An explainable multi-source unsupervised domain adaptation framework using contrastive learning and adaptive clustering for remote sensing scene classification

Binu Jose A[1], Pranesh Das[1,*], Ebrahim Ghaderpour[2] and Paolo Mazzanti[2]

[1]Machine Learning Laboratory, Department of CSE, National Institute of Technology Calicut, Kerala, India, 673601
[2]Department of Earth Sciences, Sapienza University of Rome, Piazzale Aldo Moro 5, 00185, Rome, Italy

Abstract

Unsupervised domain adaptation (UDA) has emerged as a promising approach to address domain shifts in remote-sensing scene classification. The acquisition of labelled data from diverse geographic, temporal, and sensor domains often presents significant challenges, rendering the UDA an essential tool for real-world applications. Traditional UDA methodologies typically focus on single-source domains. However, real-world scenarios frequently involve multi-source domains with diverse distributions, which introduce additional challenges such as inter-source discrepancy, label noise, class imbalance and explainability. To address these challenges, an explainable multi-source UDA framework is proposed which integrates feature extraction through contrastive-learning with an adaptive clustering-based pseudo-labeling named as XMUDA-CLAC. The pseudo-label generation process is further refined through a multi-objective optimization approach. To enhance transparency and interpretability, Explainable Artificial Intelligence (XAI) methodologies are employed to visualize the attention maps generated by contrastive learning-based Vision Transformer (ViT). The proposed XMUDA-CLAC framework is assessed using four benchmark remote sensing datasets—AID (A), NWPU-RESISC45 (N), PatternNet (P), and UC Merced (U)—under various domain-shift scenarios: (A $\rightarrow$ U), (P $\rightarrow$ N), (U $\rightarrow$ P), (A, P $\rightarrow$ U), (A, N $\rightarrow$ U), (P, U $\rightarrow$ N), (A, P, N $\rightarrow$ U), and (A, U, P $\rightarrow$ N). In this context, the proposed method demonstrates absolute accuracy improvements of 0.62%, 0.94%, 0.73%, 0.41%, 0.51%, 1.66%, 0.20%, and 0.10% over the best-performing baselines, respectively. This illustrates the efficacy of the framework in scenarios involving multi-source domain adaptation with better interpretability. The source code is available at https://github.com/BinuJoseA/XMUDA-CLAC.

Keywords

adaptive incremental density-based clustering, contrastive-learning, explainability, multi-source UDA, pseudo-labeling

1. Introduction

The UDA-based methods become an essential approach for mitigating the issue of performance degradation caused by domain shifts [1]. This degradation occurs when machine learning models, initially trained on labelled source data, are applied to distinct target domains. In the field of remote sensing, applications such as land-cover classification, disaster monitoring, and urban planning are heavily reliant on labelled datasets [2]. However, the process of annotating data for each new domain is both costly and labor-intensive [3]. UDA mitigates this challenge by aligning the feature distributions between the labelled source and the unlabelled target domains, thereby improving generalization [4].

In the field of UDA, most existing approaches are designed for single-source domains. Nevertheless, real-world scene classification frequently necessitates multi-source UDA (MUDA), where data are derived from various domains with unique distributions [5]. MUDA consists of additional challenges, including domain discrepancies, label noise, and class imbalance, all of which impede effective domain

Workshop on AI-driven Data Engineering and Reusability for Earth and Space Sciences (DARES'25), co-located with the 28th European Conference on Artificial Intelligence (ECAI 2025), Bologna, Italy, October 25, 2025.
*Corresponding author.
binujose_p200050cs@nitc.ac.in (B. J. A); praneshdas@nitc.ac.in (P. Das); ghaderpour@uniroma1.it (E. Ghaderpour); paolo.mazzanti@uniroma1.it (P. Mazzanti)
0000-0001-9325-252X (B. J. A); 0000-0002-4375-676X (P. Das); 0000-0002-5165-1773 (E. Ghaderpour); 0000-0003-0042-3444 (P. Mazzanti)

alignment [6]. Recent advancements in MUDA have addressed inter-source variation through methodologies such as domain-specific normalization, adversarial disentanglement, and attention fusion. In [7], M^3SDA implements moment matching across various sources, whereas MFSAN [8] utilizes multiple classifiers. Additionally, the methods presented in [9] and [10] employ singular value decomposition and graph neural networks, respectively, to model domain discrepancies. Nonetheless, these studies predominantly concentrate on feature alignment, often neglecting the dynamic nature of pseudo-labels and the potential benefits of explainability.

Recent advancements in UDA, particularly those incorporating contrastive learning and Vision Transformers (ViTs) [11], have demonstrated promising outcomes. Nevertheless, several significant research gaps remain inadequately addressed. Current UDA methodologies often rely on static clustering or heuristic pseudo-labeling strategies [12], which prove insufficient for managing evolving feature distributions and complex inter-domain variations, especially in multi-source remote sensing contexts. Furthermore, many approaches treat domain alignment, clustering, and pseudo-labeling as distinct processes, failing to exploit their interdependence within a unified optimization framework [13]. A significant issue is the frequent neglect of interpretability, leaving critical questions unanswered regarding the assignment of specific pseudo-labels or the achievement of domain alignment. This lack of transparency diminishes trust and limits practical applicability. Consequently, there is an urgent need for a cohesive MUDA framework capable of adaptively modeling dynamic target distributions, jointly optimizing multiple objectives for reliable pseudo-labeling, and incorporating XAI techniques such as Grad-CAM and Rollout [14] to elucidate model decisions. Addressing these gaps would substantially enhance the robustness, accuracy, and transparency of domain adaptation in remote sensing scene classification.

To address these challenges, the proposed XMUDA-CLAC framework integrates contrastive learning-based feature extraction, adaptive incremental clustering, pseudo-label generation and class-aware pseudo-label refinement through a multi-objective optimization technique. This cohesive design not only enhances domain alignment and pseudo-label quality but also improves model interpretability through attention visualization. By uniting these components, our approach offers a robust, scalable, and interpretable solution for remote sensing scene classification under domain shift conditions.

The major contributions of the paper are as follows.

1. A contrastive learning-based feature extraction mechanism for acquiring domain-invariant representations in UDA.
2. An adaptive incremental clustering module designed to produce interpretable high-quality pseudo-labels.
3. A multi-objective optimization strategy aimed at enhancing cluster reliability, pseudo-label consistency and domain alignment.
4. A class-aware pseudo-label refinement mechanism alongside dynamic centroid alignment to address issues of class imbalance, mode collapse, and temporal feature drift.
5. An XAI component for visualizing ViT attention maps and interpreting focused decisions, thereby augmenting transparency and trust in model predictions.

2. Proposed Framework

The XMUDA-CLAC framework, as depicted in Figure 1, introduces a MUDA approach for remote sensing scene classification based on substantial domain shifts. Initially, a feature encoder is pretrained using a contrastive learning model on both the source and target datasets to acquire domain-invariant representations. These features are subsequently extracted for all source domains and unlabelled target domain. A source classifier is trained on the labelled source features, and the source class centroids are calculated. An adaptive incremental density-based clustering algorithm is then employed on the target features to assign pseudo-labels by aligning the cluster centroids with the mean feature vector. Simultaneously, a domain discriminator with a gradient reversal layer (GRL) facilitates domain alignment to produce indistinguishable features across the domains. The training process is directed by four

Table 1
Objective functions and parameters used

Objective Function	Formula
Supervised classification loss [15]	$\mathcal{L}_{\text{cls}} = -\frac{1}{N_s}\sum_{i=1}^{N_s} y_i^{(s)} \log\left(g_\phi\left(f_\theta\left(x_i^{(s)}\right)\right)\right)$
Domain alignment loss [15]	$\mathcal{L}_{\text{adv}} = -\frac{1}{N_s+N_t}\sum_{i=1}^{N_s+N_t}\left[d_i \log D(f_\theta(x_i)) + (1-d_i)\log(1-D(f_\theta(x_i)))\right]$
Clustering compactness loss	$\mathcal{L}_{\text{clust}} = \sum_{k=1}^{K}\sum_{x\in C_k}\left\|f_\theta(x)-\mu_k\right\|^2$
Pseudo-label consistency loss [15]	$\mathcal{L}_{\text{pl}} = -\frac{1}{N_t}\sum_{j=1}^{N_t}\hat{y}_j^{(t)} \log\left(g_\phi(f_\theta(x_j^{(t)}))\right)$

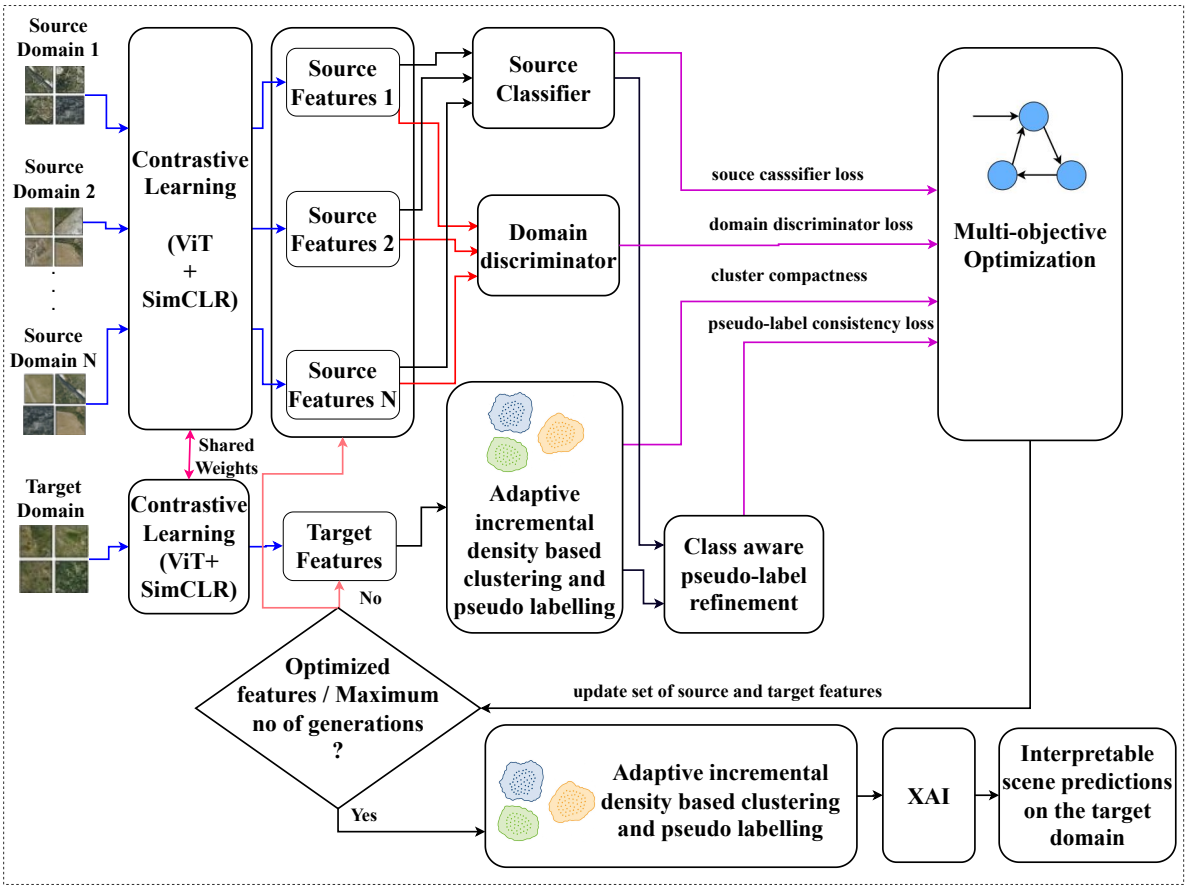


Figure 1: Proposed XMUDA-CLAC framework

objective functions: supervised classification loss on the source domain ($\mathcal{L}_{\text{cls}}$), adversarial alignment loss ($\mathcal{L}_{\text{adv}}$), clustering compactness loss ($\mathcal{L}_{\text{clust}}$), and pseudo-label consistency loss on the target ($\mathcal{L}_{\text{pl}}$). These are collectively optimized using a multi-objective strategy, such as deep learning-based pareto-front generation. Finally, interpretability is achieved through Grad-CAM, providing insights into both scene predictions and pseudo-label assignments in the target domain. The description of the objective functions are presented in Table 1. The notations used in Table 1 are N_s: number of labelled source samples, $x_i^{(s)}$: i-th source sample, $y_i^{(s)}$: ground truth label of the i-th source sample, $f_\theta(\cdot)$: feature extractor with parameters θ, $g_\phi(\cdot)$: classifier network with parameters ϕ, $D(f_\theta(x_i))$ is the domain discriminator output, $d_i = 1$ if $x_i \in$ source and $d_i = 0$ if $x_i \in$ target, K: number of clusters, x: a sample in cluster C_k, $f_\theta(x)$: Feature vector of sample x_i from feature extractor, μ_k: is the mean feature vector of cluster k, $x_j^{(t)}$: j-th target sample, $y_j^{(t)}$: pseudo-label of the j-th sample, $f_\theta(\cdot)$: feature extractor with parameters θ, and $sg_\phi(\cdot)$: classifier network with parameters ϕ.

The components in the proposed framework are detailed in subsequent subsections.

2.1. Multi-Source and Target Domains

The source domains ($S1, S2, ..., Sn$) comprise several labelled datasets derived from diverse remote sensing sources, each distinguished by variations in geographic location, acquisition time, and sensor type. Conversely, the target domain (T) is an unlabelled dataset that requires domain adaptation to facilitate accurate scene classification despite distributional changes.

2.2. Contrastive self-supervised pretraining and Feature extraction

Images from both the source and target domains are encoded using a SimCLR-based contrastive learning framework [16], to acquire domain-invariant representations. A ViT, pretrained through contrastive learning, is subsequently employed to extract high-level semantic features that demonstrate robustness to domain shifts. These features are then input into three parallel modules: source classifier, domain discriminator, and clustering module, facilitating classification, domain alignment, and pseudo-label generation.

2.3. Source classifier and Domain discriminator

A multilayer perceptron (MLP) functions as the source classifier to examine and categorize various features or characteristics of the source domain. The Adversarial Domain Discrepancy Gradient Reversal Layer (ADD-GRL) [17] is employed to mitigate the domain discrepancy between the features of the source and target domains.

2.4. Adaptive incremental density-based clustering

The adaptive incremental cluster formation with dynamic density estimation and neural network-based merging algorithm, presented as Algorithm 1. The algorithm is initiated by calculating the global average distance to inform the local parameter selection. Each incoming feature vector adaptively determines the k neighbours based on the local distance distribution and its relationship to the global threshold. This process facilitates the dynamic estimation of the neighbourhood radius ($epsilon$) and local density, from which sample-specific $MinPoints$ are derived. Subsequently, the algorithm identifies $epsilon$-neighbours and evaluates whether the sample qualifies as a core point. If so, the sample is either incorporated into an existing cluster, initiates a new cluster, or prompts merging when multiple clusters overlap. Merging decisions are executed using a two-stage neural network. The initial model computes a merge score by evaluating cluster proximity, density, and feature similarity. Subsequently, the second model dynamically adjusts the merging threshold based on these inputs and the computed score. Only the pairs that surpassed the threshold are merged. Samples that do not meet core criteria are initially classified as noise and are later re-evaluated during post-processing for potential cluster reassignment based on updated local densities.

2.5. Unsupervised deep learning-based multi-objective optimization

The proposed framework integrates unsupervised deep learning with multi-objective optimization to generate Pareto front rankings, as detailed in our previous work [18]. To enhance the diversity and generalization within the feature space, crowding distance-based selection [19] is employed to ensure a well-distributed set of solutions, which is crucial for UDA in geo-spatial contexts. Simulated Binary Crossover (SBX) [20] is used to effectively balances exploration and exploitation by preserving linear relationships among parent solutions, thereby aiding spatial and spectral coherence. Furthermore, polynomial mutation [21] is also used to reinforce spatial and spectral consistency, aligns adapted features with inherent geo-spatial structures, and enhances cross-domain generalization.

Algorithm 1: Adaptive incremental cluster formation with dynamic density estimation and neural network-based merging

Input : Target feature set $F_t = \{f_1, f_2, \ldots, f_n\}$, new feature vector f_{new}, initial clusters $\mathcal{C}$, scaling factor α_1, adaptive range constants n_1, n_2, n_3, n_4

Output: Updated cluster list $\mathcal{C}_{\text{updated}}$

// Precompute global statistics

Compute pairwise distance matrix D across F_t ;

Compute global distance mean $T = \frac{1}{n(n-1)} \sum_{i \neq j} D_{ij}$;

foreach $f_{new} \in F_t$ **do**

  // Estimate Local Distance Characteristics

  Let $S = \{D(f_{\text{new}}, f_j) \mid f_j \in F_t\}$;

  if *mean(S)* $\leq T$ **then**

    Select $k \sim \text{Uniform}(n_1, n_2)$;

  else

    Select $k \sim \text{Uniform}(n_3, n_4)$;

  Sort S in ascending order ;

  Set $epsilon = S[k]$;

  // Infer Local Density

  Let $N_{epsilon} = \{f_j \in F_t \mid D(f_{\text{new}}, f_j) \leq epsilon\}$;

  Compute local density $\rho = \frac{|N_{epsilon}|}{epsilon}$;

  Compute adaptive threshold MinPts $= \alpha_1 \cdot \rho$;

  // Decision: Assign or Evaluate

  if $|N_{epsilon}| \geq$ *MinPts* **then**

    Identify intersecting clusters $\mathcal{C}_{\text{near}} = \{C_i \in \mathcal{C} \mid N_{epsilon} \cap C_i \neq \emptyset\}$;

    if $|\mathcal{C}_{near}| = 0$ **then**

      Create new cluster $C_{\text{new}} = \{f_{\text{new}}\}$, add to $\mathcal{C}$;

    else if $|\mathcal{C}_{near}| = 1$ **then**

      Append f_{new} to the matched cluster ;

    else

      foreach *pair* $(C_a, C_b) \subseteq \mathcal{C}_{near}$ **do**

        Extract features: proximity, compactness, cross-similarity ;

        Use trained neural model to compute: ;

        merge_score $\leftarrow \text{Net}_1(\cdot)$, $\theta \leftarrow \text{Net}_2(\cdot)$;

        if *merge_score* $\geq \theta$ **then**

          Merge $C_a \cup C_b$ and add f_{new} ;

  else

    // Handle potential noise

    if *none of* $N_{epsilon}$ *belongs to any cluster* **then**

      Mark f_{new} as temporary noise ;

    else

      Find the nearest neighbor $f_{nn} \in N_{epsilon} \cap C_j$;

      Assign f_{new} to cluster of f_{nn} ;

// Noise Re-Assessment Phase

foreach *point* p *previously labelled as noise* **do**

  Recompute neighbors N_p within local $epsilon_p$;

  if $|N_p| \geq$ *MinPts*$_p$ **then**

    Assign p to the nearest valid cluster ;

return $\mathcal{C}_{\text{updated}}$

2.6. Pseudo-label generation and refinement

The class-aware adaptive pseudo-labeling algorithm, presented as Algorithm 2, offers a robust approach to MUDA without dependence on target domain centroids. The algorithm is initiated by calculating class-specific prototypes from labelled source features. During each training cycle, soft pseudo-labels are allocated to target samples based on their similarity to these prototypes, utilizing a scaled softmax function. The reliability of pseudo-labels is assessed using cluster-wise confidence scores, which are derived from adaptive incremental density-based clustering. Only samples within high-confidence clusters are fully accepted, while others are incorporated with reduced weight. To ensure class balance, the top-k most confident samples per class are selected and employed to update the source prototypes in a weighted manner, facilitating gradual adaptation to the target domain. Subsequently, prototype-based contrastive loss is computed to align pseudo-labelled target features with their corresponding prototypes. The classifier is trained using a combined loss: a standard classification loss for confident samples and a contrastive loss for alignment.

2.7. Explainable AI (XAI) module and target prediction

The contrastive learning model, adapted through UDA, effectively employs both Grad-CAM-based activations and global Rollout to make informed decisions regarding scene classification. Upon determining the optimal set of source and target features, the adaptive incremental density-based clustering algorithm is applied to the optimal target features to produce pseudo-labels. These pseudo-labels are then employed to further train the classifier, incorporating both the optimal source and target features.

3. Experimental setup

In order to evaluate performance, four prominent datasets for remote sensing scene classification have been chosen: AID (A) [22], NWPU-RESISC45 (N) [23], PatternNet (P)[24] and UC Merced (M) [25]. To ensure a uniform basis for comparison, five shared classes such as Farmland, Forest, Parking, Residential and River present in all four datasets are utilized. Experiments are conducted on an NVIDIA DGX Station A100 equipped with an AMD EPYC 7742 64-core CPU, four NVIDIA A100 (40 GB) GPUs, and 512 GB of DDR4 RAM. Training is conducted for 200 epochs, each comprising 100 genetic-algorithm generations. The crossover and mutation rates are fixed at 0.82 and 0.018, respectively. The neighbor-rank parameters n_1, n_2, n_3, and n_4 are assigned the values 5, 10, 20, and 50. The density scale α_1 is computed as the standard deviation of local neighbor distances. The confidence gate α_2 and the prototype-contrastive weight λ_{proto} values are in the range of $[0, 1]$. The description of the hyper-parameters is presented in Table 2.

4. Results and performance analysis

The performance analysis of XMUDA-CLAC is conducted using the classification accuracy, average Receiver Operating Characteristic (ROC) curve, computational cost and worst-case time complexity. Table 3 presents a comparative classification accuracy analysis of several state-of-the-art UDA methods applied to remote sensing scene classification. In the (A $\rightarrow$ U) task, the proposed framework achieves an accuracy of 0.965, representing a 0.62% improvement over the next best method (0.959). In the (P $\rightarrow$ N) task, the proposed method surpasses Hy-MSDA (0.953) with an accuracy of 0.962, indicating a 0.94% increase. In the (U $\rightarrow$ P) task, the proposed method also demonstrates significant improvements, with an increase of 0.73%. Specifically, for multi-source domain tasks such as (A, P $\rightarrow$ U), (A, N $\rightarrow$ U), (P, U $\rightarrow$ N), (A, P, N $\rightarrow$ U), and (A, U, P $\rightarrow$ N), the proposed method exhibits superiority over existing methods. The findings highlight the effectiveness of the proposed method in improving accuracy across various domain adaptation tasks when compared to other leading UDA methods.

Figure 2 illustrates the ROC curves for five classifiers: support vector machine (SVM), multi-layer perceptron (MLP), XGBoost, random forest, and logistic regression. These classifiers were assessed on

Algorithm 2: Class-aware adaptive pseudo-labeling refinement with class-wise filtering and dynamic source updates

Input: Source features F_s with labels Y_S, Target features F_t, Target clusters C_T (from Algorithm 1), Top-k value k, threshold τ, Contrastive loss weight λ_{proto}, Scaling factor α_2

Output: Refined pseudo-labels and trained classifier

// Initialize Source Class Prototypes

for *each source class* $s \in Y_S$ **do**

  Compute initial prototype $\mu_s^{(0)} = \frac{1}{|F_s^s|} \sum_{x \in F_s^s} f(x)$;

// Iteratively Update Pseudo-Labels and Source Prototypes

for *each training epoch* m **do**

  // Assign Soft Pseudo-Labels with Class-Wise Top-k Filtering

  Initialize $\mathcal{P}[s] = \emptyset$ for each class s;

  // Pseudo-label generation

  for *each target sample* $x_i \in F_t$ **do**

    Identify cluster c_i of x_i from C_T;

    Compute soft pseudo-label probabilities $P(y_i = s)$ using:

$$P(y_i = s) = \frac{\exp(\text{sim}(f(x_i), \mu_s^{(m-1)})/\tau)}{\sum_{j=1}^{Q} \exp(\text{sim}(f(x_i), \mu_j^{(m-1)})/\tau)}$$

    Let s^* be pseudo-label;

    $s^* = \arg\max P(y_i = s)$;

    // Class-aware pseudo-label refinement

    Compute cluster-level confidence γ_{c_i} using intra-cluster similarity or density;

    Compute threshold $\tau_{c_i} = \alpha_2 \cdot \text{mean}(\gamma_{c_i})$;

    if $\gamma_{c_i} \geq \tau_{c_i}$ **then**

      Add $(x_i, P(y_i = s^*), f(x_i), \textbf{weight} = 1.0)$ to $\mathcal{P}[s^*]$;

    else

      Add $(x_i, P(y_i = s^*), f(x_i), \textbf{weight} = 0.5)$ to $\mathcal{P}[s^*]$;

  // Top-k Selection

  for *each class* s **do**

    Sort $\mathcal{P}[s]$ by confidence and retain top-k samples;

  // Update Source Prototypes from Top-k Pseudo-Labelled Target Samples

  for *each class* s **do**

    Compute updated prototype $\mu_s^{(m)} = \frac{\sum_{(x_i, w_i)} w_i \cdot f(x_i)}{\sum w_i}$ from top-k $\mathcal{P}[s]$;

  // Compute Contrastive Loss

  for *each pseudo-labelled sample* $(x_i, f(x_i))$ **do**

    Compute:

$$\mathcal{L}_{\text{proto}}(x_i) = -\log \frac{\exp(\text{sim}(f(x_i), \mu_{s^*}^{(m)})/\tau)}{\sum_{j=1}^{Q} \exp(\text{sim}(f(x_i), \mu_j^{(m)})/\tau)}$$

  // Classifier Training

  Compute cross-entropy loss $\mathcal{L}_{\text{cls}}$ over confident samples;

  Total loss: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{proto}} \cdot \mathcal{L}_{\text{proto}}$;

  Update network parameters using $\mathcal{L}_{\text{total}}$;

return Refined pseudo-labels s^*;

domain adaptation tasks (A, P, N $\rightarrow$ U) and (A, U, P $\rightarrow$ N). Each curve depicts the mean performance across all five scene classes. The Area Under the Curve (AUC) scores indicate that the MLP classifier achieves the highest performance, with AUC values of 0.99 and 0.98 for the respective tasks.

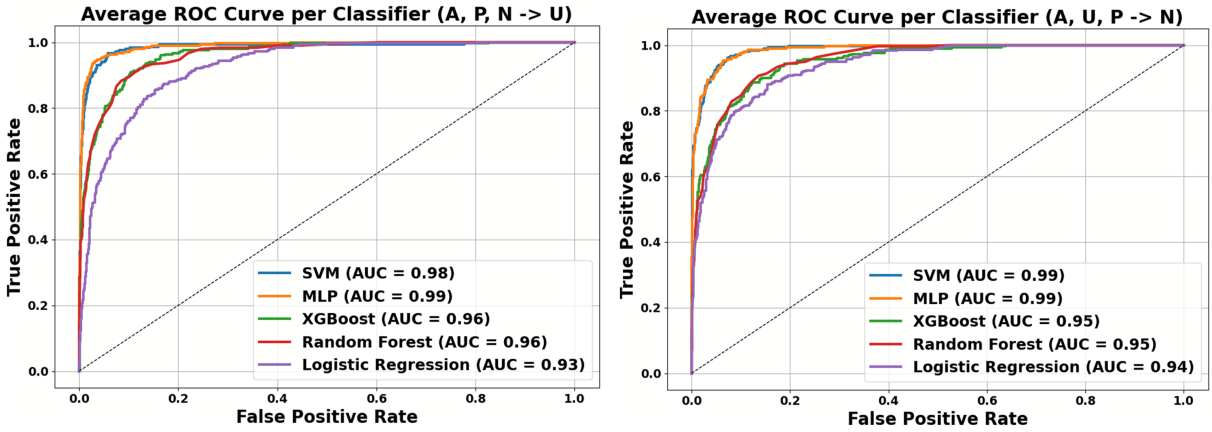
Table 4 presents a comparative computational cost analysis using parameter count, GFLOPs, model size, and training time for (A, U, P $\rightarrow$ N) across various state-of-the-art models. Although the XMUDA-

Table 2
Hyper-parameters, roles, search spaces, selection rules, and chosen values.

Param	Role	Search space	Selection rule	Chosen
n_1, n_2	dense-region neighbor rank	$[3, 8], [8, 12]$	best proxy rank-sum	$5, 10$
n_3, n_4	sparse-region neighbor rank	$[15, 30], [40, 60]$	best proxy rank-sum	$20, 50$
λ_{proto}	prototype-contrastive weight	$[0.1, 0.6]$	entropy is minimized	0.3
Top-k	per-class target selection	$\{10, 20, 50\}$	stability vs. coverage	20

Table 3
Comparison of classification accuracy on multi-source domain adaptation methods across various domain combinations

Domain	M^{3}SDA [7]	MFSAN [8]	LCt-MSDA [10]	T-SVDNet [9]	MCC-DA [26]	PTMDA [27]	SUMDA [28]	RRL [29]	Hy-MSDA [15]	XMUDA-CLAC
(A → U)	0.887	0.912	0.873	0.854	0.940	0.944	0.944	0.946	0.959	**0.965**
(P → N)	0.870	0.907	0.868	0.855	0.931	0.928	0.939	0.937	0.953	**0.962**
(U → P)	0.879	0.910	0.870	0.859	0.938	0.930	0.933	0.933	0.947	**0.954**
(A, P → U)	0.883	0.919	0.890	0.865	0.950	0.951	0.949	0.944	0.968	**0.972**
(A, N → U)	0.895	0.920	0.905	0.860	0.945	0.948	0.950	0.951	0.972	**0.977**
(U, P → N)	0.917	0.940	0.908	0.881	0.967	0.968	0.965	0.962	**0.978**	**0.978**
(A, P, N → U)	0.901	0.923	0.898	0.869	0.957	0.954	0.954	0.955	0.974	**0.976**
(A, U, P → N)	0.922	0.928	0.886	0.884	0.964	0.960	0.963	0.955	0.977	**0.978**



(a) (A, P, N → U) (b) (A, U, P → N)

Figure 2: Classifier performance (AUC-ROC) across MUDA tasks (a) (A, P, N → U) (b) (A, U, P → N).

Table 4
Computational cost analysis for (A, U, P → N) (↓ indicates lower is better).

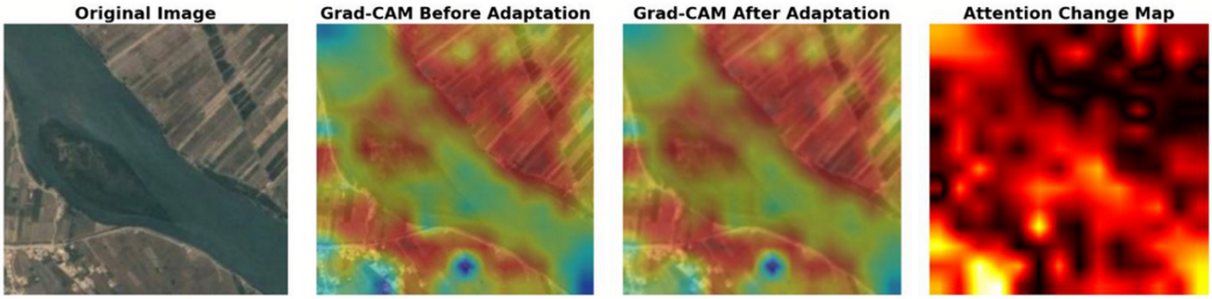
Models	Parameters ($\times 10^6$)↓	GFLOPs P_s↓	Size (MB)↓	Training Time (h)↓
PTMDA [27]	30.5	32.5	141.5	9.4
SUMDA [28]	31.1	35.6	160.5	14.0
RRL [29]	30.8	32.1	146.5	8.2
Hy-MSDA [15]	29.5	30.2	113.8	**7.8**
XMUDA-CLAC	**27.8**	**28.6**	**110.4**	8.1

CLAC achieves significant classification accuracy utilizing ViT+SimCLR, the slight increase in training time is negligible, which offers scalable solution compared to some of the existing benchmark models.

The comparative time complexity analysis of XMUDA-CLAC with some of the state-of-art approaches are detailed in Table 5. The notations used in Table 5 are N_s: number of source images, N_t: number of target images, d: feature width, C: number of classes, K: number of target clusters, E_{pre}, E_{cls}, E_{adv} are the epochs for SimCLR pretrain, source classifier, adversarial alignment, G: number of generations,

Table 5
Comparative time complexity analysis

Method	Time complexity
Hy-MSDA [15]	$T_{\text{total}} = O(E\,(N_s + N_{sd}N_t)\,F_{\text{bwd}}) + O(N_{sd}N_t\,F_{\text{fwd}}) + O(N_{sd}^2 N_t)$
RRL [29]	$T_{\text{total}} = O(E_s N_s F_{\text{bwd}} + N_t F_{\text{fwd}}) + E \cdot O\Big((N_s + N_t)F_{\text{bwd}} + b^2 + N_t F_{\text{fwd}}\Big)$
XMUDA-CLAC	$T_{\text{total}} = O\Big(E_{\text{pre}}(N_s + N_t)F_{\text{ViT}} + (N_s + N_t)F_{\text{ViT}} + E_{\text{cls}}N_s F_c + E_{\text{adv}}(N_s + N_t)\,(F_{\text{ViT}} + F_d)\Big) + G\,T_{\text{gen}}$
	$T_{\text{gen}} = O\Big(N_t F_{\text{ViT}} + N_t \log N_t + N_t C d + P^2 M\Big)$



(a) (b) (c) (d)

Figure 3: Visualization of attention shift before and after domain adaptation in (A, U, P → N)

P: population size , M: number of objectives, F_{ViT}: ViT per sample cost, F_c: classifier cost and F_d: discriminator cost, N_{sd}: number of source domains, E: training epochs, F_{fwd}: per-sample cost of a forward pass, F_{bwd}: per-sample cost of one forward+backward pass, E_s: number of epochs for the initial source-only pretraining, E: number of outer training epochs, and b: mini-batch size. From the table, it is observed that the time complexity of the proposed approach is at par with existing models, while performing well with respect to classification accuracy.

4.1. Explainability of domain shift and target scene prediction

The explainability of the XMUDA-CLAC are performed using Grad-CAM and Attention Rollout methods and are presented in Figures 3 and 4. Figure 3 depicts the alteration in attention distribution within a contrastive learning-based ViT model, observed before and after domain adaptation on a target image from NWPU-RESISC45 dataset. Figure 3a shows the original river scenario. Figure 3b presents the Grad-CAM output from the source-only model, which is trained on AID, UC Merced, and PatternNet, highlighting the initial regions of attention. Figure 3c displays the attention map post-adaptation. The regions in red/yellow signify high attention, whereas blue/green regions indicate low attention. Figure 3d exhibits the Attention Change Map, where red/yellow areas highlight shifts in attention and black areas denote stable focus, underscoring the interpretability improvements resulting from adaptation. Figure 4 presents a visual elucidation of scene prediction on a target image from NWPU-RESISC45 dataset utilizing a contrastive learning-based UDA model.

Figure 5 presents a two-dimensional UMAP projection that illustrates the alignment between the source domains, target domain, updated centroids, and top-k filtered pseudo-labelled samples in a UDA scenario (A, U, P → N). Source samples are represented as circles, target samples as crosses, top-k samples are outlined with circular borders, and the updated centroids are depicted as black pentagons.

5. Ablation study

Table 6 presents an ablation study evaluating the classification accuracy of various combinations of feature extractors and objective functions for UDA across multiple remote sensing datasets. The study

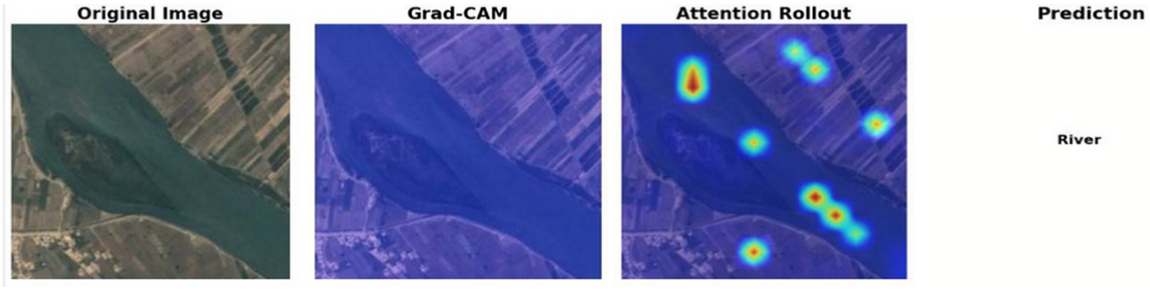


Figure 4: Visual explanation of target scene prediction Using Grad-CAM and Attention Rollout (A, U, P → N)

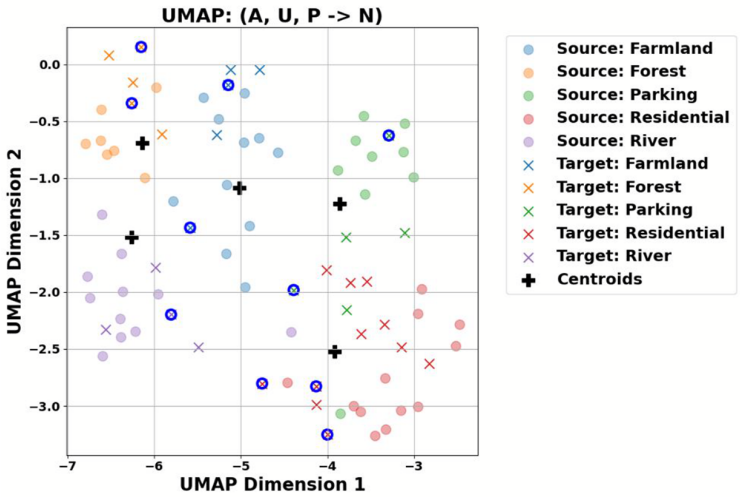


Figure 5: UMAP Visualization of Source-Target Alignment and Pseudo-Label Distribution in (A, U, P → N)

Table 6
An ablation study on classification accuracy to evaluate the impact of different feature extractors and objective function combinations for UDA

Feature extractor	$\mathcal{L}_{cls}$	$\mathcal{L}_{adv}$	$\mathcal{L}_{clust}$	$\mathcal{L}_{pl}$	(A → U)	(P → N)	(U → P)	(A, P → U)	(A, N → U)	(P, U → N)	(A, P, N → U)	(A, U, P → N)
ResNet50	✓	✓	✓		0.894	0.883	0.886	0.876	0.895	0.901	0.891	0.887
	✓	✓	✓	✓	0.927	0.903	0.914	0.903	0.914	0.923	0.907	0.915
ViT	✓	✓	✓		0.901	0.894	0.894	0.903	0.904	0.915	0.901	0.908
	✓	✓	✓	✓	0.943	0.914	0.928	0.923	0.921	0.934	0.935	0.932
ViT + SimCLR	✓	✓	✓		0.921	0.913	0.903	0.915	0.903	0.907	0.924	0.905
	✓	✓	✓	✓	0.965	0.962	0.954	0.972	0.977	0.978	0.976	0.978

examines four objective functions, utilizing feature extractors such as ResNet50, Vision Transformer (ViT), and ViT pretrained with SimCLR. Among the configurations tested, the combination of ViT with SimCLR and the full set of objective functions achieves the highest classification accuracy across all eight domain adaptation tasks, including both single-source and multi-source scenarios. Notably, it attains superior performance on tasks such as (A → U) (0.965), (A, N → U) (0.977), and (A, U, P → N) (0.978). These results highlight the efficacy of transformer-based representations enhanced by contrastive learning and the synergistic effect of multiple complementary objective functions.

6. Conclusion

This study introduces a comprehensive XMUDA-CLAC framework for remote sensing scene classification in the context of significant domain shifts. By integrating contrastive-pretrained ViTs with adaptive

incremental density-based clustering, the framework effectively extracts domain-invariant features and generates high-confidence pseudo-labels for the unlabelled target domain. The robustness to class imbalance and feature drift is further enhanced through class-aware pseudo-label refinement and dynamic centroid alignment. By framing pseudo-labeling, clustering, and domain alignment as a unified multi-objective optimization problem, the framework facilitates reliable learning in the absence of target labels. Explainability is incorporated through XAI techniques, such as Grad-CAM and attention rollout, thereby improving the interpretability and trustworthiness of model predictions. Experimental results demonstrate the framework's superior accuracy and generalization compared with state-of-the-art UDA methods. Although this approach incurs additional computational cost due to its optimization complexity, it offers solution with better classification accuracy, scalability and interpretability. Future research endeavors will focus on developing a multi-source universal domain adaptation (DA) variant of XMUDA-CLAC, incorporating minimum-cost-flow matching and conformally calibrated energy-based mechanisms for unknown rejection.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] E. Haghighi Gashti, H. Bahiraei, M. J. Valadan Zoej, E. Ghaderpour, Fusion of aerial and satellite images for automatic extraction of building footprint information using deep neural networks, Information 16 (2025) 380.

[2] I. Papoutsis, N. I. Bountos, A. Zavras, D. Michail, C. Tryfonopoulos, Benchmarking and scaling of deep learning models for land cover image classification, ISPRS Journal of Photogrammetry and Remote Sensing 195 (2023) 250–268.

[3] R. Naushad, T. Kaur, E. Ghaderpour, Deep transfer learning for land use and land cover classification: A comparative study, Sensors 21 (2021) 8083.

[4] Y. Xu, H. Cao, L. Xie, X.-l. Li, Z. Chen, J. Yang, Video unsupervised domain adaptation with deep learning: A comprehensive survey, ACM Computing Surveys 56 (2024) 1–36.

[5] P. Singhal, R. Walambe, S. Ramanna, K. Kotecha, Domain adaptation: challenges, methods, datasets, and applications, IEEE access 11 (2023) 6973–7020.

[6] L. Ding, D. Hong, M. Zhao, H. Chen, C. Li, J. Deng, N. Yokoya, L. Bruzzone, J. Chanussot, A survey of sample-efficient deep learning for change detection in remote sensing: Tasks, strategies, and challenges, IEEE Geoscience and Remote Sensing Magazine (2025).

[7] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1406–1415.

[8] Y. Zhu, F. Zhuang, D. Wang, Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 5989–5996.

[9] R. Li, X. Jia, J. He, S. Chen, Q. Hu, T-svdnet: Exploring high-order prototypical correlations for multi-source domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9991–10000.

[10] H. Wang, M. Xu, B. Ni, W. Zhang, Learning to combine: Knowledge aggregation for multi-source domain adaptation, in: European Conference on Computer Vision, Springer, 2020, pp. 727–744.

[11] S. K. Roy, A. Jamali, J. Chanussot, P. Ghamisi, E. Ghaderpour, H. Shahabi, Simpoolformer: A two-stream vision transformer for hyperspectral image classification, Remote Sensing Applications: Society and Environment 37 (2025) 101478.

[12] M. Litrico, A. Del Bue, P. Morerio, Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7640–7650.
[13] T. Burgert, M. Ravanbakhsh, B. Demir, On the effects of different types of label noise in multi-label remote sensing image classification, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–13.
[14] A. Abbas, M. Linardi, E. Vareille, V. Christophides, C. Paris, Towards explainable ai4eo: An explainable deep learning approach for crop type mapping using satellite images time series, in: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2023, pp. 1088–1091.
[15] K. Xu, Z. Zhu, W. Wang, C. Fan, B. Wu, Z. Jia, Enhancing remote sensing scene classification with hy-msda: A hybrid cnn-transformer for multi-source domain adaptation, IEEE Transactions on Geoscience and Remote Sensing (2024).
[16] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PmLR, 2020, pp. 1597–1607.
[17] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.
[18] B. Jose A, P. Das, An automated incremental density-based clustering approach using unsupervised deep learning and multi-objective optimization, Computers and Electrical Engineering 123 (2025) 110109.
[19] D. Feng, Y. Li, J. Liu, Y. Liu, A particle swarm optimization algorithm based on modified crowding distance for multimodal multi-objective problems, Applied Soft Computing 152 (2024) 111280.
[20] L. Pan, W. Xu, L. Li, C. He, R. Cheng, Adaptive simulated binary crossover for rotated multi-objective optimization, Swarm and Evolutionary Computation 60 (2021) 100759.
[21] J. L. Carles-Bou, S. F. Galán, Self-adaptive polynomial mutation in nsga-ii, Soft Computing 27 (2023) 17711–17727.
[22] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, X. Lu, Aid: A benchmark data set for performance evaluation of aerial scene classification, IEEE Transactions on Geoscience and Remote Sensing 55 (2017) 3965–3981.
[23] G. Cheng, J. Han, X. Lu, Remote sensing image scene classification: Benchmark and state of the art, Proceedings of the IEEE 105 (2017) 1865–1883.
[24] W. Zhou, S. Newsam, C. Li, Z. Shao, Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval, ISPRS journal of photogrammetry and remote sensing 145 (2018) 197–209.
[25] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, 2010, pp. 270–279.
[26] Y. Wei, L. Yang, Y. Han, Q. Hu, Multi-source collaborative contrastive learning for decentralized domain adaptation, IEEE Transactions on Circuits and Systems for Video Technology 33 (2022) 2202–2216.
[27] C.-X. Ren, Y.-H. Liu, X.-W. Zhang, K.-K. Huang, Multi-source unsupervised domain adaptation via pseudo target domain, IEEE Transactions on Image Processing 31 (2022) 2122–2135.
[28] M. Li, C. Zhang, W. Zhao, W. Zhou, Cross-domain urban land use classification via scenewise unsupervised multisource domain adaptation with transformer, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 17 (2024) 10051–10066.
[29] S. Chen, L. Zheng, H. Wu, Riemannian representation learning for multi-source domain adaptation, Pattern Recognition 137 (2023) 109271.