Contrastive Learning-Driven Hyperspectral Unmixing with Convolutional Autoencoders

Laura Manuela Castañeda Medina

February 14, 2025

Abstract

This research project explores the unmixing of hyperspectral images by incorporating contrastive learning techniques. The primary objective is to accurately estimate both abundances and endmembers within a single hyperspectral image using a novel self-supervised framework. A custom autoencoder architecture is designed, where the encoder generates abundance maps constrained by a sum-to-one layer, and the decoder reconstructs the original hyperspectral data. The training process integrates unsupervised contrastive learning, leveraging transformations such as cropping, flipping, and Gaussian blurring to create positive and negative pairs for representation learning.

Extensive experiments have been conducted to evaluate the impact of data augmentations, hyperparameter configurations, and contrastive loss functions. A supervised NT-Xent loss adapted to hyperspectral unmixing is employed, incorporating kernel-based weighting schemes that leverage endmember information. Performance is assessed using the spectral angle distance (SAD), ensuring that predicted endmembers align closely with the ground truth. Initial results indicate that specific augmentations enhance unmixing performance, and further optimization of multi-label contrastive loss functions is explored. This work advances hyperspectral unmixing methodologies and highlights the potential of contrastive self-supervised learning in remote sensing applications.

1 Introduction

1.1 Background

Hyperspectral imaging (HSI) captures rich spectral information across a wide range of wavelengths, allowing for precise material identification and analysis in diverse applications, including remote sensing, environmental monitoring, and mineral exploration. Despite its advantages, the high dimensionality and complexity of hyperspectral data present significant challenges for effective analysis. One of the core problems in HSI is hyperspectral unmixing, which aims to decompose a pixel's spectral signature into a combination of pure spectral components (endmembers) and their respective fractional abundances. This process is crucial for deriving meaningful insights from hyperspectral data.

Traditional unmixing techniques are predominantly based on mathematical models such as the linear mixing model (LMM), which assumes that each pixel is a linear combination of endmember spectra. While widely used due to its simplicity and interpretability, LMM often fails to capture the complex nonlinear interactions found in real-world scenarios [Bio+12]. To address these limitations, researchers have explored nonlinear unmixing approaches and deep learning techniques, which have demonstrated promising results in endmember extraction and abundance estimation [PSU22; Zha+22]. However, deep learning-based methods typically require large labeled datasets, which are often unavailable in hyperspectral imaging.

Recent advances in self-supervised learning (SSL), particularly contrastive learning frameworks, offer a compelling alternative by enabling representation learning without extensive labeled data [Che+20]. These techniques leverage augmentations to construct positive and negative pairs, encouraging models to learn invariant feature representations. Applying contrastive learning to hyperspectral unmixing represents a novel and promising approach, as it combines the strengths of deep learning and self-supervised methodologies to overcome the challenges of hyperspectral data analysis. This study investigates how contrastive learning can enhance hyperspectral unmixing, leveraging a custom autoencoder architecture combined with self-supervised contrastive losses to improve the accuracy of both endmember extraction and abundance estimation.

2 Method

2.1 Mathematical Background

2.1.1 Autoencoders for Hyperspectral Unmixing

Autoencoders are a class of neural networks designed for unsupervised representation learning by encoding input data into a lower-dimensional latent space and subsequently reconstructing it. Formally [Mas+11], an autoencoder consists of two primary components:

- **Encoder:** A function f_θ parameterized by neural network weights θ, which maps the input data $X \in \mathbb{R}^{m\times n\times L}$ to a lower-dimensional representation $Z \in \mathbb{R}^{m\times n\times d}$:

$$Z = f_\theta(X) \tag{1}$$

  where L is the number of spectral bands, and d is the dimensionality of the latent space.

- **Decoder:** A function g_ϕ parameterized by weights ϕ that reconstructs the original input from Z:

$$\hat{X} = g_\phi(Z) \tag{2}$$

In hyperspectral unmixing, the encoder extracts meaningful spectral-spatial features, while the decoder reconstructs the hyperspectral image using estimated abundance maps and endmembers. The constraint-based convolutional autoencoder used in this study follows the approach of [PUS21], ensuring:

- **Non-negativity:** Ensuring physically interpretable abundance maps by constraining outputs to $[0, 1]$.
- **Sum-to-One Constraint:** Enforcing $\sum_{i=1}^{N} a_i = 1$, where a_i are abundance fractions for each pixel.

Given an input hyperspectral image X, the encoder outputs abundance maps A such that:

$$A = f_\theta(X), \quad \text{s.t. } A \geq 0, \quad \sum_i A_i = 1 \tag{3}$$

The decoder reconstructs the hyperspectral image using the estimated endmembers E:

$$\hat{X} = g_\phi(A, E) = A \cdot E \tag{4}$$

where E is a set of spectral signatures learned by the model.

2.1.2 Contrastive Learning

Contrastive learning is a self-supervised learning approach that enhances feature representation by ensuring that semantically similar samples (positive pairs) are closer in representation space, while dissimilar samples (negative pairs) are pushed apart. Given an input sample x_i, two augmented views $v_i^{(1)}$ and $v_i^{(2)}$ are generated via transformations:

$$v_i^{(1)} = t_1(x_i), \quad v_i^{(2)} = t_2(x_i), \quad t_1, t_2 \sim \mathcal{T} \tag{5}$$

where $\mathcal{T}$ is a set of predefined transformations. The goal of contrastive learning is to maximize the similarity between $v_i^{(1)}$ and $v_i^{(2)}$ while minimizing similarity to all other samples in the batch.

The standard contrastive loss, known as the InfoNCE loss [OLV18], is defined as:

$$\mathcal{L}_{NCE} = -\log \frac{e^{f_\theta(v_i^{(1)}, v_i^{(2)})/\tau}}{\sum_{j=1}^{n} e^{f_\theta(v_i^{(1)}, v_j^{(2)})/\tau}} \tag{6}$$

where:

- $f_\theta(v_i^{(1)}, v_i^{(2)}) := \frac{1}{\tau} f_\theta(v_1)^T f_\theta(v_2)$
- $\tau > 0$ is a temperature parameter that controls the sharpness of the similarity distribution.

To further refine the learned representations, an extension inspired by Vicinal Risk Minimization (VRM) [Cha+00] incorporates metadata-aware adjustments. If auxiliary metadata y_i is available (e.g., abundances), a weighted similarity measure can be introduced:

$$p_{emp}^{vic}(v_1, v_2|y) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{w_\sigma(y_i, y_j)}{\sum_{k=1}^{n} w_\sigma(y_j, y_k)} \delta(v_1 - v_i^{(1)}) \delta(v_2 - v_j^{(2)}) \tag{7}$$

where $w_\sigma(y_i, y_j)$ is an RBF kernel that assigns higher weights to pairs with similar metadata values.

The resulting metadata-aware InfoNCE loss is given by:

$$\mathcal{L}_{NCE}^{y} = -\sum_{k=1}^{n} \frac{w_\sigma(y_k, y_i)}{\sum_{j=1}^{n} w_\sigma(y_j, y_i)} \log \frac{e^{f_\theta(v_i^{(1)}, v_i^{(2)})/\tau}}{\sum_{j=1}^{n} e^{f_\theta(v_i^{(1)}, v_j^{(2)})/\tau}} \tag{8}$$

This formulation refines contrastive learning by incorporating additional structural information from auxiliary variables, improving its adaptability to structured datasets such as hyperspectral images.

By integrating contrastive learning into hyperspectral unmixing, the encoder learns more robust and invariant representations, ultimately improving abundance and endmember estimation.

2.2 Dataset

This study evaluates the proposed hyperspectral unmixing methodology using six widely studied benchmark datasets: Cuprite, Jasper Ridge, Samson, Urban4, Urban5, and Urban6. These datasets encompass diverse environmental and material compositions, providing a comprehensive testbed for hyperspectral unmixing techniques.

Dataset	**Bands**	**Endmembers**	**Description**
Cuprite	188	12	Well-defined endmembers with minimal spectral variability.
Jasper Ridge	198	4	Natural environment with complex mixtures of vegetation and soil.
Samson	156	3	Small-scale dataset with highly separable endmembers, benchmark for evaluating unmixing performance.
Urban4	162	4	Urban dataset with four endmembers, different man-made materials (asphalt and concrete).
Urban5	162	5	Similar to Urban4 but includes five endmembers.
Urban6	162	6	The most complex urban dataset with six endmembers.

Table 1: Summary of Hyperspectral Datasets

Each dataset was preprocessed to normalize spectral values and ensure compatibility with the model architecture. Patch-based sampling was employed, with patch sizes carefully selected to balance computational efficiency and the preservation of spatial context. This approach enables the model to leverage both spectral and local spatial information during training and evaluation.

2.3 Model Architecture

The proposed model is based on the Convolutional Neural Network Autoencoder for Unmixing (CNNAEU) as described in [PSU22]. This architecture is designed to estimate both abundance maps and endmembers from hyperspectral images by leveraging convolutional layers for spectral-spatial feature extraction.

The model consists of two main components:

- **Encoder:** A series of convolutional layers designed to extract meaningful spectral-spatial features from hyperspectral data. The encoder outputs abundance maps, which are constrained by a Sum-to-One layer, ensuring that the fractional abundances satisfy the physical constraint of summing to one.
- **Decoder:** A reconstruction module that generates the hyperspectral image based on the estimated abundance maps and learned endmembers. Non-negativity constraints are enforced in the decoder to ensure physically interpretable results.

To enhance representation learning, the model integrates contrastive learning, inspired by [Duf+21], using data augmentations such as cropping, flipping, and Gaussian blurring. This approach enables the model to leverage self-supervised learning by creating positive and negative pairs, improving robustness in endmember estimation.

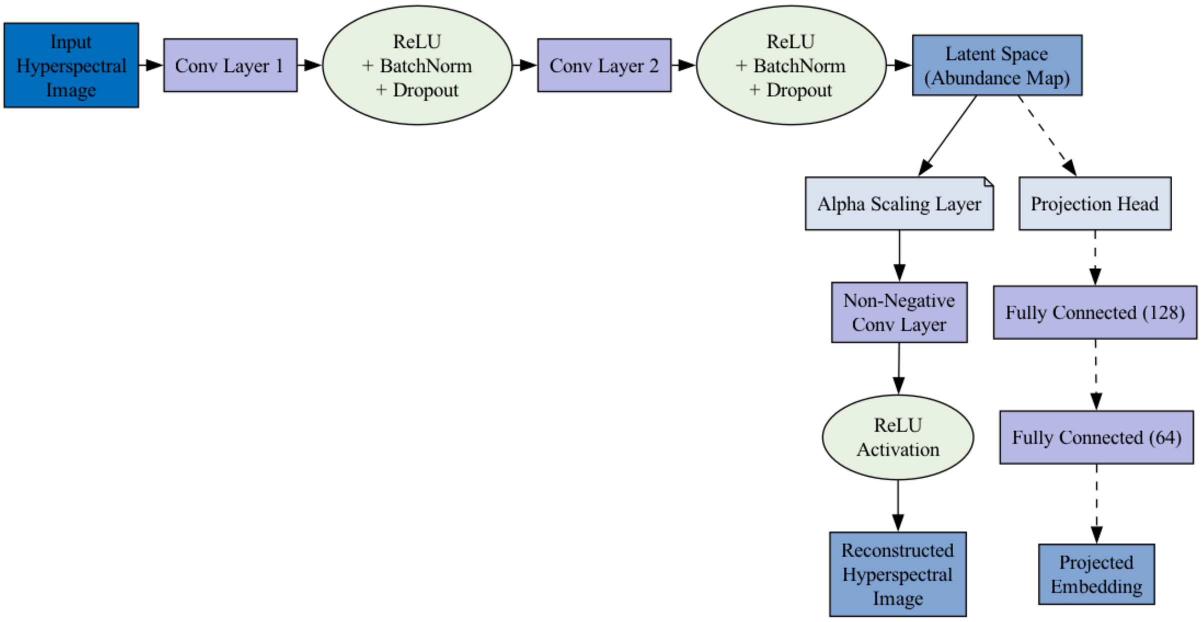

Figure 1: Convolutional Autoencoder Architecture

The projection head plays a crucial role in contrastive learning by mapping the latent space representation (abundance map) into a feature space that enhances spectral feature separability. In this work, various projection head architectures were explored to assess their impact on contrastive learning performance.

The contrastive learning framework requires an embedding space where positive pairs (augmentations of the same sample) are pulled together, and negative pairs (different samples) are pushed apart. Directly applying contrastive loss in the latent abundance space can lead to suboptimal representations, as this space is constrained by physical properties such as sum-to-one and non-negativity. The projection head allows for a transformation that is better suited for contrastive learning.

Several projection head designs were tested, each with distinct properties:

Low-Dimensional Projection Head This approach applied a simple transformation from the latent space (num_endmembers) to a fixed-dimensional space of 64:

$$Z' = \text{ReLU}(\mathbf{W}_1 Z + b_1) \tag{9}$$

where $Z \in \mathbb{R}^{\text{num\_endmembers}}$ is the input latent representation, $\mathbf{W}_1 \in \mathbb{R}^{64\times\text{num\_endmembers}}$ is the projection matrix, and $Z' \in \mathbb{R}^{64}$ is the output.

Skip Connection Projection Head Inspired by residual learning, this projection head introduced a skip connection that directly propagated the original latent features while transforming them through additional layers. The architecture followed:

$$Z' = Z + \text{ReLU}(\mathbf{W}_1 Z + b_1) \tag{10}$$

where the input latent space is summed with a transformed version of itself, allowing the projection head to preserve key spectral information.

Bottleneck Projection Head A deeper variant of the projection head was tested using an intermediate compressed representation. The architecture consisted of:

$$Z' = \mathrm{ReLU}(\mathbf{W}_1 Z + b_1) \tag{11}$$

$$Z'' = \mathrm{ReLU}(\mathbf{W}_2 Z' + b_2) \tag{12}$$

where the intermediate space was set to 256 before reducing to 64, simulating a bottleneck structure.

Deeper Projection Head A fully connected deep projection head was also explored, mapping the latent space to a higher-dimensional space before reducing it to 64:

$$Z' = \mathrm{ReLU}(\mathbf{W}_1 Z + b_1) \tag{13}$$

$$Z'' = \mathrm{ReLU}(\mathbf{W}_2 Z' + b_2) \tag{14}$$

$$Z''' = \mathrm{ReLU}(\mathbf{W}_3 Z'' + b_3) \tag{15}$$

where each transformation gradually refined the features before the final projection.

Baseline Projection Head (Selected Model) The best-performing projection head was a simple fully connected layer that expanded the latent space from num_endmembers to 128 before projecting it to 64:

$$Z' = \mathrm{ReLU}(\mathbf{W}_1 Z + b_1) \tag{16}$$

$$Z'' = \mathrm{ReLU}(\mathbf{W}_2 Z' + b_2) \tag{17}$$

2.3.1 Loss Function

The training framework is optimized using a hybrid loss function that balances two critical components:

- **Contrastive Loss ($\mathcal{L}_{NCE}$):** Encourages representations of augmented views of the same sample to be similar while enforcing separation from different samples. This improves the discriminability of spectral-spatial features.

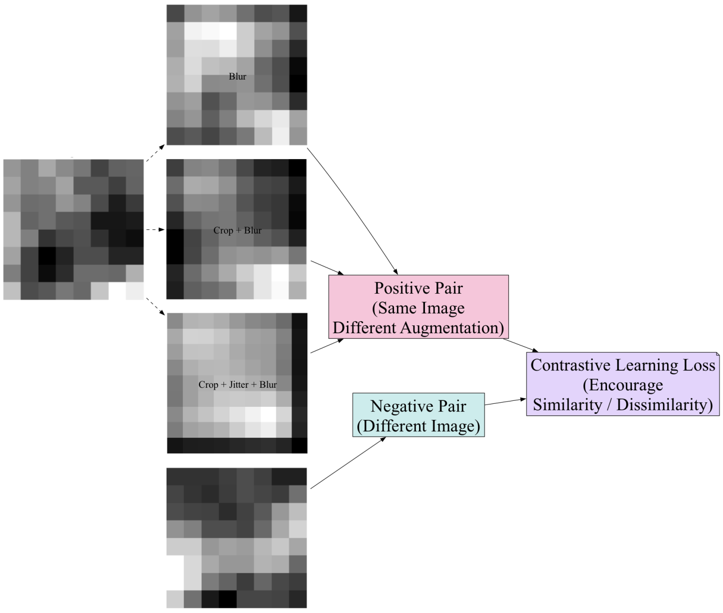

Figure 2: Contrastive Learning Framework

- **Reconstruction Loss ($\mathcal{L}_{SAD}$):** Enforced via Spectral Angle Distance (SAD), ensuring that the reconstructed hyperspectral data maintains spectral fidelity by minimizing the angular difference between predicted and ground truth spectra. The SAD loss is defined as:

$$\mathcal{L}_{SAD} = \frac{1}{N} \sum_{i=1}^{N} \arccos \left(\frac{\mathbf{x}_i \cdot \hat{\mathbf{x}}_i}{\|\mathbf{x}_i\| \|\hat{\mathbf{x}}_i\|} \right), \tag{18}$$

where:

  - $\mathbf{x}_i$ is the ground truth spectral vector for pixel i.
  - $\hat{\mathbf{x}}_i$ is the reconstructed spectral vector.
  - $\|\mathbf{x}_i\|$ and $\|\hat{\mathbf{x}}_i\|$ are their respective Euclidean norms.
  - N is the total number of pixels.

The final objective function combines these terms with a weighting factor λ:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{NCE} + \lambda \mathcal{L}_{SAD} \tag{19}$$

This formulation enables the model to leverage both contrastive learning and spectral reconstruction, enhancing robustness, generalization, and interpretability in hyperspectral unmixing.

2.4 Training Procedure

The training process was conducted in three sequential stages to evaluate and refine the performance of the proposed model:

1. **Baseline Training:** The initial phase involved training the original CNNAEU model on each dataset without any augmentations. Each hyperspectral image was used as a single input for training, and the results established a baseline for later comparisons.

2. **Augmented Training:** To increase the diversity of training samples and improve robustness, data augmentations were applied to generate transformed versions per dataset. These augmentations were essential for later contrastive learning, as they provided positive pairs of the same sample under different transformations. The transformations were categorized as follows:

Category	**Transformations**	
Single	Blur	$\sigma = [0.1, 1.0]$, $\sigma = [0.1, 2.0]$
	Crop	50%, 75%, and 95%
	Flip	Vertical, Horizontal, Random
	Jittering	Noise-based perturbation
Combinations	Crop + Blur	Crop + Flip
	Crop + Flip + Blur	Crop + Jitter
	Crop + Jitter + Blur	Crop + Jitter + Flip
	Jitter + Flip	Jitter + Flip + Blur

Table 2: Summary of Data Augmentation Transformations

The best-performing augmentations from this stage were selected for the next phase of training.

3. **Integrated Training with Contrastive Learning:** In this stage, contrastive learning was introduced to improve feature representations. The autoencoder was trained using the contrastive loss function ($\mathcal{L}_{NCE}$) combined with spectral reconstruction loss ($\mathcal{L}_{SAD}$) to balance representation learning and reconstruction accuracy:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{NCE} + \lambda\mathcal{L}_{SAD}$$

   During this stage, positive pairs were formed by selecting different augmented versions of the same hyperspectral patch, while negative pairs were drawn from different patches. A hyperparameter search was conducted to optimize the model. The key parameters included:

   - **Temperature (τ):** Controls the contrastive loss sensitivity by adjusting similarity distribution sharpness.
   - **Lambda (λ):** Defines the trade-off between contrastive loss and reconstruction loss.
   - **Batch Size:** Determines the number of samples processed together, affecting computational efficiency and gradient estimation.
   - **Patch Size:** Determines extracted spatial regions, influencing spatial context retention.
   - **Epochs:** Defines the number of complete passes through the training dataset, affecting convergence and generalization.

   This phase aimed to maximize performance, balancing contrastive learning and reconstruction quality for accurate hyperspectral unmixing. Additionally, if metadata-aware contrastive learning was employed, auxiliary variables were incorporated to refine positive and negative sample selection.

The results of these training phases, including quantitative performance comparisons between the baseline model, augmented training, and contrastive learning, will be detailed in the following sections using tables and visualizations.

3 Experiments

3.1 Experiment Design

A series of controlled experiments were conducted to evaluate the performance of the autoencoder under different augmentation strategies and training conditions. The primary goal was to determine whether specific augmentations improve the accuracy of hyperspectral unmixing, as measured by the Spectral Angle Distance (SAD) and its standard deviation (std).

The experimental setup provides a systematic approach to evaluating the role of augmentations, training variability, and contrastive learning in hyperspectral unmixing.

3.1.1 Baseline Training

The autoencoder was first trained on the original dataset without any augmentations. This baseline experiment consisted of 25 independent runs, computing the average Spectral Angle Distance (SAD) and its standard deviation to establish a reference performance level.

The original results can be found in the paper [PSU22]. The table below presents the SAD results obtained from our experiments, which closely match those reported in the paper.

Dataset	Average SAD	Standard Deviation (STD)
Samson	0.0654	0.0058
Urban4	0.0428	0.0057
Urban5	0.0766	0.0079
Urban6	0.1311	0.0387
Cuprite	0.1229	0.0109
Jasper Ridge	0.1841	0.0634

Table 3: Baseline SAD results obtained using the autoencoder.

3.1.2 Augmentation-Based Training

Inspired by the methodology in [Zha+22], 90 augmented variations were generated for each dataset. Each augmentation set was used to train the autoencoder again, and the results were evaluated by comparing the average SAD and std against the baseline. This analysis helped identify which transformations contributed to improved unmixing performance.

To assess the impact of augmentation strategies and training variability, three different training configurations were tested:

Training Setup	Seed	Augmentations	Purpose
Unseeded Training, Dynamic Augmentations	No Seed	90 augmentations per run, different each iteration	Tests robustness across varying augmentation sets.
Unseeded Training, Fixed Augmentations	No Seed	Same 90 augmentations used across all runs	Isolates augmentation effect from variability.
Seeded Training, Fixed Augmentations	Fixed Seed	Same 90 augmentations used across all runs	Tests the effect of initialization stability.

Table 4: Comparison of Training Strategies

These configurations provided insights into the role of randomness in training and data augmentation, allowing us to determine whether augmentation variability contributes to performance inconsistencies.

3.1.3 Integration with Contrastive Learning

After identifying the best-performing augmentations, the next phase of the project involved integrating contrastive learning into the training framework. The objective was to investigate whether contrastive learning, when combined with effective augmentations, could further improve abundance estimations and enhance hyperspectral unmixing.

1. **Selection of Augmentations for Contrastive Learning:**
   - Augmentations were selected based on their ability to reduce SAD while maintaining consistency across different training setups (seeded vs. unseeded).
   - The best augmentations were then applied within a contrastive learning framework, where the model was trained to optimize both contrastive loss and reconstruction loss simultaneously.

2. **Contrastive Learning Framework:**
   - Positive and negative pairs were generated using augmentations of the same patch, enforcing similarity constraints between different augmented views.
   - The contrastive loss was applied in the latent space to separate hyperspectral features while preserving meaningful reconstructions.
   - The reconstruction loss (SAD) was retained to ensure that the extracted endmembers remained physically interpretable.

3. **Contrastive Loss Functions:**
   - **NT-Xent Loss (Normalized Temperature-scaled Cross-Entropy Loss):** The NT-Xent loss [Che+20] is commonly used in SimCLR-based contrastive learning. It maximizes similarity between positive pairs (augmented views of the same patch) and minimizes similarity with negative pairs (other samples in the batch).
   - **Generalized Contrastive Loss with Spectral Labeling:** Unlike NT-Xent, which assigns hard positive and negative pairs, the Generalized Contrastive Loss introduces a multi-kernel similarity weighting to handle hyperspectral data more effectively. Instead of treating all augmented patches as strictly positive or negative, similarity is determined based on the spectral composition of the patches.

     Unlike standard contrastive loss where labels are either positive (1) or negative (0), in the generalized approach, weights w_{ij} are assigned dynamically based on the spectral similarity of the samples. The labels are computed using multi-kernel similarity functions, which measure spectral consistency between samples:

     $$w_{ij} = \prod_{e} \mathcal{K}(y_{i,e}, y_{j,e}), \tag{20}$$

     where:
     - $y_{i,e}$ represents the abundance value of endmember e for sample i.
     - $\mathcal{K}(\cdot)$ is a kernel function applied to each endmember to compute pairwise similarity, in this case:
       * **RBF (Gaussian) kernel**: $K(y_1, y_2) = \exp\left(-\frac{\|y_1-y_2\|^2}{2\sigma^2}\right)$
     - The final w_{ij} is normalized across all pairs to ensure a valid probability distribution.

4. **Projection Head**

   After identifying the best-performing hyperparameters for contrastive learning, an additional experiment was conducted to evaluate the impact of different projection head architectures. Due to time constraints, this evaluation was limited in scope and focused on assessing whether modifications to the projection head could enhance feature separability without requiring a full hyperparameter search.

   Each projection head variant was tested using the previously optimized settings for temperature, reconstruction weight, batch size, and patch size. The training process remained consistent across models, with contrastive and reconstruction losses monitored to compare performance. The objective was to determine if a more expressive transformation of the latent space could improve hyperspectral feature learning while maintaining reconstruction accuracy.

This stage aimed to explore whether contrastive learning, guided by the best augmentation strategies, could enhance spectral feature separation and improve the unmixing process. The following sections present quantitative results, comparing baseline, augmented, and contrastive learning-based training approaches.

4 Results

4.1 Augmentation-Based Results

The following tables present the Spectral Angle Distance (SAD) and standard deviation (STD) values for the datasets across selected augmentation strategies. The three augmentations reported for each dataset are those that consistently outperformed the baseline model or, at the very least, yielded the best results among all tested transformations.

Samson Dataset

Augmentation	**Average**	**STD**
Original Results	0.0654	0.0058
Blur (sigma = 0.1, 1.0)	0.0410	0.0193
Crop + Blur	0.0391	0.0046
Crop + Jitter + Blur	0.0472	0.0251

Urban4 Dataset

Augmentation	**Average**	**STD**
Original Results	0.0428	0.0057
Blur (sigma = 0.1, 1.0)	0.0425	0.0033
Crop + Blur	0.0408	0.0043
Crop + Jitter + Blur	0.0393	0.0045

Urban5 Dataset

Augmentation	**Average**	**STD**
Original Results	0.0766	0.0079
Blur (sigma = 0.1, 1.0)	0.0879	0.0054
Crop + Blur	0.0842	0.0070
Crop + Jitter + Blur	0.0841	0.0075

Urban6 Dataset

Augmentation	**Average**	**STD**
Original Results	0.1311	0.0387
Blur (sigma = 0.1, 1.0)	0.1413	0.0185
Crop + Blur	0.1217	0.0250
Crop + Jitter + Blur	0.1266	0.0294

Cuprite Dataset

Augmentation	**Average**	**STD**
Original Results	0.1229	0.0109
Blur (sigma = 0.1, 1.0)	0.1128	0.0094
Crop + Blur	0.1127	0.0099
Crop + Jitter + Blur	0.1124	0.0087

JasperRidge Dataset

Augmentation	**Average**	**STD**
Original Results	0.1841	0.0634
Blur (sigma = 0.1, 1.0)	0.1741	0.0217
Crop + Blur	0.1722	0.0195
Crop + Jitter + Blur	0.1714	0.0210

Table 5: Average and standard deviation (STD) of SAD results for datasets across different augmentation strategies.

For a complete overview of all tested augmentations and their respective results, a detailed table is at the following link: **Complete Results Spreadsheet**.

4.2 Endmember Reconstruction Results

To evaluate the performance of the autoencoder in extracting meaningful endmembers, we compare the reconstructed endmembers obtained with the three different hyperparameter configurations. The figures below present the spectral signatures of the true endmembers (solid lines) alongside the reconstructed ones (dashed lines) for the **Samson** and **Urban4** datasets.

4.2.1 Samson Dataset

Figures 3, 4, and 5 illustrate the reconstructed endmembers for the Samson dataset using the three selected hyperparameter configurations.

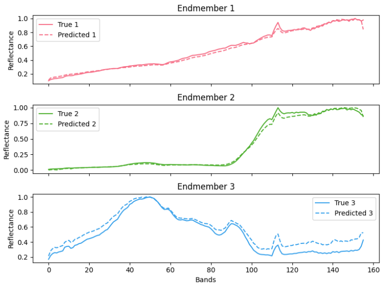

Figure 3: Samson - Best SAD (AVG)

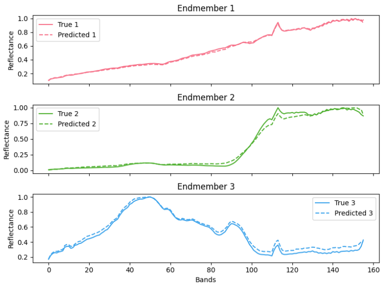

Figure 4: Samson - Best SAD (Samson)

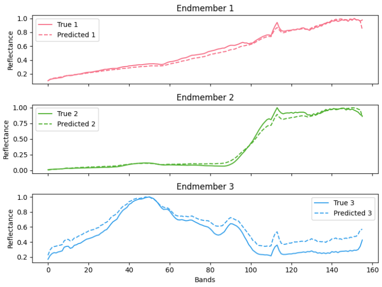

Figure 5: Samson - Best SAD (Urban4)

4.2.2 Urban4 Dataset

Figures 6, 7, and 8 show the corresponding results for the Urban4 dataset, demonstrating how different hyperparameter settings affect the reconstruction.

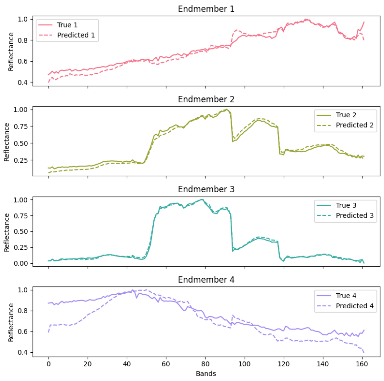

Figure 6: Urban4 - Best SAD (AVG)

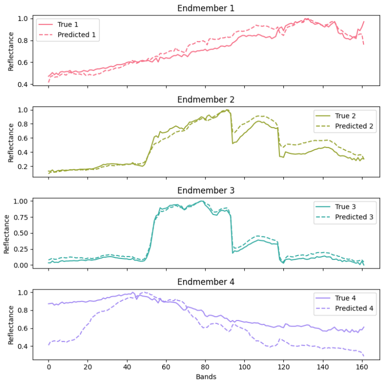

Figure 7: Urban4 - Best SAD (Samson)

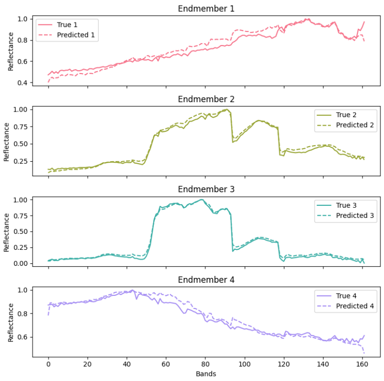

Figure 8: Urban4 - Best SAD (Urban4)

The results highlight the differences in spectral reconstruction depending on the hyperparameters used. The next section provides an in-depth analysis of these findings.

4.3 Hyperparameter Search

The resulting table presents a comparison of Spectral Angle Distance (SAD) values across different experimental setups for each dataset. For each dataset, the first row corresponds to the original baseline results, providing a reference for comparison. The next three rows summarize the results obtained from hyperparameter tuning: (1) the configuration that achieved the lowest average SAD across both datasets (Best SAD (Avg)), (2) the configuration that achieved the lowest SAD for that specific dataset (Best SAD (Local)), and (3) the configuration that achieved the lowest SAD in the other dataset (Best SAD (Other)).

This structure allows for a direct comparison of how different hyperparameter settings impact each dataset and highlights the trade-offs between optimizing for a single dataset versus achieving generalization across datasets.

Dataset	Experiment	Avg E1	Avg E2	Avg E3	Avg E4	Avg SAD
Samson	Original Results	0.0426	0.0416	0.1120	-	0.0654
Samson	Best SAD (Avg)	0.0356	0.0504	0.1143	-	0.0668
Samson	Best SAD (Local)	0.0195	0.0625	0.0513	-	0.0444
Samson	Best SAD (Other)	0.0444	0.0665	0.1403	-	0.0837
Urban4	Contrastive	0.0558	0.0429	0.0366	0.0360	0.0428
Urban4	Best SAD (Avg)	0.0485	0.0687	0.0376	0.1141	0.0672
Urban4	Best SAD (Local)	0.0576	0.0467	0.0584	0.0401	0.0507
Urban4	Best SAD (Other)	0.0636	0.1327	0.0931	0.2156	0.1263

Table 6: Comparison of Baseline and Test Results

4.4 Training Dynamics and Convergence

To analyze the stability and convergence of the training process, we monitor the evolution of both the contrastive loss and the reconstruction loss over 100 epochs. The contrastive loss measures the model's ability to separate different hyperspectral features in the latent space, while the reconstruction loss evaluates how well the model preserves spectral fidelity.

Figures 9 and 10 display the training losses for the Samson and Urban4 datasets, respectively, under different hyperparameter configurations.

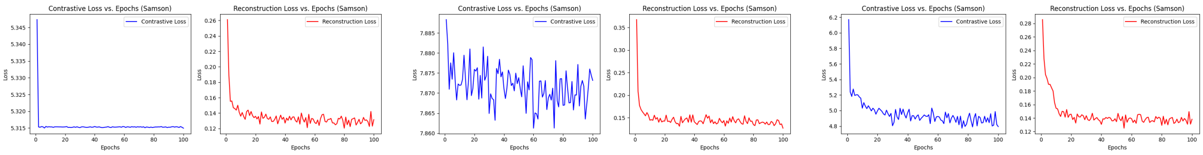

Figure 9: Contrastive and reconstruction loss evolution for the Samson dataset under different hyperparameter configurations.

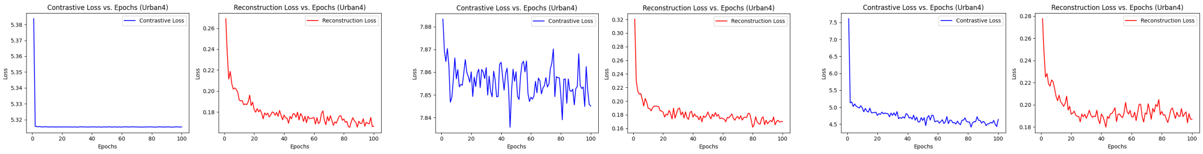

Figure 10: Contrastive and reconstruction loss evolution for the Urban4 dataset under different hyperparameter configurations.

These results highlight the necessity of carefully tuning hyperparameters to achieve a balance between spectral reconstruction accuracy and robust feature separation in the latent space.

5 Results Analysis

5.1 Impact of Augmentations on SAD Reduction

The results presented in Table 5 provide insights into the impact of various augmentation strategies on the performance of the autoencoder in hyperspectral unmixing.

First, it is evident that certain augmentations consistently led to lower SAD values compared to the original results across multiple datasets. Specifically, the **Blur (sigma = 0.1, 1.0)**, **Crop + Blur**, and **Crop + Jitter + Blur** augmentations showed the most consistent improvement across different datasets. This suggests that these augmentations help stabilize the learning process and enhance feature extraction for spectral unmixing.

Among these augmentations, **Blur (sigma = 0.1, 1.0)** often resulted in a noticeable improvement, particularly in datasets such as Urban5, Urban6, and JasperRidge. This indicates that introducing slight Gaussian blurring may reduce spectral noise while preserving relevant spatial structures, leading to more robust reconstructions.

The **Crop + Blur augmentation** also demonstrated consistent performance gains, possibly due to its combination of localized feature retention (from cropping) and spectral smoothing (from blurring). This effect was particularly pronounced in Urban4, Cuprite, and JasperRidge, where the SAD values were among the lowest observed.

Interestingly, the **Crop + Jitter + Blur augmentation**, despite being more complex, performed well across most datasets but exhibited slightly higher variance (STD values). This could indicate that while it enhances performance in some cases, it introduces additional variability in others.

5.2 Impact of Contrastive Learning

To assess the impact of integrating contrastive learning into the autoencoder framework, we analyze both the quality of the reconstructed endmembers (Figures 3, 4, 5, 6, 7 and 8) and the training loss evolution (Figure 9 and Figure 10).

5.2.1 Effect on Endmember Estimation

Figures 3, 4, 5, 6, 7 and 8 illustrate the comparison between the true and reconstructed endmembers for the Samson and Urban4 datasets, respectively, using different hyperparameter configurations. Several key observations can be made:

- Across all configurations, contrastive learning improves the alignment between the predicted and true endmember signatures. This is particularly evident in Endmember 3 of the Samson dataset, where the contrastive loss helps reduce discrepancies in the high-reflectance regions.
- The generalization capability of the model varies with the choice of hyperparameters. Some runs exhibit closer alignment with ground truth spectra, particularly when contrastive learning is combined with well-tuned temperature and batch size values.
- The Urban4 dataset presents a more complex challenge, with greater spectral variability among endmembers. Despite this, contrastive learning contributes to a reduction in deviations, particularly in Endmember 1 and Endmember 3.

These results demonstrate that integrating contrastive learning encourages the model to learn more structured feature representations in the latent space, leading to improved spectral reconstructions.

5.2.2 Training Stability and Convergence

Figures 9 and 10 provide insights into the behavior of the contrastive and reconstruction loss during training. The following trends are observed:

- The reconstruction loss shows a rapid decline during the initial epochs, stabilizing afterward. This indicates that the model quickly captures the core spectral information.

- The contrastive loss varies significantly across runs. Some configurations exhibit a smooth decline, while others display fluctuations, suggesting sensitivity to the choice of contrastive loss temperature and batch size.
- In both datasets, contrastive learning does not impede reconstruction loss minimization, confirming that the dual optimization strategy successfully balances spectral fidelity with latent space regularization.

These findings highlight the trade-offs involved in optimizing contrastive learning within hyperspectral unmixing. While it enhances spectral discrimination and robustness, careful tuning of hyperparameters is essential to ensure stability and convergence.

Overall, the results confirm that contrastive learning provides substantial benefits in endmember estimation and latent space organization, particularly when combined with effective augmentation strategies and well-calibrated training parameters.

5.3 Hyperparameter Influence on Results

The results presented in table 6 highlight the impact of different hyperparameter configurations on the autoencoder-based hyperspectral unmixing model. By comparing the baseline Spectral Angle Distance (SAD) values with those obtained from hyperparameter tuning, we can analyze how different parameter choices influence model performance.

Setup	**Temp.** (τ)	**Lambda** (λ)	**Patch**	**Batch**	**Sigma K**
Best SAD (Avg)	0.1	0.5	32	8	1.0
Best SAD (Samson)	10.0	0.5	16	30	0.1
Best SAD (Urban4)	0.01	0.9	32	8	0.1

Table 7: Best hyperparameter configurations for different datasets.

The Best SAD (Avg) configuration, which achieved the lowest SAD across datasets, used a moderate temperature ($\tau = 0.1$) and a balanced reconstruction weight ($\lambda = 0.5$). The patch size of 32 pixels provided a trade-off between spatial information and computational cost, while a small batch size of 8 ensured stable contrastive learning. The sigma kernel of 1.0 suggests that a relatively broad similarity measure was effective in distinguishing spectral features across different datasets.

In contrast, the Best SAD (Samson) configuration performed optimally with a very high temperature ($\tau = 10.0$), emphasizing more flexible similarity constraints in the contrastive learning framework. A smaller patch size (16 pixels) and larger batch size (30) indicate that, for Samson, smaller spatial contexts and more training samples per batch led to better feature learning. The reconstruction weight ($\lambda = 0.5$) suggests that contrastive loss played a more significant role in optimizing abundance representations. The sigma kernel of 0.1 indicates that sharper similarity constraints were necessary to capture spectral variations in this dataset.

For Best SAD (Urban4), the model favored a very low temperature ($\tau = 0.01$), which suggests that a stricter separation between similar and dissimilar samples was beneficial for this dataset. A higher reconstruction weight ($\lambda = 0.9$) indicates that preserving spectral fidelity was more important for Urban4, compared to the balance seen in the other configurations. The patch size of 32 pixels was the same as in the general configuration, while the batch size of 8 remained small, supporting stable gradient updates. A sigma kernel of 0.1 aligns with the need for sharper similarity constraints, similar to Samson's best-performing configuration.

These results demonstrate that hyperparameter selection significantly impacts hyperspectral unmixing performance. While a generalized configuration provides balanced performance across datasets, dataset-specific tuning yields optimal results by adjusting temperature, reconstruction weight, and similarity constraints. The findings suggest that a dynamic hyperparameter tuning approach could be beneficial, where parameters adapt based on dataset characteristics rather than being fixed across experiments.

5.4 Projection Head Performance Analysis

The evaluation of different projection head architectures revealed that the choice of mapping between the latent space and the contrastive loss significantly influenced the model's performance. While some designs introduced deeper transformations to the projected embeddings, they did not necessarily lead to improved hyperspectral unmixing.

The baseline projection head, which expanded the latent space from num_endmembers → 128 → 64, consistently performed best across datasets. This structure provided enough flexibility to refine the spectral representations while maintaining sufficient information for unmixing. Increasing the latent dimension to 128 before compressing to 64 helped the model capture complex spectral patterns without excessive transformations.

On the other hand, the skip connection projection head did not yield improvements over the baseline. While residual connections are known to help deeper networks retain important features, in this case, the direct propagation of raw features seemed redundant, as the transformation from num_endmembers → 64 was already shallow. This suggests that contrastive learning in hyperspectral unmixing may not benefit from shortcut pathways unless they preserve critical nonlinear relationships.

The bottleneck projection head, which first compressed the latent representation into 64 before expanding to 128, exhibited worse results. This compression likely caused a loss of important spectral information, making it harder for the model to learn meaningful relationships between spectral signatures. While bottleneck layers often work well in generic representation learning, hyperspectral unmixing relies on preserving fine spectral details, which this structure may have discarded.

A deeper projection head with multiple layers (e.g., num_endmembers → 256 → 128 → 64) showed improvements in some cases but lacked consistency across datasets. While a deeper network can capture more complex feature hierarchies, it also introduces a risk of overfitting, especially when datasets are relatively small or have distinct spectral characteristics. The fact that these models performed well on one dataset but failed on another suggests that they were learning dataset-specific patterns rather than generalizable spectral features.

Ultimately, these results highlight that increasing projection head complexity does not necessarily enhance contrastive learning in hyperspectral unmixing. Instead, an appropriate balance between expressiveness and regularization is crucial. Future research could explore adaptive projection heads that dynamically adjust their structure based on dataset characteristics, as well as alternative embedding spaces that better preserve spectral information.

6 Discussion

Despite the exploration of contrastive learning for hyperspectral unmixing, several limitations were observed in the experimental results:

- **Inconsistent Performance Across Datasets:** Contrastive learning did not consistently improve results across all datasets. While certain hyperparameter settings led to improvement in specific datasets, they failed to generalize well across multiple datasets.

- **Hyperparameter Sensitivity:** The performance of the model was highly dependent on the selection of hyperparameters such as temperature (τ), reconstruction weight (λ), and batch size. The best configuration for one dataset often did not translate well to others, limiting the approach's robustness.

- **Augmentation Selection Challenges:** Augmentations played a critical role in contrastive learning. While some augmentations, such as Gaussian blur and cropping, were beneficial, others (e.g., flipping) negatively impacted performance. This suggests that contrastive learning's effectiveness is tightly coupled with the choice of augmentation.

- **Computational Overhead:** The integration of contrastive learning increased computational requirements, particularly in terms of memory usage and processing time. The need for larger batch sizes and multiple augmentations per sample made training more expensive compared to traditional autoencoder-based unmixing.

- **Failure to Outperform Baseline:** While contrastive learning introduced structured representations in the latent space, it did not consistently yield lower Spectral Angle Distance (SAD) values compared to the baseline autoencoder. In some cases, hyperparameter tuning was able to reach comparable results, but no generalizable improvement was observed.

7 Conclusion

This study investigated the integration of contrastive learning within a convolutional autoencoder framework for hyperspectral unmixing. The results indicate that contrastive learning did not consistently improve performance across datasets. While it contributed to more structured feature representations in the latent space, these improvements did not always translate into lower spectral angle distance (SAD) values. This suggests that the added complexity of contrastive learning may not always be justified, especially when the baseline autoencoder already achieves competitive results.

A key limitation observed was the sensitivity of contrastive learning to hyperparameter choices. Certain hyperparameter configurations led to improvements in specific datasets, but these settings did not generalize well across different hyperspectral datasets. This highlights the need for more adaptive or dataset-specific tuning strategies to ensure that contrastive learning remains beneficial across various data distributions.

Another critical factor influencing the results was the choice of augmentations. While some augmentations, such as cropping combined with blurring, led to improvements, others had a negative impact on performance. This reinforces the importance of carefully selecting augmentations tailored to hyperspectral data characteristics, as inappropriate transformations may introduce distortions rather than enhance spectral separability.

Overall, the baseline autoencoder proved to be a strong and reliable model, often achieving robust performance without the additional constraints imposed by contrastive learning. Given that contrastive learning failed to provide consistent advantages, conventional hyperspectral unmixing techniques may still be preferable unless further refinements are made to the contrastive learning framework. Future work should focus on improving augmentation strategies, optimizing hyperparameter search, and exploring alternative self-supervised learning techniques to better leverage the strengths of contrastive learning for hyperspectral applications.

7.1 Future Work

Building on these findings, several avenues for future research can be explored:

- **Refinement of Contrastive Learning for Hyperspectral Data:** Developing a multi-label contrastive loss function that explicitly accounts for mixed spectral signatures in hyperspectral pixels could yield better results.
- **Exploring Alternative Self-Supervised Techniques:** Since contrastive learning did not provide a consistent improvement, alternative self-supervised methods, such as masked autoencoding or clustering-based pretraining, should be investigated.
- **Automated Hyperparameter Optimization:** The strong dependence on hyperparameter selection suggests the need for automated tuning strategies to identify robust configurations.
- **Testing on More Complex Real-World Data:** Future studies should explore the effectiveness of contrastive learning beyond benchmark datasets, testing on hyperspectral images from real-world remote sensing applications.
- **Reducing Computational Cost:** Given the high memory and processing requirements of contrastive learning, efficient architectures should be designed to reduce computational complexity without sacrificing performance.

By addressing these directions, the potential of self-supervised learning in hyperspectral unmixing can be better understood, potentially leading to more effective and scalable methods for spectral feature extraction.

References

[Cha+00] Olivier Chapelle et al. “Vicinal risk minimization”. In: *Advances in neural information processing systems* 13 (2000).

[Mas+11] Jonathan Masci et al. “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction”. In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by Timo Honkela et al. Vol. 6791. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2011, pp. 52–59. DOI: 10.1007/978-3-642-21735-7_7. URL: https://doi.org/10.1007/978-3-642-21735-7_7.

[Bio+12] José M. Bioucas-Dias et al. “Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5.2 (2012), pp. 354–379. DOI: 10.1109/JSTARS.2012.2194696.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[Che+20] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG]. URL: https://arxiv.org/abs/2002.05709.

[Duf+21] Benoit Dufumier et al. “Contrastive learning with continuous proxy meta-data for 3D MRI classification”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24*. Springer. 2021, pp. 58–68.

[PUS21] Burkni Palsson, Magnus O. Ulfarsson, and Johannes R. Sveinsson. “Convolutional Autoencoder for Spectral–Spatial Hyperspectral Unmixing”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.1 (2021), pp. 535–549. DOI: 10.1109/TGRS.2020.2992743.

[PSU22] Burkni Palsson, Johannes R. Sveinsson, and Magnus O. Ulfarsson. “Blind Hyperspectral Unmixing Using Autoencoders: A Critical Comparison”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 1340–1372. DOI: 10.1109/JSTARS.2021.3140154.

[Zha+22] Lin Zhao et al. “Hyperspectral Image Classification With Contrastive Self-Supervised Learning Under Limited Labeled Samples”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. DOI: 10.1109/LGRS.2022.3159549.