	$\sigma^2 = \sum_i \sum_j P(i,j) \cdot (i-\mu)^2$
Homogeneity	$\sum_i \sum_j \dfrac{P(i,j)}{1+(i-j)^2}$
Contrast	$\sum_i \sum_j (i-j)^2 \cdot P(i,j)$
Dissimilarity	$\sum_i \sum_j |i-j| \cdot P(i,j)$
Entropy	$-\sum_i \sum_j P(i,j) \cdot logP(i,j)$
Second moment	$\sum_i \sum_j P(i,j)^2$
Correlation	$\dfrac{\sum_i \sum_j (i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j}$

TABLE II
DETAILED CLASSES AND NUMBERS OF THE TRAINING AND TEST SAMPLES ON THE PAVIA UNIVERSITY DATASET

Class	Class name	Training	Testing	Total
C1	Asphalt	332	6299	6631
C2	Meadows	933	17716	18649
C3	Gravel	105	1994	2099
C4	Trees	153	2911	3064
C5	Painted metal sheets	67	1278	1345
C6	Bare Soil	251	4778	5029
C7	Bitumen	67	1263	1330
C8	Self-Blocking Bricks	184	3498	3682
C9	Shadows	47	900	947
	Total	2139	40637	42776

TABLE III
DETAILED CLASSES AND NUMBERS OF THE TRAINING AND TEST SAMPLES ON THE HOUSTON 2013 DATASET

Class	Class name	Training	Testing	Total
C1	Healthy grass	138	1236	1374
C2	Stressed grass	145	1309	1454
C3	Synthetic grass	80	715	795
C4	Tress	126	1138	1264
C5	Soil	130	1168	1298
C6	Water	34	305	339
C7	Residential	148	1328	1476
C8	Commercial	135	1219	1354
C9	Road	155	1399	1554
C10	Highway	142	1282	1424
C11	Railway	157	1409	1566
C12	Parking Lot 1	143	1286	1429
C13	Parking Lot 2	63	569	632
C14	Tennis Court	51	462	513
C15	Running Track	80	718	798
	Total	1727	15543	17270

TABLE IV
DETAILED CLASSES AND NUMBERS OF THE TRAINING AND TEST SAMPLES ON THE QINGPU DATASET

Class	Class name	Training	Testing	Total
C1	Asphalt road	162	7937	8099
C2	Greenhouse	29	1431	1460
C3	Cement road	51	2497	2548
C4	Farmland	536	26270	26806
C5	Water	490	24019	24509
C6	Bulrush	37	1819	1856
C7	Ligustrum	248	12143	12391
C8	Elaeocarpus sylvestris	112	5467	5579
C9	Camptotheca acuminata	300	14687	14987
C10	Goldenrain tree	166	8140	8306
C11	Camphor tree	247	12090	12337
	Total	2378	116500	118878

northern Italy. After removing 12 noisy bands, the dataset consists of 610 × 340 pixels and 103 spectral bands, covering a wavelength range of 430–860 nm. The spatial resolution is 1.3 m per pixel. The dataset includes nine land-cover classes. In this study, 5% of the labeled samples were randomly selected as training data. The number of samples are listed in Table II.

2) *Houston 2013:* This dataset were initially utilized in the 2013 IEEE GRSS Data Fusion Contest. It consists of 349 × 1905 pixels and comprises 144 spectral bands, covering a wavelength range from 380 to 1050 nm. The data include a total of 15 classes. To facilitate the training and testing process, 10% of the labeled samples were randomly selected for use as training data, while the remaining samples were reserved for testing purposes. The number of samples are listed in Table III.

3) *QingPu:* The QingPu dataset is a representative subset selected from The QingPu-HSI dataset [44], acquired on June 16, 2022, by the airborne multimodality imaging spectrometer (AMMIS) developed by the Shanghai Institute of Technical Physics at the Chinese Academy

of Sciences, Beijing, China. The dataset consists of 396 × 1448 pixels with 256 spectral bands, covering a wavelength range of 400–1000 nm. After removing five defective bands, 251 spectral bands remained available for analysis. Prior to analysis, the dataset was preprocessed through dark current correction, radiometric calibration, and geometric calibration [45]. The dataset includes 11 different land-cover categories. In this study, 2% of the labeled samples were randomly selected for training. The number of samples are listed in Table IV.

B. Experimental Settings

To validate the feature extraction capabilities of the proposed HI-MAFE algorithm, comparative experiments were conducted with eight different methods. These included five AutoFE algorithms and three DRL feature selection algorithms. Among the different methods, the random method generates new features by randomly applying transformations to each original feature. AutoFeat [46] is a widely used AutoFE toolkit in Python that constructs features using an "expansion–reduction" framework. FETCH [36] is a DRL-based end-to-end AutoFE framework that can achieve a

TABLE V
CLASSIFICATION RESULTS OF THE DIFFERENT METHODS WITH THE THREE METRICS ON THE DIFFERENT DATASETS

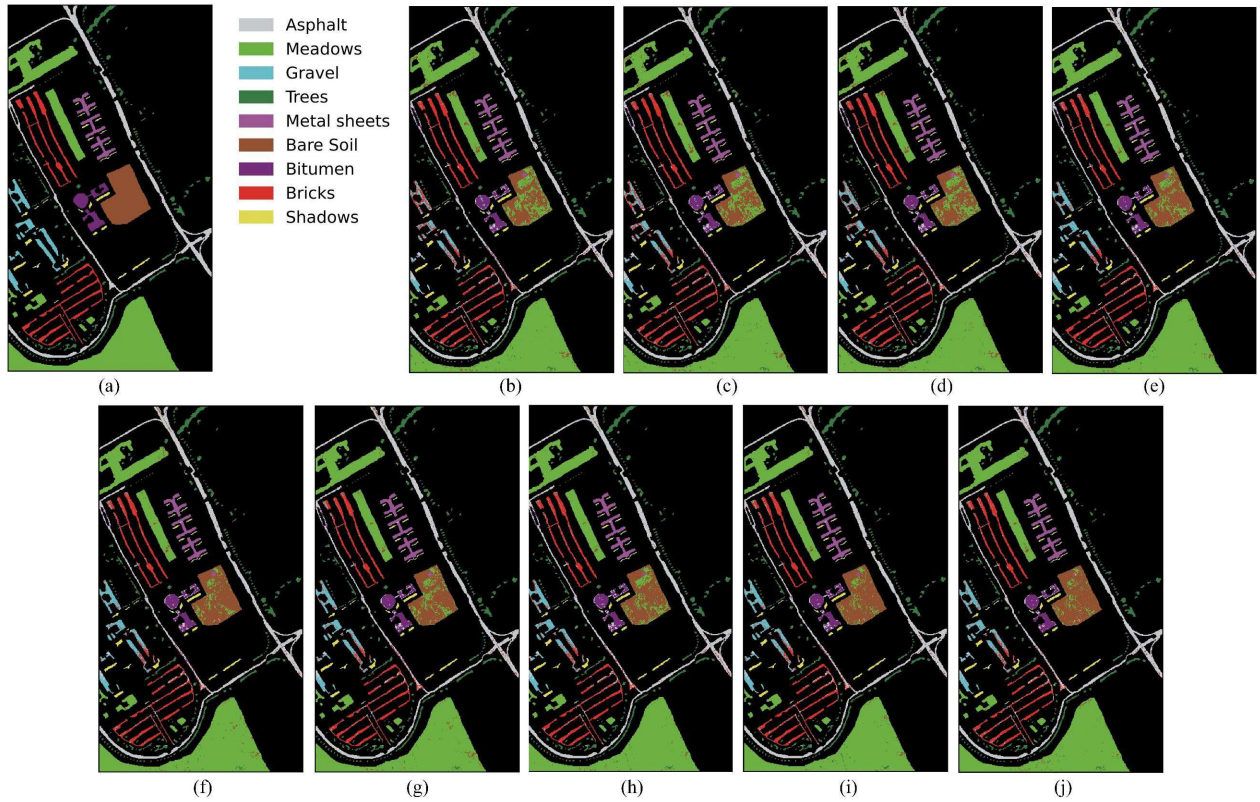| Dataset	Metric	DRLBS	RLSFR-cv	MH-DRL	Random	AutoFeat	NFS	DIFER	FETCH	HI-MAFE
Pavia University	OA (%)	88.50	88.48	89.28	91.97	93.04	92.07	91.84	_93.56_	**93.64**
	AA (%)	87.87	88.09	89.07	91.92	92.17	91.48	91.56	_93.12_	**93.29**
	Kappa × 100	84.55	84.51	85.60	89.24	90.72	89.40	89.05	_91.41_	**91.53**
Houston 2013	OA (%)	88.66	88.70	89.26	92.37	_93.00_	92.78	92.77	92.85	**93.64**
	AA (%)	90.48	90.51	91.00	93.63	_94.34_	93.95	93.97	94.05	**94.80**
	Kappa ×100	87.73	87.78	88.39	91.75	_92.43_	92.19	92.18	92.27	**93.12**
QingPu	OA (%)	90.66	90.48	90.80	92.74	91.21	91.59	_93.08_	92.96	**93.34**
	AA (%)	88.28	88.23	88.36	90.88	89.10	89.73	90.80	_91.11_	**92.11**
	Kappa × 100	89.09	88.88	89.26	91.52	89.74	90.18	_91.92_	91.78	**92.22**

Fig. 5. Visual classification maps of all compared methods on the Pavia University dataset using RF classifier. (a) Ground truth. (b) DRLBS. (c) RLSFR-cv. (d) MH-DRL. (e) Random. (f) AutoFeat. (g) NFS. (h) DIFER. (i) FETCH. (j) HI-MAFE.

state-of-the-art performance. DIFER [47] introduces a differentiable AutoFE method that utilizes an encoder–decoder framework, converting features into function strings at each step. NFS [35] is a DRL-based AutoFE method inspired by neural architecture search (NAS), exploring the feature space with an RNN controller and automating the feature construction and selection process. Among the DRL feature selection methods, DRLBS [11] was the first to transform the feature selection problem of hyperspectral imagery into a DRL problem. RLSFR-cv [12] explores the inherent relationships between hyperspectral features by introducing two spectral feature evaluation methods. MH-DRL [13] is based on a multi-

agent DRL-based approach, combined with hybrid teacher guidance, to address the hyperspectral band selection problem.

The experiments were performed on a server equipped with an Nvidia H800 GPU and an Inter[1] Xeon[1] Gold 6442Y processor with 96 cores. The proposed method was implemented using the PyTorch framework in Python.

In the experiments, all the methods were optimized using fivefold cross-validation on the training set, and the final model performance was evaluated on the test set. Three metrics are employed here for the evaluation: overall accuracy (OA),
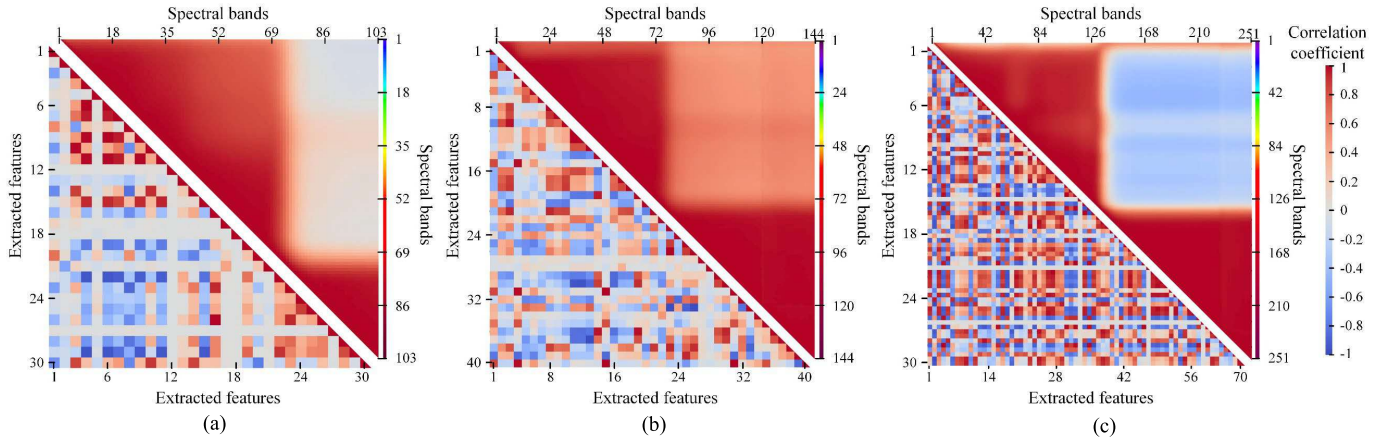
[1]Registered trademark.

Fig. 6. Correlation matrices of the original features and the features extracted by HI-MAFE on the different datasets. (a) Pavia University. (b) Houston 2013. (c) QingPu.

average accuracy (AA), and the Kappa coefficient (Kappa). The classification assessments were conducted using random forest (RF). For the RF classifier, the number of decision trees was set to 200. To ensure consistency, the same feature operators were utilized across the five AutoFE algorithms, with the maximum feature order set to 2. All the methods were configured to use the default parameters whenever possible. To reduce computational and memory burdens, certain parameters for some methods were adjusted. Specifically, we set the parallel sampling number for FETCH to 5 for each iteration. In the initialization step of DIFER, we set the initial number of randomly selected features to 128. For NFS and AutoFeat, we applied RF to perform feature importance selection on the original dataset, identifying the top 50 important features. All the methods, including HI-MAFE, utilized the spatial features in the same manner.

In HI-MAFE, we configured the training process for a total of 400 epochs, with the first 100 epochs designated for the generation step and the remaining 300 epochs for the selection step. During the generation step, the learning rate was set to 0.0001, the discount factor to 0.95, the parallel sampling size to 5, and the transformer parameters to $d_{\text{model}} = 128, n_{\text{head}} = 8$. In the selection step, the learning rate was adjusted to 0.001, the discount factor was set to 0.999, the reward coefficient for selecting new bands was set to 0.1, and the penalty coefficient for selecting duplicate bands was set to -0.1. In addition, the number of parallel environments was established as 8.

C. Experimental Results

This section presents a detailed comparison of the OA, mean accuracy, and Kappa coefficient of the various methods across the three HSI datasets: Pavia University, Houston 2013, and QingPu. Consistent with previous studies, we set the final number of features to 30, 40, and 70, respectively, for these datasets [13]. It is noteworthy that the random, AutoFeat, NFS, DIFER, and FETCH methods do not output a fixed number of features; instead, the final feature count is

adaptively determined by the algorithms. As shown in Table V, the performance of the band selection methods is inferior to that of the feature engineering algorithms. This discrepancy arises because the band selection methods often eliminate a substantial number of bands, leading to significant information loss. Among the various band selection algorithms evaluated, MH-DRL, which employs a hybrid teacher-guided multi-agent DRL framework, provides agents with rich external knowledge. This enhances the exploration process and improves the agents' autonomous learning capabilities, resulting in a superior performance, compared to the other band selection algorithms. The random method demonstrates a poor performance, mainly because it relies entirely on randomly selecting transformation functions, which weaken the stability of the model's prediction. AutoFeat uses an "expansion–reduction" strategy but encounters the issue of feature explosion, which makes it less effective for high-dimensional hyperspectral data that include hundreds to thousands of bands. Consequently, its predictive performance is suboptimal. In contrast, NFS constructs an RNN controller for each feature, allowing for efficient parallel execution of transformations through a robust search algorithm. This effectively addresses the feature explosion issue, which is a key factor in its performance advantage over AutoFeat. However, processing high-dimensional data incurs substantial computational costs. To mitigate the feature space, NFS first selects from the original features, which unfortunately results in considerable information loss. DIFER, which is the first differentiable AutoFE method, incorporates a tree structure for better feature representation, which enhances its performance, relative to the other baseline methods. Nonetheless, DIFER evaluates the impact of individual features without considering the overall performance of the generated feature set. The previous NAS-like AutoFeat frameworks operate with the data-unobserved paradigm, significantly deviating from the approach taken by human experts. In contrast, FETCH establishes a novel DRL-based AutoFE framework, characterized by a fundamentally different data-driven MDP setup that simulates human expertise, thereby outperforming the other comparative methods. In addition, the proposed HI-MAFE method exhibits a superior performance
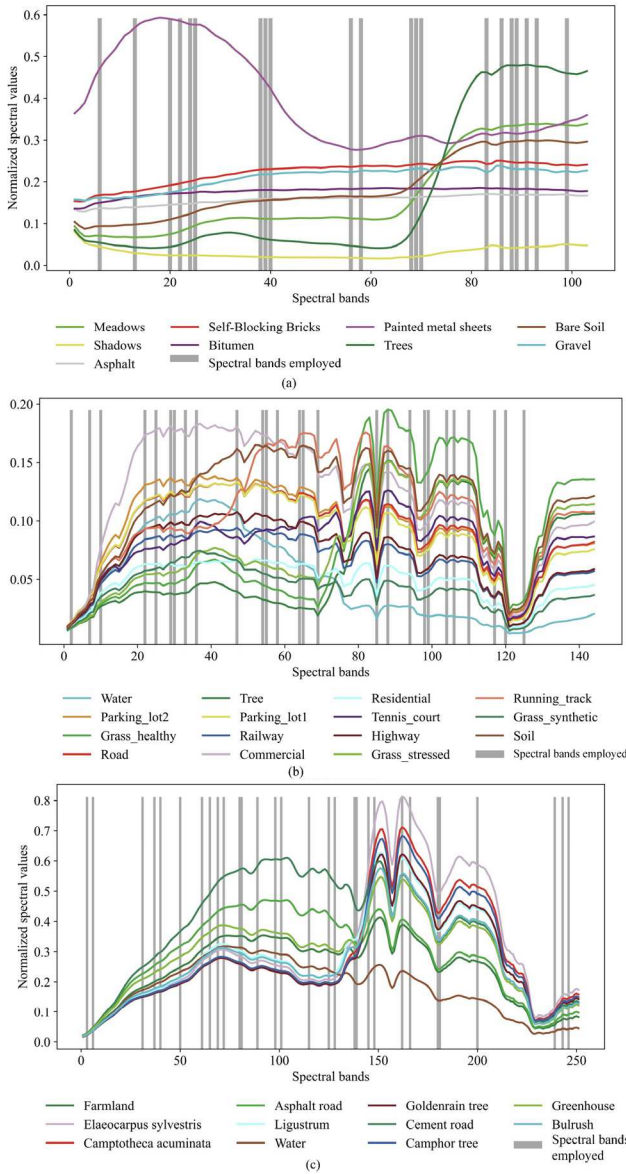
Fig. 7. Visualization of the bands employed by the proposed method on (a) Pavia University dataset, (b) Houston 2013 dataset, and (c) QingPu dataset. We also show the average spectral signature of each class.

TABLE VI
CORRELATION AND ENTROPY RESULTS OF HI-MAFE AND THE BASE METHOD ON THE DIFFERENT DATASETS

Dataset	Metric	Base	HI-MAFE
Pavia	Correlation	0.5881	0.2793
University	Entropy	11.2312	15.0229
Houston	Correlation	0.7787	0.3662
2013	Entropy	13.0689	18.3265
QingPu	Correlation	0.6053	0.5102
	Entropy	11.4361	17.0853

information redundancy, suggesting improved effectiveness of the feature extraction. Information entropy is a crucial metric for quantifying the informational content of images [49]. The entropy value of each feature reflects the amount of information it contains, with higher entropy indicating richer information content. Table VI presents the correlation coefficients and information entropy of HI-MAFE and the base method (original features) across the various HSI datasets. It can be observed that the information entropy increases by 33.76%, 40.23%, and 49.40% across the three datasets, respectively, while the correlation coefficients decrease by 52.51%, 52.97%, and 15.71%. These results indicate that the extracted features exhibit low correlation and contain a substantial amount of information. Furthermore, Fig. 6 illustrates the changes in the correlation coefficients before and after feature extraction on the different datasets.

E. Analysis of the Employed Spectral Bands

To provide a clearer visualization of the features employed by HI-MAFE, the usage of spectral bands across the three different datasets is presented in Fig. 7. To prevent stacking caused by an excessive number of bands, the final output feature count was fixed at 10. In HSI data, adjacent bands typically exhibit higher correlations, leading to information redundancy. The figures reveal that HI-MAFE tends to favor bands with lower correlation. Notably, in the spectral range with higher information entropy, the distribution of the bands selected by HI-MAFE is denser.

F. Sensitivity to the Numbers of Features

In this section, we describe how the impact of varying the number of output features on the classification results was investigated. For the QingPu dataset, the number of features was restricted to the range of [5, 100]. For the Pavia University and Houston 2013 datasets, the feature counts were limited to [5, 50] and [5, 70], respectively. We present the OA for the different feature counts in Fig. 8. It should be noted that the AutoFE methods are unable to fix the number of output features. Consequently, the number of features is determined adaptively by the algorithm. To ensure the consistency and reliability of the experimental outcomes, only the spectral features were utilized in the experiments conducted for each method. The experimental results demonstrate that the OA of several of the methods exhibits a gradual increase with the number of features, reaching a plateau after a certain number

across all the datasets, substantiating the effectiveness of the proposed approach.

Furthermore, as illustrated in Fig. 5, the visual classification results obtained from the Pavia University dataset demonstrate that the proposed HI-MAFE method yields more accurate classification results, compared to its competitors. It is noteworthy that bare soil and meadows are two categories within the Pavia University dataset that are particularly prone to confusion.

D. Feature Quality Assessment

In this study, in addition to evaluating the feature extraction results through classification outcomes, we also assessed the feature quality based on the correlation among features and their information entropy. The correlation coefficient serves as a measure of the relationship between variables [48]; generally speaking, a lower correlation among features indicates reduced
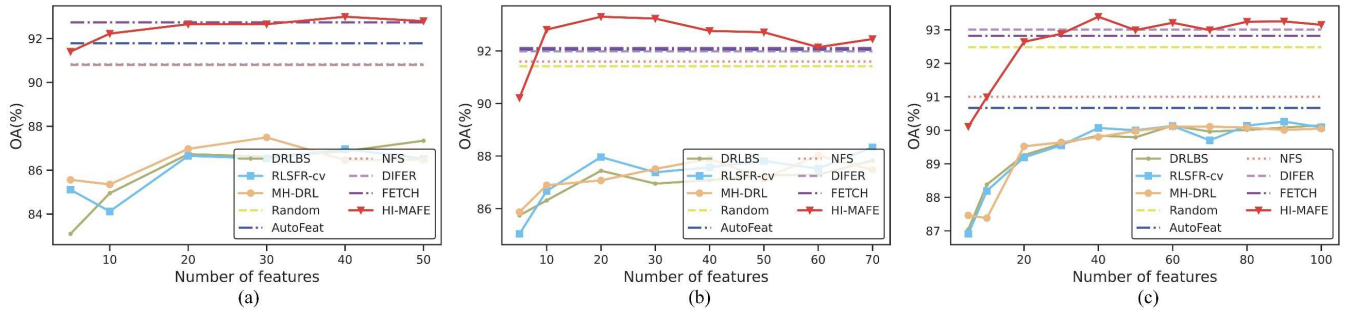
Fig. 8. Influence of the different numbers of features on (a) Pavia University dataset, (b) Houston 2013 dataset, and (c) QingPu dataset.

TABLE VII
TIME EFFICIENCY COMPARISON OF HI-MAFE WITH THE OTHER AUTOFE
METHODS ON THE DIFFERENT DATASETS

Dataset	Execution Time(m)				
	AutoFeat	NFS	DIFER	FETCH	HI-MFE
Pavia University	202.87	1097.56	252.13	71.25	23.60
Houston 2013	328.28	924.94	298.30	83.37	28.99
QingPu	142.50	1216.29	686.01	209.03	45.72

TABLE VIII
ABLATION EXPERIMENTS ON HI-MAFE WITH THE DIFFERENT DATASETS

Dataset	Metric	Components			
		Gen	Gen+Sel	Spa	Gen+Sel+Spa
Pavia University	OA(%)	92.83	92.65	67.92	93.64
	AA(%)	92.41	92.33	66.97	93.29
	Kappa×100	90.44	90.20	54.42	91.53
Houston 2013	OA(%)	92.20	92.76	44.99	93.64
	AA(%)	93.25	93.74	45.72	94.80
	Kappa×100	91.56	92.16	40.51	93.12
QingPu	OA(%)	93.14	92.99	67.91	93.34
	AA(%)	91.81	91.78	53.96	92.11
	Kappa×100	91.99	91.81	62.67	92.22

of features. The OA of HI-MAFE consistently outperforms that of the feature selection methods, as HI-MAFE is capable of extracting more comprehensive feature information. In addition, HI-MAFE requires a minimal number of features to achieve a high level of accuracy. For example, on the Houston 2013 dataset, HI-MAFE achieves a high accuracy with a feature count of 10. Overall, HI-MAFE demonstrates a superior performance and stability, achieving a high accuracy with a smaller number of features.

G. Time Efficiency Comparison

This section presents a comparative analysis of the time efficiency of the proposed method. Table VII displays the total running times of the different AutoFE methods on the aforementioned datasets. The results indicate that HI-MAFE is significantly more efficient than AutoFeat, DIFER, and FETCH. Among these methods, NFS builds an RNN controller for each original feature to learn the optimal feature transformations. During the training process, NFS uses RF to evaluate the gain brought by each feature. Therefore, its training speed is constrained by the efficiency of the evaluation algorithm. AutoFeat applies all possible transformation functions or randomly sampled transformations to the original features, resulting in considerable time consumption. The performance of the DIFER and FETCH algorithms is significantly affected by the feature dimensionality, with their runtime on the QingPu dataset being more than twice as long as that on the Houston 2013 dataset. FETCH employs single-agent DRL for feature extraction, which undermines the overall performance of AutoFE. Compared to the

suboptimal algorithms, the HI-MAFE method demonstrates feature extraction efficiency improvements of 66.88%, 65.23%, and 78.13% across the three datasets, respectively. These results indicate that the HI-MAFE approach can effectively handle the high dimensionality of hyperspectral data, exhibiting a superior performance in terms of temporal efficiency.

H. Ablation Experiments on HI-MAFE

This section describes how we verified the effectiveness of the generation step, selection step, and spatial features through ablation experiments. Table VIII lists the classification results for each component. The results demonstrate that, during the generation step, the HI-MAFE method effectively extracts candidate spectral features while achieving a high accuracy. In the selection step, it further reduces the feature redundancy while maintaining the classification accuracy, and the incorporation of spatial features enhances the overall performance of the classification.

V. CONCLUSION

In this article, we have introduced a novel automatic feature extraction algorithm for hyperspectral imagery, leveraging MADRL to reduce the reliance on manually crafted SIs. The model employs a heuristic "generation-selection" strategy that emulates the decision-making process of experts, with distinct DRL formulated for the generation and selection steps. Moreover, to expedite exploration in the high-dimensional

action space characteristic of hyperspectral data, a MADRL algorithm is introduced. Comparative experiments with several feature selection and AutoFE methods demonstrated that the proposed approach outperforms the others in both extraction efficiency and overall performance.

A primary limitation of the proposed HI-MAFE method lies in its modular design. Future research will focus on developing an end-to-end network architecture to further enhance the feature extraction process. In addition, the proposed method has only been validated on airborne hyperspectral datasets. Subsequent studies will consider evaluating the model's applicability and robustness across different platforms and sensor types. Moreover, knowledge-based feature extraction methods can effectively compensate for the shortcomings of data-driven approaches. Therefore, a dual-driven feature learning strategy that integrates both knowledge and data may represent an important direction for future research.

REFERENCES

[1] D. Chutia, D. K. Bhattacharyya, K. K. Sarma, R. Kalita, and S. Sudhakar, "Hyperspectral remote sensing classifications: A perspective survey," *Trans. GIS*, vol. 20, no. 4, pp. 463–490, Aug. 2016, doi: 10.1111/tgis.12164.

[2] R. Vaddi, B. L. N. P. Kumar, P. Manoharan, L. Agilandeeswari, and V. Sangeetha, "Strategies for dimensionality reduction in hyperspectral remote sensing: A comprehensive overview," *Egyptian J. Remote Sens. Space Sci.*, vol. 27, no. 1, pp. 82–92, Mar. 2024, doi: 10.1016/j.ejrs.2024.01.005.

[3] S. Asadzadeh, X. Zhou, and S. Chabrillat, "Assessment of the spaceborne EnMAP hyperspectral data for alteration mineral mapping: A case study of the Reko Diq porphyry Cu–Au deposit, Pakistan," *Remote Sens. Environ.*, vol. 314, Dec. 2024, Art. no. 114389, doi: 10.1016/j.rse.2024.114389.

[4] B. G. Ram, P. Oduor, C. Igathinathane, K. Howatt, and X. Sun, "A systematic review of hyperspectral imaging in precision agriculture: Analysis of its current state and future prospects," *Comput. Electron. Agricult.*, vol. 222, Jul. 2024, Art. no. 109037, doi: 10.1016/j.compag.2024.109037.

[5] Z. Wang, K. Tan, X. Wang, and W. Zhang, "Leveraging multi-class background description and token dictionary representation for hyperspectral anomaly detection," *Pattern Recognit.*, vol. 169, Jan. 2026, Art. no. 111945, doi: 10.1016/j.patcog.2025.111945.

[6] P. K. Srivastava, S. Srivastava, P. Singh, A. Gupta, and V. Dugesar, "Soil chemical properties estimation using hyperspectral remote sensing: A review," in *Earth Observation for Monitoring and Modeling Land Use*. New York, NY, USA: Elsevier, 2024, pp. 25–43.

[7] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968, doi: 10.1109/TIT.1968.1054102.

[8] S. Hongjun, "Dimensionality reduction for hyperspectral remote sensing: Advances, challenges and prospects," *Nat. Remote Sens. Bull.*, vol. 26, no. 8, pp. 1504–1529, Aug. 2022, doi: 10.11834/jrs.20210354.

[9] B. Zhang, "Advancement of hyperspectral image processing and information extraction," *Nat. Remote Sens. Bull.*, vol. 20, no. 5, pp. 1062–1090, Sep. 2016, doi: 10.11834/jrs.20166179.

[10] L. Mou, S. Saha, Y. Hua, F. Bovolo, L. Bruzzone, and X. X. Zhu, "Deep reinforcement learning for band selection in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5504414, doi: 10.1109/TGRS.2021.3067096.

[11] J. Feng et al., "Deep reinforcement learning for semisupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5501719, doi: 10.1109/TGRS.2021.3049372.

[12] L. Zhao et al., "Hyperspectral feature selection for SOM prediction using deep reinforcement learning and multiple subset evaluation strategies," *Remote Sens.*, vol. 15, no. 1, p. 127, Dec. 2022, doi: 10.3390/rs15010127.

[13] J. Feng et al., "Multi-agent deep reinforcement learning for hyperspectral band selection with hybrid teacher guide," *Knowl.-Based Syst.*, vol. 299, Sep. 2024, Art. no. 112044, doi: 10.1016/j.knosys.2024.112044.

[14] Y. Guo, Q. Wang, B. Hu, X. Qian, and H. Ye, "Two-stage unsupervised hyperspectral band selection based on deep reinforcement learning," *Remote Sens.*, vol. 17, no. 4, p. 586, Feb. 2025, doi: 10.3390/rs17040586.

[15] J. Feng, G. Bai, D. Li, X. Zhang, R. Shang, and L. Jiao, "MR-selection: A meta-reinforcement learning approach for zero-shot hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5500320, doi: 10.1109/TGRS.2022.3231870.

[16] C.-I. Chang, Q. Du, T.-L. Sun, and M. L. G. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2631–2641, Nov. 1999, doi: 10.1109/36.803411.

[17] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009, doi: 10.1109/TGRS.2008.2005729.

[18] L. Qian, X. Wang, M. Hong, H. Wang, and Y. Zhang, "An efficient and adaptive reconstructive homogeneous block-based local tensor robust PCA for feature extraction of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 4392–4407, 2024.

[19] S. Wang, N. A. A. Braham, and X. X. Zhu, "Weak–strong graph contrastive learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5512217, doi: 10.1109/TGRS.2025.3562261.

[20] L. Tu, X. Huang, J. Li, J. Yang, and J. Gong, "A multi-task learning method for extraction of newly constructed areas based on bi-temporal hyperspectral images," *ISPRS J. Photogramm. Remote Sens.*, vol. 208, pp. 308–323, Feb. 2024, doi: 10.1016/j.isprsjprs.2024.01.016.

[21] J. Feng et al., "S4DL: Shift-sensitive spatial–spectral disentangling learning for hyperspectral image unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 9, pp. 16894–16908, Sep. 2025, doi: 10.1109/TNNLS.2025.3556386.

[22] N. Dahiya, S. Singh, and S. Gupta, "A review on deep learning classifier for hyperspectral imaging," *Int. J. Image Graph.*, vol. 23, no. 4, Jul. 2023, Art. no. 2350036, doi: 10.1142/s0219467823500365.

[23] H. Wang, X. Liu, Z. Qiao, and H. Tao, "Inducing causal meta-knowledge from virtual domain: Causal meta-generalization for hyperspectral domain generalization," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5538416, doi: 10.1109/TGRS.2024.3494796.

[24] C. T. Nguyen, A. Chidthaisong, P. K. Diem, and L.-Z. Huo, "A modified bare soil index to identify bare land features during agricultural fallow-period in Southeast Asia using Landsat 8," *Land*, vol. 10, no. 3, p. 231, Feb. 2021, doi: 10.3390/land10030231.

[25] H. Xu et al., "Immediate assessment of forest fire using a novel vegetation index and machine learning based on multi-platform, high temporal resolution remote sensing images," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 134, Nov. 2024, Art. no. 104210, doi: 10.1016/j.jag.2024.104210.

[26] J. Yu, X. Li, X. Guan, and H. Shen, "A remote sensing assessment index for urban ecological livability and its application," *Geo-Spatial Inf. Sci.*, vol. 27, no. 2, pp. 289–310, Mar. 2024, doi: 10.1080/10095020.2022.2072775.

[27] Y. Xiong et al., "Assessment of spatial–temporal changes of ecological environment quality based on RSEI and GEE: A case study in Erhai Lake Basin, Yunnan province, China," *Ecolog. Indicators*, vol. 125, Jun. 2021, Art. no. 107518, doi: 10.1016/j.ecolind.2021.107518.

[28] D. Montero, C. Aybar, M. D. Mahecha, F. Martinuzzi, M. Söchting, and S. Wieneke, "A standardized catalogue of spectral indices to advance the use of remote sensing in Earth system research," *Scientific Data*, vol. 10, no. 1, Apr. 2023, Art. no. 197, doi: 10.1038/s41597-023-02096-0.

[29] F. J. Kriegler, "Preprocessing transformations and their effects on multispectral recognition," in *Proc. 6th Int. Symp. Remote Sesning Environ.*, 1969, pp. 97–131.

[30] J. M. Chen and T. A. Black, "Defining leaf area index for non-flat leaves," *Plant, Cell Environ.*, vol. 15, no. 4, pp. 421–429, May 1992, doi: 10.1111/j.1365-3040.1992.tb00992.x.

[31] H. Xu, Y. Wang, H. Guan, T. Shi, and X. Hu, "Detecting ecological changes with a remote sensing based ecological index (RSEI)produced time series and change vector analysis," *SinRemote Sens.*, vol. 11, no. 20, 2024, Art. no. 2345, doi: 10.3390/rs11202345.

[32] K. Yan et al., "A global systematic review of the remote sensing vegetation indices," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 139, May 2025, Art. no. 104560, doi: 10.1016/j.jag.2025.104560.

[33] L. Yan, J. Yang, and J. Wang, "Domain knowledge-guided self-supervised change detection for remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4167–4179, 2023, doi: 10.1109/JSTARS.2023.3270498.

[34] H. Tian, P. Wang, K. Tansey, and S. Zhang, "A knowledge-guided deep learning framework with remotely sensed variables and meteorological variables for improving wheat yield estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 4415813, doi: 10.1109/TGRS.2025.3600418.

[35] X. Chen et al., "Neural feature search: A neural architecture for automated feature engineering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Beijing, China, Nov. 2019, pp. 71–80.

[36] L. Li, H. Wang, and L. Zha, "Learning a data-driven policy network for pre-training automated feature engineering," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–19.

[37] T. Zhang et al., "OpenFE: Automated feature generation with expert-level performance," in *Proc. Mach. Learn. Res.*, 2022, pp. 41880–41901.

[38] Z. Liu, D. Zhang, H. Liu, Z. Dong, W. Jia, and J. Tan, "Visible-hidden hybrid automatic feature engineering via multi-agent reinforcement learning," *Knowl.-Based Syst.*, vol. 299, Sep. 2024, Art. no. 111941, doi: 10.1016/j.knosys.2024.111941.

[39] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

[40] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Appl. Intell.*, vol. 53, no. 11, pp. 13677–13722, Jun. 2023, doi: 10.1007/s10489-022-04105-y.

[41] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[43] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, Mar. 2005, doi: 10.1109/TGRS.2004.841417.

[44] R. Ji et al., "PatchOut: A novel patch-free approach based on a transformer-CNN hybrid framework for fine-grained land-cover classification on large-scale airborne hyperspectral images," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 138, Apr. 2025, Art. no. 104457, doi: 10.1016/j.jag.2025.104457.

[45] C. Niu, K. Tan, X. Wang, P. Du, and C. Pan, "A semi-analytical approach for estimating inland water inherent optical properties and chlorophyll a using airborne hyperspectral imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 128, Apr. 2024, Art. no. 103774, doi: 10.1016/j.jag.2024.103774.

[46] F. Horn, R. C. Pack, and M. L. Rieger, "The autofeat Python library for automated feature engineering and selection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2020, pp. 111–120.

[47] G. Zhu, Z. Xu, X. Guo, C. Yuan, and Y. Huang, "DIFER: Differentiable automated feature engineering," in *Proc. Mach. Learn. Res.*, 2022, pp. 1–17.

[48] H. Su, H. Yang, Q. Du, and Y. Sheng, "Semisupervised band clustering for dimensionality reduction of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 6, pp. 1135–1139, Nov. 2011, doi: 10.1109/LGRS.2011.2158185.

[49] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 8, pp. 1026–1041, Aug. 2007, doi: 10.1109/TKDE.2007.1048.