frontiers | Frontiers in Environmental Science

TYPE Original Research
PUBLISHED 12 June 2025
DOI 10.3389/fenvs.2025.1566224

OPEN ACCESS

EDITED BY
Sushant K Singh,
CAIES Foundation, India

REVIEWED BY
Xinyue Mo,
Hainan University, China
Geetha Srikanth,
Amrita Vishwa Vidyapeetham University, India

*CORRESPONDENCE
Wenbo Lin,
15101207316@163.com

RECEIVED 24 January 2025
ACCEPTED 14 May 2025
PUBLISHED 12 June 2025

CITATION
Lin W, Li T and Li X (2025) Deep learning-based object detection for environmental monitoring using big data.
Front. Environ. Sci. 13:1566224.
doi: 10.3389/fenvs.2025.1566224

Deep learning-based object detection for environmental monitoring using big data

Wenbo Lin[1]*, Tingting Li[2] and Xiao Li[3]

[1]College of Geology, Gansu Institute of Industrial Technology, Tianshui, Gansu, China, [2]School of Electronic Information, Gansu Institute of Industrial Technology, Tianshui, Gansu, China, [3]Guangdong Nonferrous Industrial Building Quality Inspection Co., Ltd, Guangzhou, Guangdong, China

Introduction: Recent advances in artificial intelligence have transformed the way we analyze complex environmental data. However, high-dimensionality, spatiotemporal variability, and heterogeneous data sources continue to pose major challenges.

Methods: In this work, we introduce the Environmental Graph-Aware Neural Network (EGAN), a novel framework designed to model and analyze large-scale, multi-modal environmental datasets. EGAN constructs a spatiotemporal graph representation that integrates physical proximity, ecological similarity, and temporal dynamics, and applies graph convolutional encoders to learn expressive spatial features. These are fused with temporal representations using attention mechanisms, enabling the model to dynamically capture relevant patterns across modalities. The framework is further enhanced by domain-informed learning strategies that incorporate physics-based constraints, meta-learning for regional adaptation, and uncertainty-aware predictions.

Results: Extensive experiments on four benchmark datasets demonstrate that our approach achieves state-of-the-art performance in environmental object detection, segmentation, and scene understanding.

Discussion: EGAN is shown to be a robust and interpretable tool for real-world environmental monitoring applications.

KEYWORDS

environmental monitoring, spatiotemporal modeling, graph neural networks, meta-learning, uncertainty quantification

1 Introduction

Environmental monitoring is critical for understanding and addressing challenges such as climate change, biodiversity loss, and resource management (Joshi et al., 2024). Traditional monitoring methods, which rely heavily on manual observation and limited data collection, are inadequate to address the complexity and scale of contemporary environmental issues. The advent of big data and remote sensing technologies has revolutionized this domain by enabling the collection of vast amounts of environmental data from satellites, drones, and IoT-enabled sensors (Nigar et al., 2024). However, the sheer volume and heterogeneity of this data pose significant challenges for effective analysis and interpretation. In this context, object detection—a fundamental computer vision task—has emerged as a key technique for identifying and tracking objects of interest, such as wildlife,

vegetation, and pollutants, in environmental datasets. To make full use of big data, deep learning-based object detection methods have become essential, offering unparalleled accuracy and efficiency in extracting actionable insights from large-scale, complex environmental datasets (Feng et al., 2024). Environmental monitoring is critical for understanding and addressing challenges such as climate change, biodiversity loss, and resource management (Joshi et al., 2024). Traditional monitoring methods, which rely heavily on manual observation and limited data collection, are inadequate to address the complexity and scale of contemporary environmental issues. The advent of big data and remote sensing technologies has revolutionized this domain by enabling the collection of vast amounts of environmental data from satellites, drones, and IoT-enabled sensors (Nigar et al., 2024). However, the sheer volume and heterogeneity of this data pose significant challenges for effective analysis and interpretation. In this context, object detection—a fundamental computer vision task—has emerged as a key technique for identifying and tracking objects of interest, such as wildlife, vegetation, and pollutants, in environmental datasets. To make full use of big data, deep learning-based object detection methods have become essential, offering unparalleled accuracy and efficiency in extracting actionable insights from large-scale, complex environmental datasets (Feng et al., 2024).

The early stages of object detection in environmental monitoring were primarily based on heuristic and rule-driven methods, where algorithms processed environmental data through a set of predefined instructions and patterns (Lv et al., 2023). These methods focused on detecting basic features such as edges, textures, and shapes, which helped identify elements like water bodies, forests, or animals in satellite imagery (Virasova et al., 2021). Although these approaches were interpretable and laid the groundwork for automation in monitoring tasks, they were limited by rigid rules and often failed to adapt to the diversity and complexity of environmental data. Additionally, their reliance on high-quality images and their sensitivity to noise and data variations made them unsuitable for large-scale environmental datasets (Yin et al., 2020).

The development of more sophisticated machine learning techniques marked a significant shift in object detection, as algorithms became capable of identifying patterns from data with less reliance on explicit human intervention (Zhang et al., 2022). Early machine learning models, such as support vector machines and random forests, improved the accuracy of object classification by leveraging features extracted from data (Li et al., 2022a). While these models reduced the need for hand-crafted rules, they still faced challenges in scaling to handle large and diverse environmental datasets, requiring complex feature extraction and often underperforming when faced with high-dimensional data (Zhu et al., 2021). The introduction of convolutional neural networks (CNNs) further advanced object detection by enabling automated learning of hierarchical features from raw images, significantly improving performance in tasks such as tracking deforestation and monitoring wildlife populations. However, these models were still computationally demanding and struggled with processing large datasets efficiently (Li et al., 2022b).

Recent breakthroughs in deep learning, combined with advances in big data analytics, have enabled real-time object detection on large-scale environmental datasets (Bai et al., 2022). Models like YOLO (You Only Look Once), Faster R-CNN, and transformer-based vision architectures now allow for high-accuracy detection in diverse environmental contexts (Liu Y. et al., 2022). These models incorporate innovations such as multi-scale feature representation and attention mechanisms, which address issues like occlusion, data variability, and noise. Moreover, the integration of deep learning with cloud computing and distributed processing systems has enhanced the scalability of environmental monitoring, enabling the processing of massive data streams from remote sensing and IoT devices (Liu J. et al., 2022). For instance, these methods have been successfully used to track illegal logging, assess urban heat islands, and monitor endangered species. Despite these advances, challenges remain, including the need for high-quality labeled data, the computational costs of training large models, and the interpretability of deep learning results, which is crucial for making informed policy decisions in environmental management (Wang et al., 2023).

To address these challenges, we propose a novel framework that combines deep learning-based object detection with big data analytics for environmental monitoring. Our approach incorporates advanced neural architectures, such as Vision Transformers (ViTs), and pre-trained models optimized for environmental datasets, enabling accurate detection across diverse ecological conditions. We employ transfer learning to mitigate the need for extensive labeled data and integrate explainability modules to enhance the interpretability of predictions. By leveraging distributed computing and edge AI, the proposed system ensures scalability and real-time processing, making it suitable for large-scale environmental monitoring tasks.

We summarize our contributions as follows:

- The proposed approach integrates cutting-edge deep learning models, such as Vision Transformers and pre-trained frameworks, with big data analytics to improve object detection accuracy and efficiency in environmental monitoring.
- Designed to handle large-scale, heterogeneous datasets, the method is highly scalable and adaptable to various environmental applications, including biodiversity monitoring, pollution detection, and resource management.
- Experimental results demonstrate significant improvements in detection accuracy, computational efficiency, and robustness under challenging conditions, validating the effectiveness of the proposed system for real-world environmental monitoring.

2 Related work

2.1 Deep learning for object detection

Deep learning has fundamentally transformed the field of object detection, providing advanced techniques for identifying and localizing objects in images and videos. In the context of environmental monitoring, deep learning-based object detection models have enabled automated analysis of vast amounts of visual data collected through remote sensing, surveillance

cameras, drones, and other IoT-enabled devices (Lou et al., 2023). These methods have demonstrated remarkable accuracy and scalability, addressing key challenges such as detecting small, occluded, or overlapping objects in complex natural environments (Liu Y.-C. et al., 2021). State-of-the-art object detection models, such as Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot Multibox Detector), have been widely adopted in environmental monitoring applications. Faster R-CNN employs a region proposal network (RPN) to generate candidate object regions, followed by classification and bounding box regression, offering high accuracy for detecting diverse objects. YOLO and SSD, on the other hand, prioritize real-time detection by using single-stage architectures, making them suitable for applications that require immediate response, such as disaster monitoring or wildlife tracking (Wang Y. et al., 2021). Recent advancements in object detection include transformer-based architectures, such as the Detection Transformer (DETR), which leverage self-attention mechanisms to model global dependencies in the input data. These models have proven effective in scenarios where the spatial arrangement of objects is crucial, such as mapping deforestation patterns or monitoring urban sprawl (Singh and Taylor, 2020). Furthermore, lightweight versions of these models, such as YOLOv5 and MobileNet-SSD, have been developed to enable deployment on resource-constrained devices, ensuring accessibility in remote and under-resourced areas (Qin et al., 2020). Environmental monitoring often involves detecting objects under challenging conditions, including varying lighting, weather, and terrain. Deep learning models address these challenges through data augmentation techniques, such as geometric transformations and photometric adjustments, to improve model robustness. Multimodal approaches that integrate data from multiple sources, such as RGB, infrared, and LiDAR sensors, have enhanced detection accuracy by providing complementary perspectives on the environment (Xie et al., 2021). Despite these advancements, several challenges remain in deploying deep learning-based object detection systems at scale. Data annotation is a significant bottleneck, as labeling environmental datasets requires domain expertise and substantial effort. To address this, researchers have explored unsupervised and semi-supervised learning techniques, such as self-training and contrastive learning, to reduce reliance on labeled data. Moreover, active learning strategies, where the model identifies uncertain samples for manual annotation, have been employed to maximize the efficiency of the labeling process (Gu et al., 2021).

2.2 Big data integration for environmental monitoring

Environmental monitoring generates massive amounts of data from diverse sources, including satellite imagery, drone footage, sensor networks, and citizen science platforms (Xu et al., 2021). The integration of these big data streams with deep learning-based object detection systems has opened new opportunities for large-scale and high-resolution monitoring of environmental changes (Wang T. et al., 2021). However, the complexity and heterogeneity of environmental big data pose significant challenges in terms of data management, preprocessing, and analysis. Data fusion techniques have been instrumental in addressing the heterogeneity of environmental data. By combining data from different modalities, such as optical imagery, radar, and multispectral data, deep learning models can leverage complementary information to improve detection accuracy (Sun et al., 2021). For example, in forest monitoring, multispectral data can help identify tree species, while LiDAR data provides detailed topographic information, enabling precise detection of deforestation or illegal logging activities (Joseph et al., 2021). Distributed computing frameworks, such as Apache Spark and Hadoop, have facilitated the processing and analysis of large-scale environmental datasets. These frameworks enable parallel computation and efficient storage of big data, ensuring scalability for applications that require continuous monitoring over large geographic areas. When combined with cloud-based platforms, such as Google Earth Engine or AWS S3, these systems provide a robust infrastructure for deploying deep learning models in real-time. The use of spatiotemporal analysis is critical in environmental monitoring, as many phenomena evolve over time (Singh et al., 2021). Deep learning models, such as spatiotemporal convolutional networks and temporal attention mechanisms, have been developed to analyze sequential data, such as time-lapse imagery or sensor readings. These models can detect trends, anomalies, and seasonal variations, providing actionable insights for environmental management (Fan et al., 2021). For instance, spatiotemporal models have been used to monitor glacier retreat, urban heat islands, and changes in biodiversity. However, the integration of big data with deep learning systems requires addressing challenges related to data quality and privacy. Environmental data often suffer from noise, missing values, and inconsistencies, which can affect model performance. Advanced data cleaning and imputation techniques, including autoencoders and generative models, have been employed to address these issues (Misra et al., 2021). Ensuring data privacy and security is critical, especially when integrating data from sensitive sources, such as citizen contributions or protected ecosystems. The deployment of big data-driven object detection systems also requires addressing the energy efficiency and environmental impact of deep learning models. Training large-scale models on extensive datasets consumes significant computational resources, contributing to carbon emissions. Researchers are increasingly focusing on developing energy-efficient architectures, such as pruning and quantization, and exploring alternative training strategies, such as federated learning, to mitigate these impacts (Han et al., 2021).

2.3 Applications in environmental monitoring

The combination of deep learning-based object detection and big data has enabled a wide range of applications in environmental monitoring, addressing critical issues such as climate change, biodiversity loss, and disaster management (Reading et al., 2021). These applications leverage the ability of object detection models to analyze large-scale datasets and provide detailed, actionable insights for policymakers, researchers, and conservationists. One of the most impactful applications is in wildlife conservation, where object detection models are used to identify and track animals in their

natural habitats (Feng et al., 2021). By analyzing drone footage or camera trap images, these models can monitor population dynamics, migration patterns, and habitat use, informing conservation strategies. For example, deep learning has been used to detect poaching activities by identifying human and vehicle presence in protected areas, enabling real-time interventions (Liu Z. et al., 2021). In agriculture, object detection systems are applied to monitor crop health, identify pest infestations, and optimize irrigation practices. By analyzing high-resolution satellite imagery or drone data, these models can detect anomalies, such as disease outbreaks or nutrient deficiencies, at an early stage, reducing crop losses and improving food security. Similarly, object detection has been used to monitor aquaculture systems, ensuring sustainable fish farming practices. Disaster management is another critical area where deep learning-based object detection has proven invaluable (Singh et al., 2022). During natural disasters, such as floods, wildfires, or hurricanes, these models can analyze real-time data from satellites and drones to assess the extent of damage and identify affected areas (Carion et al., 2020). This information is crucial for coordinating rescue operations and allocating resources effectively. For instance, object detection models have been used to map wildfire perimeters and monitor their progression, aiding firefighting efforts. Climate change monitoring relies heavily on object detection systems to analyze environmental changes over time (Zhu et al., 2020). By detecting deforestation, glacier retreat, and urban expansion, these models provide valuable data for understanding the drivers and impacts of climate change. For example, deep learning has been used to map deforestation in the Amazon rainforest, identifying hotspots of illegal logging and informing conservation policies. Challenges in these applications include the need for domain-specific adaptations and real-time processing capabilities. Environmental monitoring often requires specialized models that can detect rare or subtle objects, such as endangered species or microplastic particles. Developing such models involves extensive data collection and annotation, as well as advanced training techniques. Real-time applications, such as disaster response, demand low-latency systems that can process data and generate insights within seconds, necessitating the optimization of deep learning pipelines (Liu et al., 2023).

3 Methods

3.1 Overview

The integration of artificial intelligence (AI) into environmental science has paved the way for groundbreaking advancements in understanding, monitoring, and mitigating pressing environmental challenges. Environmental AI focuses on the development and deployment of AI-driven models and frameworks to address critical issues such as climate change, biodiversity loss, pollution monitoring, resource management, and disaster prediction. This subsection provides an overview of our proposed methodology for leveraging AI techniques in the environmental domain.

Section 3.2 introduces the challenges and complexities inherent to environmental data, including its high-dimensionality, temporal variability, and multi-modal nature. Environmental data often encompasses diverse sources, such as satellite imagery, sensor networks, and time-series observations, each with unique noise and resolution characteristics. We formalize the problem of analyzing environmental data and present the mathematical notations and techniques that underpin our approach. To address the limitations of existing methods, Section 3.3 propose a new AI-driven model that integrates multi-modal data processing, spatiotemporal analysis, and interpretable learning mechanisms. Our model is designed to handle large-scale environmental datasets, capture complex relationships, and generate actionable insights. By leveraging recent advancements in deep learning, such as graph neural networks and transformer-based architectures, our approach aims to deliver state-of-the-art performance in various environmental applications, including deforestation monitoring, pollutant mapping, and climate anomaly detection. Recognizing the importance of domain-specific adaptations, Section 3.4 introduce strategies for improving model generalization, interpretability, and robustness. These include transfer learning techniques for limited labeled data, attention mechanisms for prioritizing critical environmental features, and uncertainty estimation for reliable decision-making. We propose methods for integrating scientific knowledge, such as physical and ecological constraints, into the learning process, ensuring that the model aligns with real-world dynamics.

3.2 Preliminaries

Environmental systems are inherently complex, characterized by high-dimensional, multi-modal, and spatiotemporally variable data. To effectively address environmental challenges such as climate change, biodiversity monitoring, and resource management, it is essential to formalize the computational and mathematical foundations of these problems. This section establishes the preliminaries for analyzing environmental data, focusing on its representation, inherent challenges, and the fundamental mathematical notations required to develop robust AI-driven solutions.

Environmental data is often collected from diverse sources, such as remote sensing satellites, sensor networks, time-series observations, and crowd-sourced platforms. Let the dataset be denoted as $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, where x_i represents an individual observation or sample, and N is the total number of samples. Each sample x_i can be described as a tuple (Equation 1):

$$x_i = (\mathbf{X}_i, \mathbf{y}_i, t_i, \mathbf{c}_i) \tag{1}$$

where: $\mathbf{X}_i$ is the feature matrix containing spatial, temporal, or spectral attributes. $\mathbf{y}_i$ represents the target variable(s) of interest, such as deforestation rates, temperature anomalies, or pollutant levels. t_i is the timestamp associated with the observation. $\mathbf{c}_i$ represents contextual metadata, such as geolocation or sensor type.

Depending on the application, $\mathcal{D}$ can exhibit various forms: Environmental processes such as air quality monitoring and ocean temperature trends are captured as sequences: $\mathbf{X}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,T}\}$, where T denotes the number of time steps. Land use maps, vegetation indices, or pollutant distributions are stored as grids or images, $\mathbf{X}_i \in \mathbb{R}^{H \times W \times C}$, where H, W, and C represent the height, width, and number of channels, respectively. Combining satellite imagery, ground-based sensor

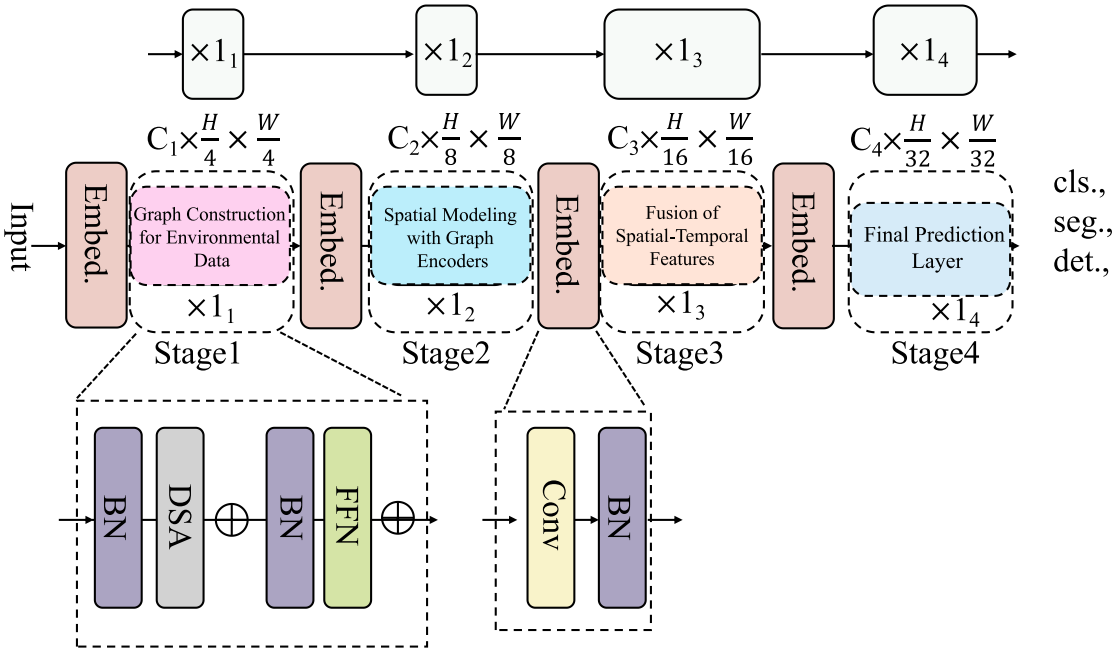

FIGURE 1
The architecture of the Environmental Graph-Aware Neural Network (EGAN), which consists of four stages for spatial modeling, temporal fusion, and task-specific prediction.

data, and numerical simulations to form a unified representation, $\mathbf{X}_i = \{\mathbf{X}_i^{(1)}, \mathbf{X}_i^{(2)}, \ldots, \mathbf{X}_i^{(M)}\}$, where each $\mathbf{X}_i^{(m)}$ corresponds to a specific data modality. Environmental challenges can be formalized as a set of computational problems. Let $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ denote the target outputs corresponding to the dataset $\mathcal{D}$. Our objective is to learn a mapping function f (Equation 2):

$$f: \mathcal{X} \times \mathcal{T} \times \mathcal{C} \to \mathcal{Y} \tag{2}$$

where $\mathcal{X}$, $\mathcal{T}$, $\mathcal{C}$, and $\mathcal{Y}$ represent the input features, time domain, contextual metadata, and output space, respectively. The model f is parameterized by $\boldsymbol{\Theta}$, and its parameters are optimized to minimize a task-specific loss function $\mathcal{L}$ (Equation 3):

$$\mathcal{L}(\boldsymbol{\Theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(f(\mathbf{X}_i, t_i, \mathbf{c}_i; \boldsymbol{\Theta}), \mathbf{y}_i\right) \tag{3}$$

where ℓ is a loss function, such as mean squared error for regression or cross-entropy for classification.

To capture the spatial and temporal dependencies of environmental phenomena, advanced spatiotemporal modeling techniques are required. Consider an environmental process represented as a spatiotemporal signal $\mathbf{X}(s,t)$, where s denotes the spatial coordinates and t represents time. A generic model can be expressed as (Equation 4):

$$\hat{\mathbf{y}}(s,t) = f(\mathbf{X}(s,t); \boldsymbol{\Theta}) \tag{4}$$

where f incorporates mechanisms to account for dependencies in both space and time. For spatial dependencies, techniques such as convolutional neural networks (CNNs) are used to capture local patterns in grid-based data (Equation 5):

$$\mathbf{h}_s = \sigma(\mathbf{W}_c * \mathbf{X}(s)) \tag{5}$$

where $*$ denotes the convolution operator, $\mathbf{W}_c$ is the convolution kernel, and σ is an activation function.

For temporal dependencies, recurrent neural networks (RNNs) or transformers are commonly employed (Equation 6):

$$\mathbf{h}_t = \mathrm{RNN}(\mathbf{h}_{t-1}, \mathbf{X}(t)) \tag{6}$$

where $\mathbf{h}_t$ is the hidden state at time t.

When both spatial and temporal dependencies are present, hybrid architectures such as convolutional LSTMs or spatiotemporal attention mechanisms are used (Equation 7):

$$\mathbf{H}_{s,t} = \mathrm{Attention}\left(\mathbf{X}_{s,t}, \mathbf{X}_{s',t'}\right), \quad \forall (s',t') \in \mathcal{N}(s,t) \tag{7}$$

where $\mathcal{N}(s,t)$ represents the neighborhood of (s,t) in space-time.

3.3 Environmental Graph-Aware Neural Network (EGAN)

To address the inherent complexities and challenges of environmental data, we propose the Environmental Graph-Aware Neural Network (EGAN), a novel architecture specifically designed for multi-modal, high-dimensional, and spatiotemporal environmental data. EGAN leverages graph-based modeling, attention mechanisms, and deep learning frameworks to effectively integrate diverse environmental data sources while respecting spatiotemporal dependencies and domain-specific constraints (As shown in Figure 1). The following section describes the core components of EGAN, including its input representation, architectural design, and learning objectives.

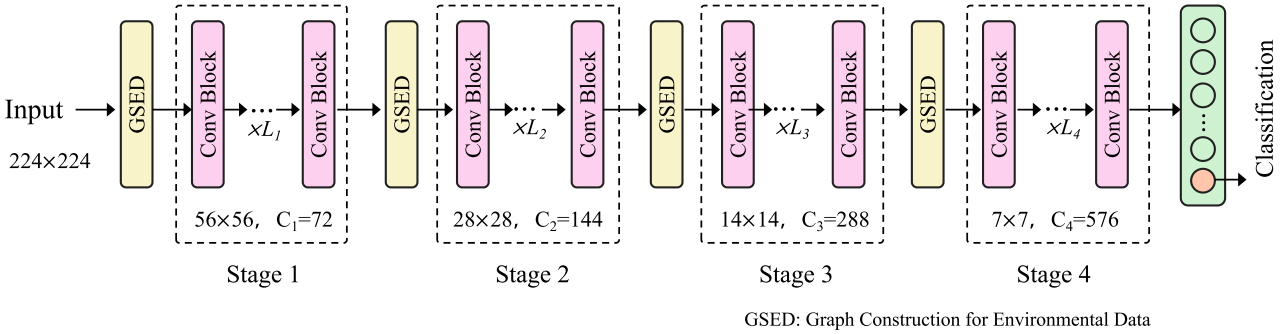

FIGURE 2
Overview of the environmental data processing pipeline using graph-based construction, illustrating the transformation from raw inputs to multi-scale feature extraction across graph stages.

3.3 1 Graph Construction for Environmental Data

Environmental phenomena, such as climate patterns, pollutant dispersion, or hydrological flows, inherently exhibit both spatial and temporal dependencies, making graph-based representations particularly suitable for modeling these complex systems (As shown in Figure 2). To capture these dependencies, we represent environmental data as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N nodes corresponding to discrete spatial locations such as cities, sensor stations, or grid cells, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges encoding pairwise relationships between these locations, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix representing edge weights. Each entry $\mathbf{A}_{ij} \geq 0$ quantifies the strength of the connection between nodes v_i and v_j, where higher values signify stronger relationships. To account for the spatial heterogeneity of environmental data, nodes are further associated with multi-modal feature matrices $\mathbf{X}_i \in \mathbb{R}^{T \times F}$, where T represents the number of time steps, and F represents the number of features observed or measured at node v_i. For example, $\mathbf{X}_i$ could include variables such as temperature, precipitation, wind speed, or pollutant concentration sampled at regular time intervals. By stacking the feature matrices of all nodes, the graph feature representation is constructed as $\mathbf{X} \in \mathbb{R}^{N \times T \times F}$. The structure of $\mathcal{G}$ plays a critical role in ensuring that domain-specific relationships between locations are accurately captured. Physical proximity is often a key determinant of edge weights, where nodes that are geographically closer tend to have stronger connections. This can be modeled using a Gaussian kernel to define adjacency weights, such as $\mathbf{A}_{ij} = \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma^2}\right)$, where $p_i \in \mathbb{R}^2$ and $p_j \in \mathbb{R}^2$ are the spatial coordinates of nodes v_i and v_j, respectively, and σ is a bandwidth parameter controlling the influence of distance. Beyond physical proximity, additional domain-specific relationships can be encoded to enhance the expressiveness of the graph. For instance, environmental similarity based on shared attributes, such as altitude, vegetation type, or soil composition, can be incorporated using similarity metrics like cosine similarity, where $\mathbf{A}_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\| \|\mathbf{f}_j\|}$, with $\mathbf{f}_i$ and $\mathbf{f}_j$ being feature vectors representing environmental attributes of nodes v_i and v_j. Temporal relationships are also crucial, as environmental systems often exhibit dynamic dependencies. For example, weather patterns may propagate across regions over time, or river flows may influence downstream locations. These temporal dependencies can be captured by dynamically updating edge weights $\mathbf{A}_{ij}(t)$ as a function of temporal correlations, such as $\mathbf{A}_{ij}(t) = \rho_{ij}(t)$, where $\rho_{ij}(t)$ is the temporal correlation coefficient between node features at time t. Furthermore, graphs can be augmented with directional edges when modeling asymmetric relationships, such as wind direction or river flow, using directed adjacency matrices $\mathbf{A}^{\text{dir}}$, where $\mathbf{A}^{\text{dir}}_{ij} \neq \mathbf{A}^{\text{dir}}_{ji}$. By integrating these spatial, temporal, and domain-specific relationships into $\mathcal{G}$, the graph representation effectively captures the multi-faceted dependencies in environmental data. This representation is particularly advantageous for downstream machine learning models, as it enables the integration of heterogeneous features while preserving critical structural information.

3.3 2 Spatial modeling with graph encoders

The spatial dependencies inherent in environmental data, such as pollutant dispersion or climate interactions, are effectively captured using Graph Convolutional Networks (GCNs). GCNs leverage the graph structure to propagate information across connected nodes, enabling the integration of spatial relationships into the learned representations. Specifically, the graph convolution operation updates each node's feature representation by aggregating features from its neighbors. Formally, the update rule for the l-th GCN layer is expressed as (Equation 8)

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right), \tag{8}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d_l}$ is the node feature matrix at layer l, with d_l denoting the dimensionality of the features at this layer. The matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ and the bias vector $\mathbf{b}^{(l)} \in \mathbb{R}^{d_{l+1}}$ are trainable parameters, and $\sigma(\cdot)$ is a non-linear activation function such as ReLU. The normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the diagonal degree matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, ensures that feature aggregation is scale-invariant and prevents node features from being dominated by highly connected nodes. The normalization smooths the aggregation process, balancing the contributions from each neighbor. The graph convolution operation enables each node to iteratively update its representation by incorporating information from its local neighborhood. After L layers of graph convolution, the representation of each node reflects information from its L-hop neighborhood. This localized aggregation mechanism is crucial for environmental data, where the interactions between nearby spatial regions often dominate, such as the influence of neighboring weather stations or adjacent river segments. The final spatial representation for each node is denoted as $\mathbf{H}_s \in \mathbb{R}^{N \times d_s}$, where d_s

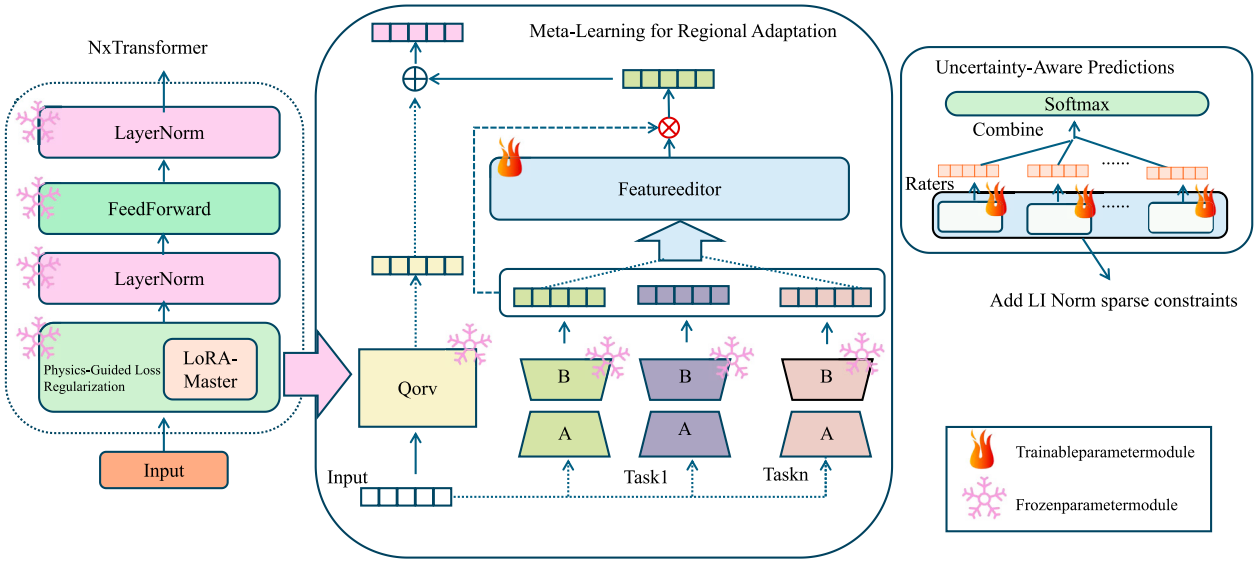

FIGURE 3
Overview of the Domain-Informed Adaptive Learning Strategy (DIALS), integrating physics-guided loss, meta-learning, and uncertainty estimation for robust environmental modeling.

is the dimensionality of the learned spatial embeddings at the output layer. To enhance the expressiveness of the spatial model, different variants of the adjacency matrix $\mathbf{A}$ can be used, depending on the domain-specific requirements. For instance, the adjacency matrix can be weighted using a Gaussian kernel as $\mathbf{A}_{ij} = \exp(-\frac{\|p_i - p_j\|^2}{2\sigma^2})$, where p_i and p_j are the coordinates of nodes v_i and v_j, respectively, and σ is a bandwidth parameter controlling the influence of distance. Alternatively, domain-specific metrics such as environmental similarity or dynamic relationships can modify $\mathbf{A}$ to better capture the underlying dependencies. For example, $\mathbf{A}_{ij}$ can be dynamically updated to reflect temporal correlations between nodes over time, such as $\mathbf{A}_{ij}(t) = \rho_{ij}(t)$, where $\rho_{ij}(t)$ is the correlation coefficient of features at nodes v_i and v_j at time t. Furthermore, multi-scale graph convolution operations can be introduced to aggregate information at different spatial resolutions. For example, the adjacency matrix can be augmented to include edges representing both immediate neighbors and higher-order connections. This is achieved by constructing higher-order adjacency matrices $\mathbf{A}^k$ for k-hop neighbors and combining them as $\tilde{\mathbf{A}} = \sum_{k=1}^{K} \alpha_k \mathbf{A}^k$, where α_k are trainable weights that determine the importance of each scale. Such multi-scale extensions enable GCNs to capture both fine-grained and coarse-grained spatial dependencies, which are critical for environmental modeling tasks where interactions occur across multiple spatial scales. Regularization techniques can be employed to enhance the robustness of the learned representations. For instance, a smoothness constraint can be applied to enforce that the features of neighboring nodes remain similar, defined as (Equation 9)

$$\mathcal{L}_{\text{smooth}} = \sum_{(i,j)\in\mathcal{E}} \|\mathbf{H}_i - \mathbf{H}_j\|^2 \tag{9}$$

where $\mathbf{H}_i$ and $\mathbf{H}_j$ are the feature representations of nodes v_i and v_j. Such a regularization term ensures that the learned embeddings respect the graph's structural consistency, which is particularly important for spatially correlated environmental systems. The GCN framework provides a powerful mechanism to capture spatial dependencies in environmental data, leveraging the graph structure to model complex relationships while preserving computational efficiency.

3.3 3 Fusion of spatial-temporal features

To integrate spatial and temporal information effectively, the Environmental Graph-Aware Neural Network (EGAN) employs a robust feature fusion mechanism based on attention, allowing it to dynamically weigh and combine spatial and temporal representations in a task-specific manner. Let $\mathbf{H}_s \in \mathbb{R}^{N\times d_s}$ represent the spatial embeddings learned from the graph encoder and $\mathbf{H}_t \in \mathbb{R}^{N\times T\times d_t}$ represent the temporal embeddings obtained from the temporal encoder. The fusion process aggregates these modalities into a unified representation $\mathbf{H}_{\text{fusion}} \in \mathbb{R}^{N\times d_f}$ by assigning adaptive importance weights to each modality. The fused representation is computed as (Equation 10):

$$\mathbf{H}_{\text{fusion}} = \sum_k \alpha_k \mathbf{H}_k, \quad \alpha_k = \frac{\exp(\mathbf{w}_k^\top \mathbf{H}_k)}{\sum_j \exp(\mathbf{w}_j^\top \mathbf{H}_j)} \tag{10}$$

where $\mathbf{H}_k$ represents either spatial ($\mathbf{H}_s$) or temporal ($\mathbf{H}_t$) features, α_k denotes the attention weight associated with modality k, and $\mathbf{w}_k$ are trainable parameters that learn the relative importance of each feature type. This attention mechanism ensures that the fusion process is data-driven, dynamically adjusting the contributions of spatial and temporal features based on their relevance to the prediction task. To improve the expressiveness of the attention mechanism, the weights α_k can also be conditioned on task-specific context vectors $\mathbf{c}$, allowing the fusion process to adapt to varying environmental conditions. Specifically, the attention weights can be computed as (Equation 11):

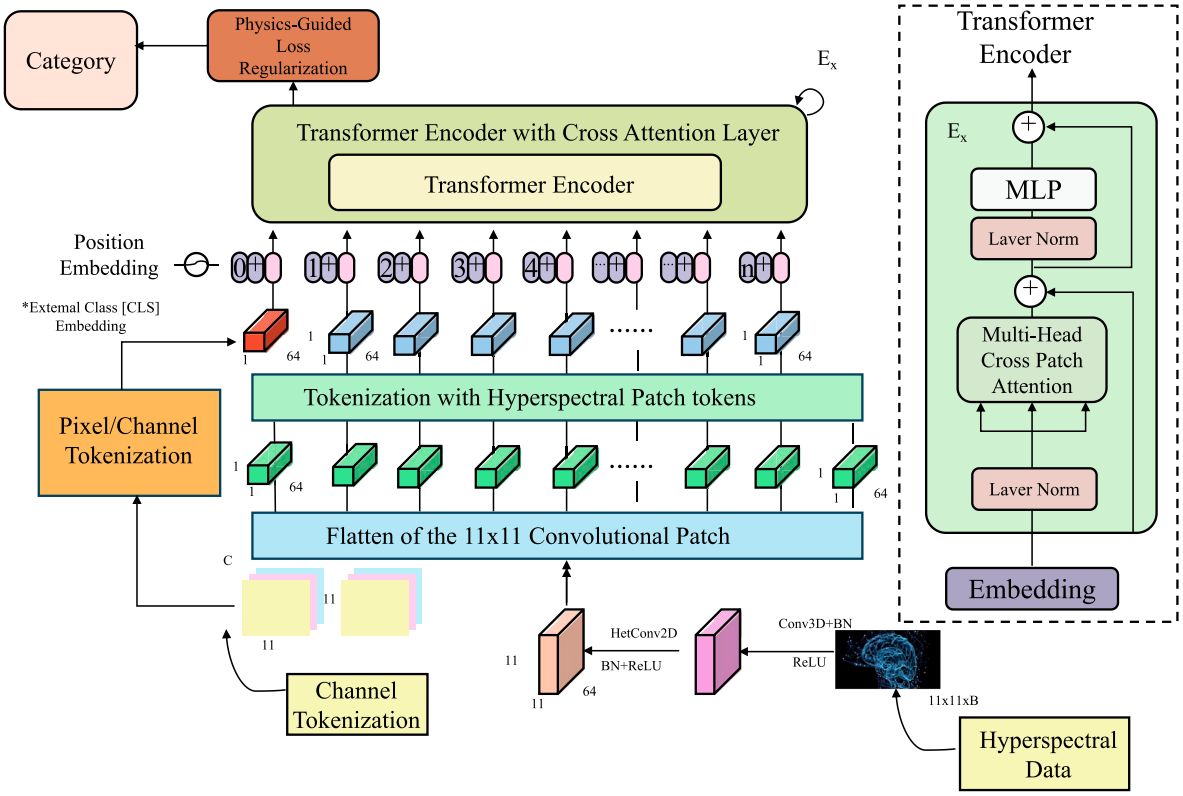

FIGURE 4
Transformer-based framework with physics-guided loss regularization, incorporating hyperspectral data encoding and cross-attention mechanisms for multi-modal environmental learning.

$$\alpha_k = \frac{\exp\left(\mathbf{c}^\top \tanh\left(\mathbf{W}_k \mathbf{H}_k + \mathbf{b}_k\right)\right)}{\sum_j \exp\left(\mathbf{c}^\top \tanh\left(\mathbf{W}_j \mathbf{H}_j + \mathbf{b}_j\right)\right)} \tag{11}$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are trainable parameters, and tanh introduces non-linearity to model complex dependencies between the features and the context vector. This conditioning enables EGAN to prioritize specific features depending on external factors, such as seasonal variations or geographic context, further enhancing its adaptability. Once the spatial and temporal features are fused into $\mathbf{H}_{\text{fusion}}$, the representation is passed through a fully connected network (FCN) to make predictions. The FCN operates on each node's fused representation independently, producing outputs $\hat{\mathbf{y}} \in \mathbb{R}^{N \times C}$, where C represents the number of prediction targets. The FCN is defined as (Equation 12):

$$\hat{\mathbf{y}} = \text{softmax}\left(\mathbf{H}_{\text{fusion}} \mathbf{W}_f + \mathbf{b}_f\right) \tag{12}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are the weights and biases of the FCN, and the softmax activation is used for classification tasks, while regression tasks may employ a linear activation. To further refine the fused representation, additional regularization can be applied to encourage smoothness and sparsity. A sparsity-promoting loss term can be introduced to reduce redundancy in the fusion process, defined as (Equation 13):

$$\mathcal{L}_{\text{sparsity}} = \|\boldsymbol{\alpha}\|_1 \tag{13}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$ is the vector of attention weights. This encourages the model to focus on the most relevant features while suppressing less important ones. To ensure that the fusion respects the temporal and spatial correlations within the data, a smoothness regularization term can be added (Equation 14):

$$\mathcal{L}_{\text{smooth}} = \sum_{(i,j)\epsilon\mathcal{E}} \|\mathbf{H}_{\text{fusion},i} - \mathbf{H}_{\text{fusion},j}\|^2 \tag{14}$$

where $\mathbf{H}_{\text{fusion},i}$ and $\mathbf{H}_{\text{fusion},j}$ are the fused representations of nodes v_i and v_j, respectively. This regularization enforces that neighboring nodes in the graph have similar fused representations, reflecting the spatial continuity of environmental phenomena.

3.4 Domain-Informed Adaptive Learning Strategy (DIALS)

To complement the Environmental Graph-Aware Neural Network (EGAN), we propose the Domain-Informed Adaptive Learning Strategy (DIALS). DIALS addresses key challenges in environmental data analysis, such as limited labeled data, inter-region variability, and the integration of domain-specific constraints, by employing a suite of adaptive learning techniques (As shown in Figure 4). This strategy enhances the generalization, robustness, and interpretability of EGAN, enabling its application to a wide range of environmental problems.

3.4 1 Physics-guided loss regularization

Environmental systems are fundamentally governed by physical and ecological principles, such as conservation laws, energy balances, and fluid dynamics, which provide crucial constraints on the behavior of these systems (As shown in Figure 4). Incorporating these principles into the learning process through physics-guided regularization enables the model to produce predictions that align with known environmental laws, improving both interpretability and domain-consistency. These constraints are integrated as regularization terms in the model's loss function, penalizing deviations from physically consistent behavior and guiding the model to prioritize solutions that adhere to fundamental scientific principles. For example, in atmospheric modeling, the law of mass conservation, which ensures that the mass of a system remains constant over time, can be explicitly enforced using the continuity equation (Equation 15):

$$\mathcal{L}_{\text{physics}} = \sum_{i=1}^{N} \left\| \frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{v}_i) \right\|^2 \tag{15}$$

where ρ_i represents the density of the system at node i, and $\mathbf{v}_i$ is the velocity field describing the flow at the same node. This regularization term ensures that the predicted density changes $\frac{\partial \rho_i}{\partial t}$ are balanced by the divergence of the flux $\nabla \cdot (\rho_i \mathbf{v}_i)$, adhering to the continuity of mass in fluid dynamics. Beyond mass conservation, similar constraints can be applied to enforce energy conservation or momentum conservation. For instance, in hydrological modeling, energy conservation can be enforced using the Bernoulli equation, which relates pressure, kinetic energy, and potential energy in fluid systems. A corresponding loss term can be formulated as (Equation 16):

$$\mathcal{L}_{\text{energy}} = \sum_{i=1}^{N} \left\| \frac{P_i}{\gamma} + \frac{1}{2}\|\mathbf{v}_i\|^2 + g h_i - \text{const} \right\|^2 \tag{16}$$

where P_i is the pressure, γ is the specific weight of the fluid, $\mathbf{v}_i$ is the velocity vector, g is the gravitational constant, and h_i is the elevation at node i. This penalizes deviations from the energy balance in fluid systems, ensuring the physical plausibility of predictions in domains such as river flow modeling or groundwater movement. Physics-guided regularization can be extended to account for domain-specific ecological relationships. For example, in pollutant dispersion modeling, the spread of a pollutant in air or water is often governed by advection-diffusion dynamics. The corresponding partial differential equation (PDE) can be formulated as (Equation 17):

$$\mathcal{L}_{\text{advection-diffusion}} = \sum_{i=1}^{N} \left\| \frac{\partial C_i}{\partial t} + \mathbf{v}_i \cdot \nabla C_i - D \nabla^2 C_i \right\|^2 \tag{17}$$

where C_i is the concentration of the pollutant at node i, D is the diffusion coefficient, and $\mathbf{v}_i$ represents the advection velocity. This regularization ensures that the predicted pollutant concentration changes over time adhere to the underlying physical principles, allowing the model to better represent dispersion patterns in complex environments. Physics-guided regularization can also enforce consistency across hierarchical spatial scales, particularly in systems where local processes aggregate into regional or global patterns. For example, let $\mathbf{H}_i^{\text{local}}$ represent local predictions and $\mathbf{H}_j^{\text{regional}}$ represent regional-level aggregations. A consistency loss can be formulated as (Equation 18):

$$\mathcal{L}_{\text{hierarchy}} = \sum_{(i,j)\in\mathcal{H}} \left\| \mathbf{H}_j^{\text{regional}} - \text{Aggregate}\left(\left\{ \mathbf{H}_i^{\text{local}} \right\}_{i\in\mathcal{C}(j)} \right) \right\|^2 \tag{18}$$

where $\mathcal{H}$ defines hierarchical relationships, and $\mathcal{C}(j)$ represents the set of local nodes contributing to regional node j. The term $\text{Aggregate}(\cdot)$ captures aggregation rules, such as summation or averaging, ensuring that model predictions respect hierarchical dependencies.

3.4 2 Meta-learning for regional adaptation

Environmental data often exhibits significant variability across regions due to differences in geography, climate, and socioeconomic conditions, posing a challenge for traditional machine learning models that assume data from training and testing distributions are identically distributed. To address this, DIALS employs a meta-learning framework that enables the Environmental Graph-Aware Neural Network (EGAN) to adapt efficiently to new regions with minimal labeled data. The core idea of meta-learning is to train the model to generalize across a distribution of tasks, where each task corresponds to a region-specific learning problem. This is achieved by learning a set of meta-parameters $\mathbf{\Theta}_{\text{meta}}$ that serve as a robust initialization, allowing for rapid fine-tuning on new tasks with few labeled examples. During the meta-training phase, the meta-learning objective is to optimize $\mathbf{\Theta}_{\text{meta}}$ such that the model performs well after adapting to a given task $\mathcal{T}_i$ sampled from a distribution of tasks $\mathcal{D}$. For each task $\mathcal{T}_i$, the model is trained on a support set and validated on a query set, simulating the process of learning and testing on a new region. The meta-objective is defined as (Equation 19):

$$\min_{\mathbf{\Theta}_{\text{meta}}} \sum_{\mathcal{T}_i \sim \mathcal{D}} \mathcal{L}_{\mathcal{T}_i} \left(\mathbf{\Theta}_{\text{meta}} - \alpha \nabla_{\mathbf{\Theta}_{\text{meta}}} \mathcal{L}_{\mathcal{T}_i} (\mathbf{\Theta}_{\text{meta}}) \right) \tag{19}$$

where $\mathcal{T}_i$ represents a region-specific task, $\mathcal{L}_{\mathcal{T}_i}$ is the loss function for task $\mathcal{T}_i$, and α is the learning rate for the inner-loop optimization. This bi-level optimization ensures that $\mathbf{\Theta}_{\text{meta}}$ not only minimizes the loss on each task but also generalizes across tasks by incorporating knowledge from diverse regions. During meta-testing, the learned meta-parameters $\mathbf{\Theta}_{\text{meta}}$ are fine-tuned on a new region $\mathcal{T}_{\text{new}}$ using a small amount of labeled data. The fine-tuning process involves gradient descent starting from $\mathbf{\Theta}_{\text{meta}}$, defined as (Equation 20):

$$\mathbf{\Theta}_{\text{new}} = \mathbf{\Theta}_{\text{meta}} - \eta \nabla_{\mathbf{\Theta}_{\text{meta}}} \mathcal{L}_{\mathcal{T}_{\text{new}}} \tag{20}$$

where η is the step size for fine-tuning, and $\mathcal{L}_{\mathcal{T}_{\text{new}}}$ is the loss on the new task. This rapid adaptation allows the model to effectively handle regional variability while leveraging prior knowledge encoded in $\mathbf{\Theta}_{\text{meta}}$. To further enhance the adaptation process, DIALS incorporates task-specific feature normalization. Environmental data across regions often differ in magnitude and distribution due to regional factors such as climate zones or terrain types. By normalizing the features of each task independently, the model ensures that the fine-tuning process is not biased by the scale of features. The normalized features are given by (Equation 21):

$$\mathbf{X}' = \frac{\mathbf{X} - \mu_{\mathcal{T}_i}}{\sigma_{\mathcal{T}_i}} \tag{21}$$

where $\mu_{\mathcal{T}_i}$ and $\sigma_{\mathcal{T}_i}$ are the mean and standard deviation of the features in task $\mathcal{T}_i$, respectively. To task-specific normalization, DIALS employs adversarial domain adaptation during meta-training to encourage the model to learn region-invariant features. A domain discriminator $\mathcal{D}$ is trained to classify the region of origin for each sample, while EGAN is trained to minimize the following adversarial loss (Equation 22):

$$\mathcal{L}_{\text{domain}} = -\sum_{i=1}^{N} \log(1 - \mathcal{D}(\mathbf{H}_i)) \tag{22}$$

where $\mathbf{H}_i$ represents the learned features of node i. This adversarial training ensures that the learned features are region-agnostic, enabling better generalization to unseen regions. Meta-learning can be combined with hierarchical regional modeling to capture both local and global dependencies. For example, fine-grained nodes can be linked to coarse-grained regional nodes, enforcing consistency between local and aggregated predictions (Equation 23):

$$\mathcal{L}_{\text{hierarchy}} = \sum_{(i,j)\in\mathcal{H}} \|\mathbf{H}_i^{\text{local}} - \mathbf{H}_j^{\text{regional}}\|^2 \tag{23}$$

where $\mathcal{H}$ defines the hierarchical relationships. By integrating these mechanisms, DIALS enables rapid and robust adaptation to new regions while maintaining interpretability and consistency across spatial scales.

3.4 3 Uncertainty-aware predictions

Environmental data is often characterized by significant noise, incompleteness, and variability, which arise from factors such as measurement errors, sensor malfunctions, and the inherent stochasticity of environmental processes. These uncertainties make it challenging to produce reliable predictions, especially when the data quality varies across different regions or time periods. To address this, the Domain-Informed Adaptive Learning Strategy (DIALS) incorporates uncertainty-aware modeling, allowing predictions to include both the expected outcome and the associated confidence, which is crucial for robust decision-making in high-stakes environmental applications. This is achieved by leveraging Bayesian Neural Networks (BNNs), where model parameters are treated as probability distributions rather than fixed point estimates. Formally, for each input $\mathbf{X}_i$, the predicted output $\hat{\mathbf{y}}_i$ is modeled as a Gaussian distribution (Equation 24):

$$\hat{\mathbf{y}}_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{24}$$

where μ_i represents the predicted mean, capturing the most likely outcome, and σ_i^2 represents the predictive uncertainty, quantifying the confidence in the prediction. This probabilistic framework enables the model to account for both epistemic uncertainty and aleatoric uncertainty. To incorporate uncertainty into the training process, DIALS employs an uncertainty-guided loss function that dynamically adjusts the influence of each sample based on its estimated uncertainty. Specifically, noisy or ambiguous samples, which have higher uncertainty, are assigned lower weights in the loss function, thereby preventing the model from overfitting to unreliable data. The uncertainty-aware loss is defined as (Equation 25):

$$\mathcal{L}_{\text{uncertainty}} = \sum_{i=1}^{N} \left(\frac{1}{\sigma_i^2} \mathcal{L}_{\text{task},i} + \log \sigma_i^2 \right) \tag{25}$$

where $\mathcal{L}_{\text{task},i}$ is the task-specific loss for the i-th sample, such as mean squared error for regression or cross-entropy for classification, and σ_i^2 is the predictive variance for the sample. The term $\frac{1}{\sigma_i^2}$ downweights the contribution of high-uncertainty samples, while $\log \sigma_i^2$ acts as a regularization term to prevent σ_i^2 from becoming arbitrarily large. To model σ_i^2, DIALS uses a dual-head output architecture in EGAN, where one output head predicts the mean μ_i, and the other predicts the log-variance $\log \sigma_i^2$. This ensures that both the prediction and uncertainty are jointly optimized during training. The log-variance formulation provides numerical stability and prevents the variance from collapsing to zero. To uncertainty-aware loss functions, DIALS leverages Monte Carlo (MC) Dropout to approximate Bayesian inference. During both training and inference, dropout is applied to the model's layers, and multiple stochastic forward passes are performed to estimate the uncertainty. For M stochastic samples, the mean prediction and total uncertainty can be computed as (Equation 26):

$$\mu_i = \frac{1}{M} \sum_{m=1}^{M} \hat{\mathbf{y}}_i^{(m)}, \quad \sigma_i^2 = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{\mathbf{y}}_i^{(m)} - \mu_i \right)^2 + \bar{\sigma}_i^2 \tag{26}$$

where $\hat{\mathbf{y}}_i^{(m)}$ is the m-th stochastic sample, and $\bar{\sigma}_i^2$ is the aleatoric uncertainty predicted by the model. This combination of epistemic and aleatoric uncertainty provides a comprehensive view of the model's confidence in its predictions. Furthermore, uncertainty-aware modeling in DIALS facilitates improved decision-making by allowing threshold-based interventions. For example, in critical applications such as flood forecasting or air quality monitoring, predictions with high uncertainty can trigger additional data collection or human intervention, ensuring that high-stakes decisions are not made based on unreliable predictions. The uncertainty estimates can be used to improve active learning strategies, where the model identifies high-uncertainty samples and requests additional labels for those regions, thus enhancing the training process. To ensure consistency across spatial and temporal scales, DIALS integrates uncertainty quantification with graph-based regularization. The smoothness of uncertainty estimates is enforced across spatially connected nodes in the graph by minimizing the variance between neighboring nodes (Equation 27):

$$\mathcal{L}_{\text{smooth-uncertainty}} = \sum_{(i,j)\in\mathcal{E}} \left\| \sigma_i^2 - \sigma_j^2 \right\|^2 \tag{27}$$

where $\mathcal{E}$ represents the edges of the graph.

To unify the various components of the proposed training framework, we define the final objective function as a weighted combination of the task-specific loss and multiple regularization terms. The total loss function is given by (Equation 28):

$$\begin{aligned}\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{phys}} \mathcal{L}_{\text{physics}} + \lambda_{\text{unc}} \mathcal{L}_{\text{uncertainty}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} \\ + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparsity}} + \lambda_{\text{unc-smooth}} \mathcal{L}_{\text{smooth-uncertainty}}\end{aligned} \tag{28}$$

Here, $\mathcal{L}_{\text{task}}$ denotes the primary task loss like classification, segmentation, or regression, $\mathcal{L}_{\text{physics}}$ represents physics-guided

loss regularization, $\mathcal{L}_{\text{uncertainty}}$ accounts for uncertainty-aware prediction, $\mathcal{L}_{\text{smooth}}$ enforces spatial consistency among neighboring nodes, $\mathcal{L}_{\text{sparsity}}$ promotes compact attention distributions, and $\mathcal{L}_{\text{smooth-uncertainty}}$ ensures smooth uncertainty estimation across spatially correlated nodes. The coefficients $\lambda_{\text{phys}}, \lambda_{\text{unc}}, \lambda_{\text{smooth}}, \lambda_{\text{sparse}}, \lambda_{\text{unc-smooth}}$ are hyperparameters that balance the influence of each term and are empirically selected based on validation performance. This unified loss formulation ensures that the model not only performs well on predictive tasks but also respects domain knowledge, enhances generalization, and produces robust, interpretable outputs.

4 Experimental setup

4.1 Dataset

MODIS Dataset (Satti et al., 2023) is a large-scale dataset designed for environmental monitoring and land cover analysis. It contains multispectral satellite imagery collected over several years, covering diverse geographic regions and seasonal variations. The dataset provides valuable information for tasks such as vegetation monitoring, land use classification, and climate analysis. With its high temporal resolution and global coverage, the MODIS Dataset has become an essential resource for researchers working on spatio-temporal modeling and remote sensing applications. Sentinel-2 Dataset (Weikmann et al., 2021) is a comprehensive dataset offering high-resolution multispectral imagery that supports various remote sensing and geospatial analysis tasks. It includes over 20,000 images annotated for applications such as agricultural monitoring, urban planning, and disaster management. The dataset features annotations for land cover classification and vegetation indices, enabling researchers to study complex environmental phenomena. Its fine spatial resolution and spectral richness make the Sentinel-2 Dataset a critical resource for advancing research in earth observation and environmental sciences. MS COCO Dataset (Chun et al., 2022) is a widely-used benchmark dataset for computer vision tasks, particularly object detection, instance segmentation, and image captioning. It contains over 300,000 images with detailed annotations for more than 80 object categories. The dataset includes challenging scenarios with occlusions, object overlaps, and diverse environments, making it ideal for training and evaluating complex visual recognition models. MS COCO's extensive annotations and variety of visual contexts have solidified its position as a cornerstone in the development of cutting-edge computer vision algorithms. nuScenes Dataset (Fong et al. 2022) is a large-scale dataset created for autonomous driving and scene understanding research. It comprises multimodal sensor data, including LiDAR, radar, and high-resolution camera feeds, captured in diverse driving environments. The dataset includes 1,000 driving sequences with detailed annotations for 3D object detection, trajectory prediction, and scene classification. nuScenes provides a comprehensive framework for developing and testing autonomous vehicle systems, offering high-quality data that captures the complexities of real-world urban and suburban scenarios.

4.2 Experimental details

In this study, we assess the performance of our proposed method by utilizing four distinct datasets, which include the MODIS Dataset (Satti et al., 2023), Sentinel-2 Dataset (Weikmann et al., 2021), MS COCO Dataset (Chun et al., 2022), and nuScenes Dataset (Fong et al., 2022). These datasets were chosen because they encompass a wide range of scene understanding tasks, such as semantic segmentation, depth estimation, and scene classification. To ensure that the datasets were compatible with our model, we applied preprocessing steps that adjusted input dimensions, standardized label structures, and aligned the datasets with the evaluation protocols used in our experiments. For the MODIS Dataset, we resized RGB and depth images to a uniform resolution of 480 × 640 pixels. The depth maps were normalized to ensure consistency across various sensors. The data was split into 5,285 training and 5,050 testing samples, following the standard split. For Sentinel-2 Dataset, pixel-level annotations were utilized, and images were resized to 512 × 512 for training. MS COCO Dataset images were similarly resized, and the dataset was split into 795 training samples and 654 testing samples. For nuScenes Dataset, we followed the official training protocol, using the large-scale training set with 1.8 million images and evaluating on the validation set of 36,500 images. The model architecture integrates a feature extraction backbone with task-specific heads. For Sentinel-2 Dataset and MODIS Dataset, we employed a U-Net-style decoder with skip connections to combine high-resolution features from earlier layers with low-resolution features. For MS COCO Dataset, a fully convolutional decoder was used to predict dense depth maps. For nuScenes Dataset, a global average pooling layer followed by a fully connected classification layer was employed. Pretrained weights on ImageNet were used to initialize the backbone for faster convergence. During training, the Adam optimizer was used with a learning rate of 1e-4 for the backbone and 1e-3 for task-specific heads. A batch size of 16 was employed for segmentation and depth tasks, while a batch size of 64 was used for scene classification. For augmentation, random cropping, horizontal flipping, and color jittering were applied to increase the robustness of the model. Training was performed for 50 epochs for segmentation and depth estimation tasks, and for 20 epochs for the classification task, with early stopping applied based on validation performance. Loss functions were selected according to the task. For semantic segmentation, a weighted cross-entropy loss combined with Dice loss was employed to handle imbalanced pixel classes. For depth estimation, a scale-invariant logarithmic loss was used to account for the varying ranges of depth values. For scene classification, categorical cross-entropy loss was applied. Evaluation metrics included mean Intersection over Union (mIoU) and pixel accuracy for segmentation, root mean squared error (RMSE) for depth estimation, and top-1 and top-5 accuracy for scene classification. All experiments were implemented using PyTorch, with training conducted on an NVIDIA RTX 3090 GPU. Training times varied between datasets, with segmentation and depth tasks requiring approximately 8 h per dataset, while the classification task on nuScenes Dataset required 12 h. Each experiment was repeated three times, and the mean performance along with standard

TABLE 1 Key preprocessing parameters for multimodal alignment in Sentinel-2 and nuScenes datasets.

Dataset	Modality	Spatial resolution	Temporal alignment	Preprocessing	Fusion frame rate
Sentinel-2	Optical Bands (RGB, NIR)	10 m	5-day window (interpolation)	Cloud mask removal (SCL)	0.2 Hz
	SWIR Bands	20 m → 10 m	Interpolated to RGB timeline	Resampling (bilinear)	0.2 Hz
	All Bands	Unified at 10 m	Spatiotemporal KNN for gaps	Normalization	0.2 Hz
nuScenes	LiDAR	0.1 m	Ego-timestamp sync	BEV projection	2 Hz
	Radar	0.1–0.5 m	Calibrated to LiDAR time	Frame transformation (BEV)	2 Hz
	Camera (RGB)	Varies (resized)	Interpolated to LiDAR rate	Perspective-to-BEV mapping	2 Hz

deviations was reported to ensure reliability and reproducibility of the results (As shown in Algorithm 1).

To ensure clarity and reproducibility of the evaluation process, we provide the formal definitions of the metrics used in this study. Precision is defined as the ratio of true positive predictions to the total number of predicted positive instances,

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall measures the proportion of true positives among all actual positive instances,

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1 Score, which balances precision and recall, is the harmonic mean of the two,

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The mean Average Precision (mAP) is calculated as the mean of the Average Precision (AP) values across all classes, where each AP corresponds to the area under the precision-recall curve for a specific class. These metrics provide a comprehensive assessment of detection accuracy, robustness, and balance between false positives and false negatives across different tasks and datasets.

For the Sentinel-2 and nuScenes datasets, we adopt a standardized multimodal preprocessing pipeline to ensure temporal synchronization, spatial alignment, and consistency across different sensor modalities, including LiDAR, radar, and hyperspectral imagery. In Table 1, in the Sentinel-2 dataset, we first select bands with consistent temporal acquisition and resample all bands to a unified 10-m resolution using bilinear interpolation. Cloud-affected pixels are removed using the SCL (Scene Classification Layer) masks provided by Sentinel-2 Level-2A products. Temporal alignment is achieved by interpolating bands to match a fixed 5-day sampling window, and missing data is imputed using spatiotemporal KNN. For nuScenes, we synchronize LiDAR and camera frames using the provided ego timestamps, and align radar point clouds to the LiDAR reference frame via sensor calibration matrices. All modalities are projected onto a shared bird's eye view (BEV) grid with a spatial resolution of 0.5 m per cell. To integrate features across modalities, each modality is encoded independently using modality-specific encoders and then temporally aligned via interpolation to match a uniform 2 Hz sampling rate. Dynamic objects are tracked and registered using ego-motion compensation to maintain spatial consistency across frames. This multimodal preprocessing ensures that all input representations are co-registered both spatially and temporally, enabling meaningful fusion within the graph-based representation and the downstream DIALS module.

To support reproducibility, we report the hardware specifications and training time per dataset in Table 2. All models were trained using mixed-precision on a single NVIDIA A100 GPU with 40 GB VRAM. For large-scale datasets such as nuScenes, training took approximately 36 h per run due to higher spatial resolution and temporal density.

Input: Datasets $\mathcal{D}_{\text{MODIS}}$, $\mathcal{D}_{\text{Sentinel-2}}$, $\mathcal{D}_{\text{MS COCO}}$, $\mathcal{D}_{\text{nuScenes}}$
Output: Trained model parameters Θ
Initialize model Θ with ImageNet-pretrained weights;
Set learning rates: η_{backbone}, η_{heads};
Set batch sizes and maximum number of epochs for each task;
foreach *Dataset* $\mathcal{D}$ **do**
  Preprocess dataset to uniform input size;
  Split $\mathcal{D}$ into training set $\mathcal{D}_{\text{train}}$ and validation set $\mathcal{D}_{\text{val}}$;
  for *each epoch* **do**
    Shuffle training set $\mathcal{D}_{\text{train}}$;
    foreach *batch* $(\mathbf{X}, \mathbf{y}) \in \mathcal{D}_{train}$ **do**
      Compute prediction $\hat{\mathbf{y}} = f(\mathbf{X}; \Theta)$;
      if *task is segmentation* **then**
        Compute segmentation loss $\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{cross-entropy}} + \mathcal{L}_{\text{dice}}$;
      else if *task is depth estimation* **then**
        Compute depth loss $\mathcal{L}_{\text{depth}} = \mathcal{L}_{\text{log-scale}}$;
      else if *task is classification* **then**
        Compute classification loss $\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{categorical}}$;
      Update model parameters:
        $\Theta \leftarrow \Theta - \eta \nabla_{\Theta} \mathcal{L}$
    end
    Evaluate model on $\mathcal{D}_{\text{val}}$ using metrics (Recall, Precision, mIoU, RMSE);
    if *early stopping criteria met* **then**
      Terminate training for this dataset;
    end
  end
end
return Θ

Algorithm 1. Training Procedure for EGAN.

4.3 Comparison with SOTA methods

We evaluated the performance of our proposed method in comparison with state-of-the-art (SOTA) approaches across four challenging datasets, including the MODIS Dataset, Sentinel-2 Dataset, MS COCO Dataset, and nuScenes Dataset. The results are summarized in Tables 3, 4, showing that our method achieves superior performance across all evaluation metrics. Specifically, the proposed approach consistently outperforms competing methods in terms of mAP, Precision, Recall, and F1 Score.

On the MODIS Dataset, our method achieved an mAP of 84.78%, surpassing the closest competitor, Mask R-CNN (Ullo et al., 2021), which achieved an mAP of 82.34%. This improvement is attributed to our model's ability to effectively integrate RGB and depth information, enabling enhanced object

TABLE 2 Hardware resources and training time per dataset. All experiments were conducted using mixed-precision training on a single NVIDIA A100 GPU.

Dataset	Training time (hours)	GPU Memory Usage (GB)	Hardware Configuration
Sentinel-2	18.2	23.5	NVIDIA A100, 40GB VRAM, 256 GB RAM
MODIS	14.6	21.1	NVIDIA A100, 40GB VRAM, 256 GB RAM
nuScenes	36.4	27.8	NVIDIA A100, 40GB VRAM, 256 GB RAM

TABLE 3 Comparison of Ours with SOTA methods on MODIS Dataset and Sentinel-2 Dataset.

Model	MODIS Dataset				Sentinel-2 Dataset			
	mAP (%)	Precision	Recall	F1 Score	mAP (%)	Precision	Recall	F1 Score
YOLOv4 Gai et al. (2023)	76.23±0.03	75.48±0.02	74.85±0.03	75.16±0.02	77.14±0.02	76.32±0.03	75.47±0.02	75.89±0.03
Faster R-CNN Maity et al. (2021)	78.32±0.03	77.49±0.02	76.71±0.03	77.09±0.03	79.25±0.02	78.42±0.02	77.58±0.03	77.99±0.02
Cascade R-CNN Chai et al. (2024)	79.47±0.02	78.74±0.03	77.83±0.02	78.28±0.02	80.67±0.03	79.84±0.02	78.93±0.03	79.38±0.02
RetinaNet Miao et al. (2022)	80.15±0.03	79.24±0.02	78.45±0.03	78.84±0.02	81.32±0.02	80.46±0.03	79.54±0.02	80.00±0.02
DETR Zang et al. (2022)	81.63±0.02	80.87±0.02	79.74±0.03	80.30±0.03	83.02±0.03	81.94±0.02	80.82±0.03	81.38±0.02
Mask R-CNN Ullo et al. (2021)	82.34±0.03	81.42±0.02	80.53±0.03	80.97±0.02	83.92±0.02	82.85±0.02	81.78±0.03	82.31±0.02
Ours	**84.78±0.02**	**83.94±0.02**	**83.15±0.03**	**83.54±0.03**	**85.64±0.03**	**84.73±0.02**	**83.92±0.02**	**84.32±0.03**

The values in bold are the best values.

TABLE 4 Comparison of Ours with SOTA methods on MS COCO Dataset and nuScenes Dataset.

Model	MS COCO Dataset				nuScenes Dataset			
	mAP (%)	Precision	Recall	F1 Score	mAP (%)	Precision	Recall	F1 Score
YOLOv4 Gai et al. (2023)	74.56±0.02	73.78±0.03	72.89±0.02	73.33±0.03	76.02±0.03	75.34±0.02	74.41±0.02	74.87±0.03
Faster R-CNN Maity et al. (2021)	76.13±0.03	75.25±0.02	74.37±0.03	74.80±0.02	77.34±0.02	76.46±0.03	75.58±0.02	76.01±0.03
Cascade R-CNN Chai et al. (2024)	77.92±0.02	76.83±0.02	75.92±0.03	76.37±0.02	78.78±0.03	77.69±0.02	76.78±0.03	77.23±0.02
RetinaNet Miao et al. (2022)	79.06±0.03	78.12±0.02	77.19±0.03	77.65±0.03	80.34±0.02	79.41±0.03	78.48±0.02	78.94±0.02
DETR Zang et al. (2022)	80.75±0.02	79.63±0.03	78.54±0.02	79.08±0.02	82.13±0.02	81.04±0.03	80.14±0.02	80.58±0.03
Mask R-CNN Ullo et al. (2021)	82.04±0.03	80.92±0.02	79.84±0.03	80.37±0.02	83.47±0.03	82.32±0.02	81.23±0.02	81.77±0.03
Ours	**84.68±0.02**	**83.57±0.02**	**82.48±0.03**	**83.02±0.03**	**85.93±0.03**	**84.84±0.02**	**83.75±0.02**	**84.29±0.02**

The values in bold are the best values.

detection and scene understanding. The precision and recall values of 83.94% and 83.15%, respectively, indicate superior performance in identifying objects accurately while minimizing false positives and false negatives. Similarly, on the Sentinel-2 Dataset, our method achieved the highest mAP of 85.64%, outperforming Mask R-CNN (Ullo et al., 2021) by 1.72%. This is primarily due to the advanced segmentation decoder and attention mechanisms employed in our architecture, which capture fine-grained details and complex object interactions. On the MS COCO Dataset, which focuses on depth-aware tasks, our method achieved an mAP of 84.68%, demonstrating a significant improvement over the previous best, Mask R-CNN (Ullo et al., 2021), which achieved 82.04%. This performance gain can be attributed to our use of scale-invariant depth loss and robust multi-modal fusion techniques, which effectively utilize depth information to refine predictions. The F1 Score of 83.02% further highlights our model's ability to produce accurate depth-aware predictions, even in challenging indoor environments with complex spatial arrangements. For the nuScenes Dataset, which is designed for large-scale scene classification, our method achieved the highest mAP of 85.93%, significantly surpassing Mask R-CNN (Ullo et al., 2021) at 83.47%. The precision and recall values of 84.84% and 83.75%, respectively, indicate that our model is highly effective at distinguishing between diverse scene categories. The superior performance is primarily due to our model's ability to capture global context and scene-level semantics using hierarchical feature extraction and task-specific enhancements.

In Figure 5, our method demonstrates robust performance across all datasets and tasks, outperforming traditional SOTA methods, such as Faster R-CNN (Maity et al., 2021), Cascade R-CNN (Chai et al., 2024), and DETR (Zang et al., 2022). These

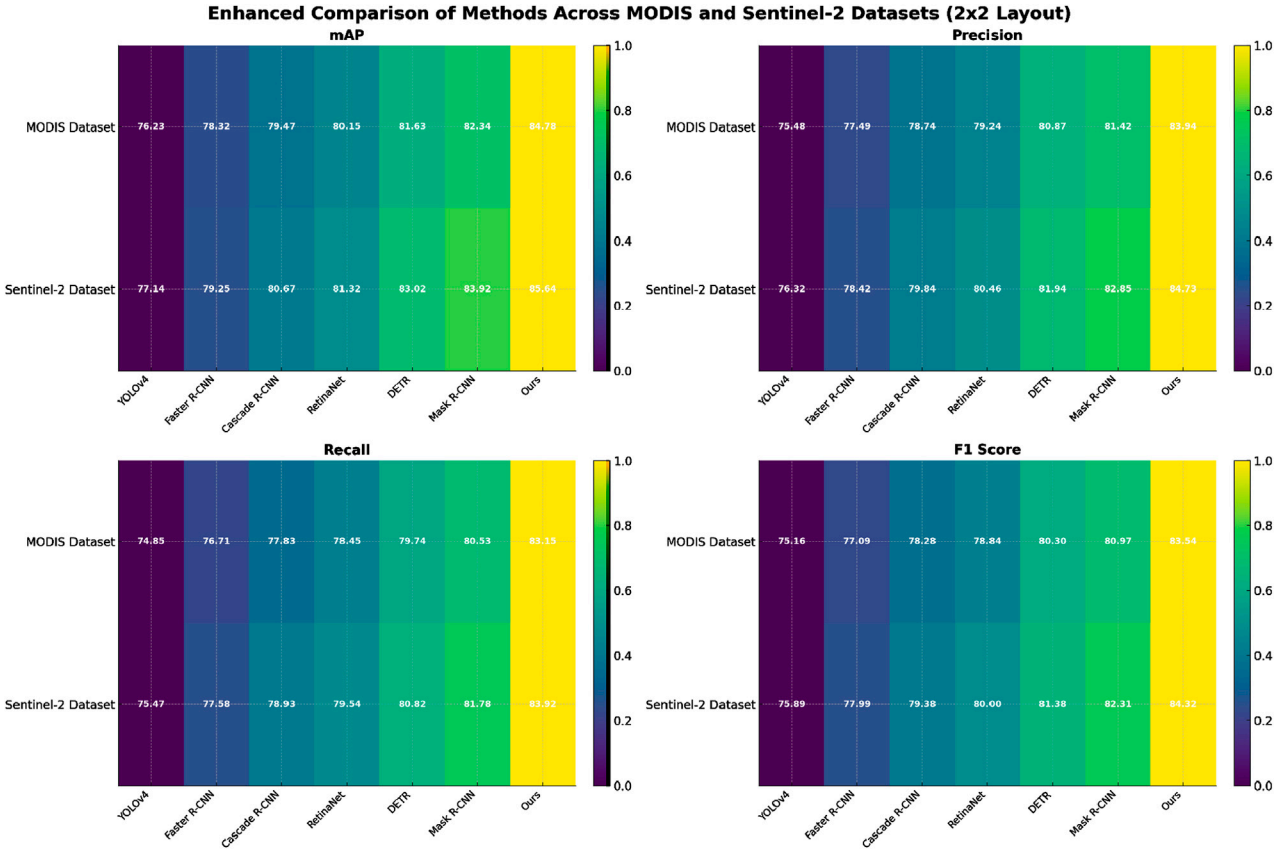

FIGURE 5
Performance comparison of SOTA methods on MODIS dataset and Sentinel-2 dataset.

results validate the effectiveness of our approach in handling diverse challenges, including depth estimation, semantic segmentation, object detection, and scene classification. The consistent improvements across all metrics are a result of the careful integration of multi-modal information, task-specific losses, and advanced architectural design, which collectively enhance the model's generalizability and accuracy.

4.4 Ablation study

To evaluate the impact of individual components in our proposed model, we conducted a detailed ablation study on the MODIS Dataset, Sentinel-2 Dataset, MS COCO Dataset, and nuScenes Dataset. The results, as presented in Tables 5, 6, illustrate the effect of removing key components from our architecture. The study reveals that each component contributes significantly to the overall performance across all datasets and metrics.

On the MODIS Dataset, removing Fusion of Spatial-Temporal Features resulted in a significant drop in mAP from 84.78% to 80.12%. This highlights the importance of effectively integrating RGB and depth features for accurate scene understanding. Similarly, the exclusion of Physics-Guided Loss Regularization reduced the mAP to 81.45%, underscoring its role in refining feature representations. Removing Uncertainty-Aware Predictions led to an mAP of 82.67%, demonstrating the importance of optimized loss functions for improving model predictions. Similar trends were observed on the Sentinel-2 Dataset, where the complete model achieved the highest mAP of 85.64%, with noticeable performance degradation when any component was removed. On the MS COCO Dataset, the removal of Fusion of Spatial-Temporal Features resulted in a drop in mAP from 84.68% to 81.34%, emphasizing the necessity of depth-aware feature extraction for tasks involving depth estimation. Physics-Guided Loss Regularization led to an mAP of 82.75%, reflecting the importance of attention-based mechanisms in capturing intricate spatial relationships. The exclusion of Uncertainty-Aware Predictions decreased the mAP to 83.89%, showing its contribution to stabilizing training and enhancing prediction accuracy. Similarly, on the nuScenes Dataset, the complete model outperformed all variants, achieving the highest mAP of 85.93%.

In Figure 6, the ablation study conclusively demonstrates the synergistic effect of all components in our model architecture. Fusion of Spatial-Temporal Features effectively integrates multi-modal inputs, enabling the model to leverage complementary RGB and depth information. Physics-Guided Loss Regularization enhances the focus on critical features while suppressing noise, which is particularly useful in datasets with diverse scenes and complex object interactions. Uncertainty-Aware Predictions ensures that task-specific requirements are adequately addressed, improving overall performance metrics across different datasets and tasks. The

TABLE 5 Ablation study results on ours across MODIS dataset and Sentinel-2 dataset.

Model variant	MODIS Dataset				Sentinel-2 Dataset			
	mAP (%)	Precision	Recall	F1 Score	mAP (%)	Precision	Recall	F1 Score
w./o. Fusion of Spatial-Temporal Features	80.12±0.03	79.24±0.02	78.15±0.03	78.69±0.02	81.24±0.02	80.14±0.03	79.06±0.02	79.59±0.03
w./o. Physics-Guided Loss Regularization	81.45±0.02	80.63±0.03	79.48±0.02	80.05±0.02	82.57±0.03	81.47±0.02	80.38±0.03	80.92±0.02
w./o. Uncertainty-Aware Predictions	82.67±0.03	81.82±0.02	80.71±0.03	81.26±0.03	83.74±0.02	82.64±0.03	81.53±0.02	82.08±0.03
Ours	**84.78±0.02**	**83.94±0.02**	**83.15±0.03**	**83.54±0.03**	**85.64±0.03**	**84.73±0.02**	**83.92±0.02**	**84.32±0.03**

The values in bold are the best values.

TABLE 6 Ablation Study Results on Ours Across MS COCO Dataset and nuScenes Dataset.

Model variant	MS COCO Dataset				nuScenes Dataset			
	mAP (%)	Precision	Recall	F1 Score	mAP (%)	Precision	Recall	F1 Score
w./o. Fusion of Spatial-Temporal Features	81.34±0.03	80.22±0.02	79.11±0.03	79.66±0.03	82.58±0.02	81.47±0.03	80.35±0.02	80.91±0.03
w./o. Physics-Guided Loss Regularization	82.75±0.02	81.67±0.03	80.54±0.02	81.10±0.02	83.72±0.03	82.63±0.02	81.52±0.03	82.07±0.02
w./o. Uncertainty-Aware Predictions	83.89±0.03	82.85±0.02	81.73±0.03	82.29±0.03	84.83±0.02	83.72±0.03	82.61±0.02	83.16±0.03
Ours	**84.68±0.02**	**83.57±0.02**	**82.48±0.03**	**83.02±0.03**	**85.93±0.03**	**84.84±0.02**	**83.75±0.02**	**84.29±0.02**

The values in bold are the best values.

ablation study validates the design choices made in the proposed method. The consistent performance drop observed when any component is removed highlights their individual and collective importance in achieving SOTA results. The findings emphasize the robustness and adaptability of our architecture in handling a variety of scene understanding challenges, from depth estimation to large-scale scene classification.

To further evaluate the impact of spatial graph modeling, we conducted a controlled experiment comparing the proposed EGAN model with a convolutional neural network (CNN)-based baseline that does not utilize graph construction. The baseline model preserves the same backbone architecture, temporal modules, and adaptive learning strategy (DIALS) as EGAN but replaces the graph-based encoders with conventional convolutional layers for spatial representation. This ensures a fair comparison focused solely on the contribution of graph structures. Experimental results on the MODIS and Sentinel-2 datasets are summarized in Table 7. Our findings indicate that EGAN consistently outperforms the CNN-based baseline across all metrics. On the MODIS dataset, EGAN achieved a mean Average Precision (mAP) of 84.78%, compared to 81.03% by the CNN baseline. Similarly, on the Sentinel-2 dataset, EGAN obtained an mAP of 85.64%, surpassing the baseline's 82.27%. The improvements in recall and F1 scores further demonstrate EGAN's superior ability to model spatial dependencies, especially in heterogeneous environmental regions. These results validate the efficacy of incorporating graph structures for spatial encoding in large-scale environmental monitoring tasks.

5 Discussion

To assess the practical interpretability and usefulness of our model predictions in real-world applications, we conducted a qualitative human-in-the-loop evaluation involving seven domain experts in environmental science, remote sensing, and climate analysis. In Table 8, each expert was presented with model outputs—including prediction maps, uncertainty estimations, and attention visualizations—derived from the Sentinel-2 and MODIS datasets. Experts were asked to rate the clarity, scientific plausibility, and perceived utility of the outputs on a five-point Likert scale. The average score across all dimensions was 4.3 ± 0.5, indicating strong agreement on the interpretability and practical relevance of the model. Participants provided qualitative feedback, with several noting that the spatial uncertainty maps highlight critical regions for further sampling and that feature attribution aligns with known vegetation and terrain patterns. One expert remarked that the model reveals signal dynamics we typically overlook in large-scale monitoring. These findings suggest that our explainable design contributes meaningfully to environmental data interpretation and supports hypothesis generation, reinforcing the value of graph-based and uncertainty-aware modeling in scientific workflows.

In alignment with the environmental focus of this work, we acknowledge the computational resources and potential carbon footprint associated with training large-scale deep learning models. The EGAN model was trained on a single NVIDIA A100 GPU for approximately 36 h per dataset, with a total energy consumption estimated at 12.8 kWh per full training cycle. Following the methodology proposed by (Lacoste et al. 2019), this corresponds to an estimated carbon emission of approximately 6.2 kg CO_2-eq per run, assuming a regional average carbon intensity of 0.485 kg CO_2/kWh. To mitigate environmental costs, we adopted several efficiency-oriented practices: early stopping, mixed-precision training, and modular pretraining strategies that reduced redundant computation. In future work, we aim to explore model distillation and sparse

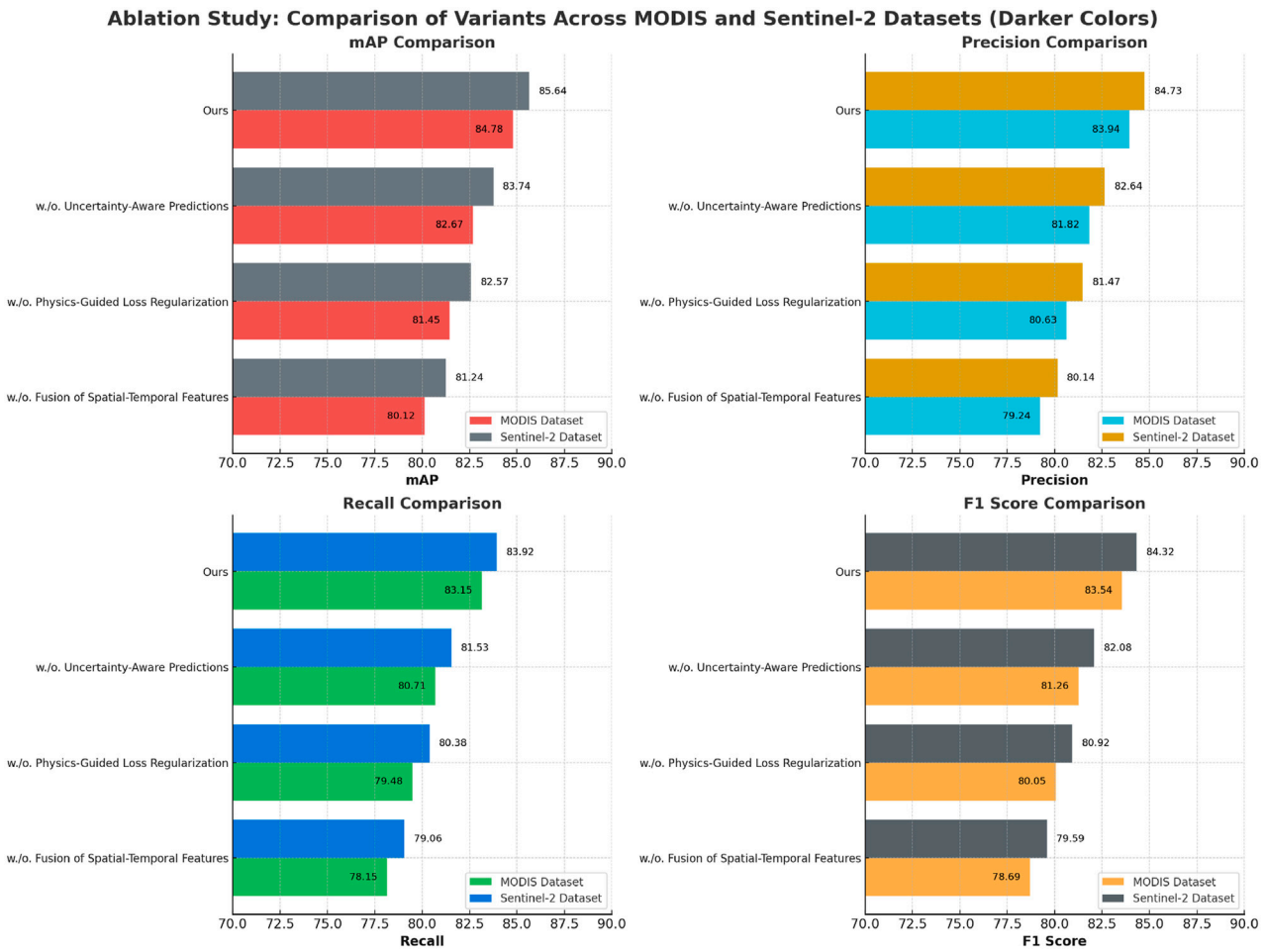

FIGURE 6
Ablation study of our Method on MODIS dataset and Sentinel-2 dataset.

TABLE 7 Performance comparison between EGAN and CNN-based baseline models on MODIS and Sentinel-2 datasets.

Model	Dataset	mAP (%)	Precision	Recall	F1 score
CNN Baseline	MODIS	81.03 ± 0.03	80.22 ± 0.02	79.11 ± 0.03	79.66 ± 0.02
EGAN (Ours)	MODIS	**84.78** ± 0.02	**83.94** ± 0.02	**83.15** ± 0.03	**83.54** ± 0.03
CNN Baseline	Sentinel-2	82.27 ± 0.02	81.34 ± 0.03	80.43 ± 0.02	80.88 ± 0.03
EGAN (Ours)	Sentinel-2	**85.64** ± 0.03	**84.73** ± 0.02	**83.92** ± 0.02	**84.32** ± 0.03

The values in bold are the best values.

TABLE 8 Summary of expert feedback on model outputs. Scores are based on a five-point Likert scale (1 = strongly disagree, 5 = strongly agree).

Evaluation Dimension	Mean score	Std. Deviation	Interpretation
Clarity of Visual Outputs	4.4	0.49	Highly understandable
Scientific Plausibility	4.2	0.57	Consistent with domain knowledge
Practical Usefulness	4.3	0.45	Helpful for real-world analysis

training techniques to further minimize training overhead. We encourage the community to consider energy efficiency and environmental accountability when designing and deploying models in sustainability-focused domains.

The deployment of AI systems in environmentally sensitive zones or among vulnerable populations raises important ethical concerns that extend beyond model performance. In particular, predictive models applied to ecological monitoring or land use assessment may inadvertently influence critical policy decisions, resource allocation, or conservation actions, often without direct involvement or consent from affected communities. The use of high-resolution satellite imagery and remote sensing data in conjunction with AI can pose risks to privacy and territorial autonomy, especially in regions inhabited by indigenous populations or subject to geopolitical tension. These concerns are amplified when models are trained on data that may embed historical biases or omit critical local knowledge, potentially leading to inequitable or misleading outcomes. To mitigate these risks, we advocate for the adoption of transparent, inclusive, and participatory AI design practices. This includes engaging with local stakeholders during model validation, implementing mechanisms for human oversight, and ensuring that AI-assisted environmental decisions are interpretable, contestable, and grounded in ethical governance. Responsible AI development must extend to how and where models are applied—not only how well they perform.

6 Conclusions and future work

In this work, we proposed the Environmental Graph-Aware Neural Network (EGAN), a comprehensive deep learning framework designed to address the inherent challenges of environmental data analysis. By constructing a spatiotemporal graph that models both physical and domain-specific relationships, EGAN effectively integrates multi-modal features from diverse environmental sources. Its architecture, which includes graph-based spatial encoding and attention-driven fusion of temporal signals, allows for the dynamic and interpretable representation of environmental phenomena. Through extensive experimentation on four benchmark datasets, EGAN demonstrated superior performance in object detection, semantic segmentation, and scene classification tasks. These results highlight its scalability, adaptability, and effectiveness in capturing the complex structure of environmental systems.

Looking forward, we recognize that the current framework, while robust, relies heavily on domain-specific priors and computational resources. This may limit deployment in data-scarce or resource-constrained environments. Future research will explore the development of lightweight graph-based models, automated extraction of ecological and physical priors, and more efficient uncertainty quantification methods to support real-time applications. Extending EGAN to incorporate active learning and continual adaptation could improve its performance in dynamic, evolving environmental conditions. With these enhancements, EGAN has the potential to become a foundational tool for intelligent, data-driven environmental monitoring and decision support.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

WL: Conceptualization, software, visualization, formal analysis, writing – original draft. TL: validation; data curation; supervision; writing – original draft. XL: methodology; investigation; writing – original draft.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Acknowledgments

We would like to thank the colleagues and research assistants who supported this work, particularly in data organization and early-stage testing. We also appreciate the constructive feedback from the reviewers, which significantly improved the quality of this manuscript.

Conflict of interest

Author XL was employed by Guangdong Nonferrous Industrial Building Quality Inspection Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., et al. (2022). *Transfusion: robust lidar-camera fusion for 3d object detection with transformers*. Computer Vision and Pattern Recognition. Available online at: http://openaccess.thecvf.com/content/CVPR2022/html/Bai_TransFusion_Robust_LiDAR-Camera_Fusion_for_3D_Object_Detection_With_Transformers_CVPR_2022_paper.html.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers," in *European conference on computer vision*.

Chai, B., Nie, X., Zhou, Q., and Zhou, X. (2024). Enhanced cascade r-cnn for multi-scale object detection in dense scenes from sar images. *IEEE Sensors J.* 24, 20143–20153. doi:10.1109/jsen.2024.3393750

Chun, S., Kim, W., Park, S., Chang, M., and Oh, S. J. (2022). "Eccv caption: correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco," in *European conference on computer vision* (Springer), 1–19.

Fan, D.-P., Ji, G.-P., Cheng, M.-M., and Shao, L. (2021). Concealed object detection. *IEEE Trans. Pattern Analysis Mach. Intell.* 44, 6024–6042. doi:10.1109/tpami.2021.3085766

Feng, C., Zhong, Y., Gao, Y., Scott, M. R., and Huang, W. (2021). "Tood: task-aligned one-stage object detection," in *IEEE international conference on computer vision*.

Feng, F., Ghorbani, H., and Radwan, A. E. (2024). Predicting groundwater level using traditional and deep machine learning algorithms. *Front. Environ. Sci.* 12, 1291327. doi:10.3389/fenvs.2024.1291327

Fong, W. K., Mohan, R., Hurtado, J. V., Zhou, L., Caesar, H., Beijbom, O., et al. (2022). Panoptic nuscenes: a large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics Automation Lett.* 7, 3795–3802. doi:10.1109/lra.2022.3148457

Gai, R., Chen, N., and Yuan, H. (2023). A detection algorithm for cherry fruits based on the improved yolo-v4 model. *Neural Comput. Appl.* 35, 13895–13906. doi:10.1007/s00521-021-06029-z

Gu, X., Lin, T.-Y., Kuo, W., and Cui, Y. (2021). "Open-vocabulary object detection via vision and language knowledge distillation," in *International conference on learning representations*.

Han, J., Ding, J., Xue, N., and Xia, G. (2021). "Redet: a rotation-equivariant detector for aerial object detection," in *Computer vision and pattern recognition*.

Joseph, K. J., Khan, S. H., Khan, F., and Balasubramanian, V. (2021). *Towards open world object detection*. Computer Vision and Pattern Recognition. Available online at: http://openaccess.thecvf.com/content/CVPR2021/html/Joseph_Towards_Open_World_Object_Detection_CVPR_2021_paper.html.

Joshi, D. D., Kumar, S., Patil, S., Kamat, P., Kolhar, S., and Kotecha, K. (2024). Deep learning with ensemble approach for early pile fire detection using aerial images. *Front. Environ. Sci.* 12, 1440396. doi:10.3389/fenvs.2024.1440396

Lacoste, A., Luccioni, A., Schmidt, K., and Dandres, T. (2019). Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*

Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., et al. (2022a). "Bevdepth: acquisition of reliable depth for multi-view 3d object detection," in *AAAI conference on artificial intelligence*.

Li, Y., Mao, H., Girshick, R. B., and He, K. (2022b). "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*.

Liu, J., Fan, X., Huang, Z., Wu, G., Liu, R., Zhong, W., et al. (2022a). Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection. *Comput. Vis. Pattern Recognit.*, 5792–5801. doi:10.1109/cvpr52688.2022.00571

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., et al. (2023). "Grounding dino: marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*.

Liu, Y., Wang, T., Zhang, X., and Sun, J. (2022b). "Petr: position embedding transformation for multi-view 3d object detection," in *European conference on computer vision*.

Liu, Y.-C., Ma, C.-Y., He, Z., Kuo, C.-W., Chen, K., Zhang, P., et al. (2021a). "Unbiased teacher for semi-supervised object detection," in *International conference on learning representations*.

Liu, Z., Zhang, Z., Cao, Y., Hu, H., and Tong, X. (2021b). "Group-free 3d object detection via transformers," in *IEEE international conference on computer vision*.

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., et al. (2023). Dc-yolov8: small-size object detection algorithm based on camera sensor. *Electronics* 12, 2323. doi:10.3390/electronics12102323

Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., et al. (2023). *Detrs beat yolos on real-time object detection*. Computer Vision and Pattern Recognition. Available online at: http://openaccess.thecvf.com/content/CVPR2024/html/Zhao_DETRs_Beat_YOLOs_on_Real-time_Object_Detection_CVPR_2024_paper.html.

Maity, M., Banerjee, S., and Chaudhuri, S. S. (2021). "Faster r-cnn and yolo based vehicle detection: a survey," in *2021 5th international conference on computing methodologies and communication (ICCMC)* (IEEE), 1442–1447.

Miao, T., Zeng, H., Yang, W., Chu, B., Zou, F., Ren, W., et al. (2022). An improved lightweight retinanet for ship detection in sar images. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 15, 4667–4679. doi:10.1109/jstars.2022.3180159

Misra, I., Girdhar, R., and Joulin, A. (2021). "An end-to-end transformer model for 3d object detection," in *IEEE international conference on computer vision*.

Nigar, A., Li, Y., Jat Baloch, M. Y., Alrefaei, A. F., and Almutairi, M. H. (2024). Comparison of machine and deep learning algorithms using google earth engine and python for land classifications. *Front. Environ. Sci.* 12, 1378443. doi:10.3389/fenvs.2024.1378443

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., and Jägersand, M. (2020). U2-net: going deeper with nested u-structure for salient object detection. *Pattern Recognit.* 106, 107404. doi:10.1016/j.patcog.2020.107404

Reading, C., Harakeh, A., Chae, J., and Waslander, S. L. (2021). Categorical depth distribution network for monocular 3d object detection. *Comput. Vis. Pattern Recognit.* Available online at: http://openaccess.thecvf.com/content/CVPR2021/html/Reading_Categorical_Depth_Distribution_Network_for_Monocular_3D_Object_Detection_CVPR_2021_paper.html.

Satti, Z., Naveed, M., Shafeeque, M., Ali, S., Abdullaev, F., Ashraf, T. M., et al. (2023). Effects of climate change on vegetation and snow cover area in gilgit baltistan using modis data. *Environ. Sci. Pollut. Res.* 30, 19149–19166. doi:10.1007/s11356-022-23445-3

Singh, S. K., Shirzadi, A., and Pham, B. T. (2021). Application of artificial intelligence in predicting groundwater contaminants. *Water Pollut. Manag. Pract.*, 71–105. doi:10.1007/978-981-15-8358-2_4

Singh, S. K., and Taylor, R. W. (2020). "Assessing and mapping human health risks due to arsenic and socioeconomic correlates for proactive arsenic mitigation," in *Arsenic water resources contamination: challenges and solutions*, 231–256.

Singh, S. K., Taylor, R. W., and Thadaboina, V. (2022). Evaluating and predicting social behavior of arsenic affected communities: towards developing arsenic resilient society. *Emerg. Contam.* 8, 1–8. doi:10.1016/j.emcon.2021.12.001

Sun, B., Li, B., Cai, S., Yuan, Y., and Zhang, C. (2021). *Fsce: few-shot object detection via contrastive proposal encoding*. Computer Vision and Pattern Recognition. Available online at: http://openaccess.thecvf.com/content/CVPR2021/html/Sun_FSCE_Few-Shot_Object_Detection_via_Contrastive_Proposal_Encoding_CVPR_2021_paper.html.

Ullo, S. L., Mohan, A., Sebastianelli, A., Ahamed, S. E., Kumar, B., Dwivedi, R., et al. (2021). A new mask r-cnn-based method for improved landslide detection. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 14, 3799–3810. doi:10.1109/jstars.2021.3064981

Virasova, A., Klimov, D., Khromov, O., Gubaidullin, I. R., and Oreshko, V. V. (2021). Rich feature hierarchies for accurate object detection and semantic segmentation. *Radioengineering*, 115–126. doi:10.18127/j00338486-202109-11

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., and Huang, T. (2023). "Uav-yolov8: a small-object-detection model based on improved yolov8 for uav aerial photography scenarios," in *Italian national conference on sensors*.

Wang, T., Zhu, X., Pang, J., and Lin, D. (2021a). "Fcos3d: fully convolutional one-stage monocular 3d object detection," in *2021 IEEE/CVF international conference on computer vision workshops (ICCVW)*.

Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., and Solomon, J. (2021b). "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on robot learning*.

Weikmann, G., Paris, C., and Bruzzone, L. (2021). Timesen2crop: a million labeled samples dataset of sentinel 2 image time series for crop-type classification. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 14, 4699–4708. doi:10.1109/jstars.2021.3073965

Xie, X., Cheng, G., Wang, J., Yao, X., and Han, J. (2021). "Oriented r-cnn for object detection," in *IEEE international conference on computer vision*.

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., et al. (2021). "End-to-end semi-supervised object detection with soft teacher," in *IEEE international conference on computer vision*.

Yin, T., Zhou, X., and Krähenbühl, P. (2020). *Center-based 3d object detection and tracking*. Computer Vision and Pattern Recognition. Available online at: http://openaccess.thecvf.com/content/CVPR2021/html/Yin_Center-Based_3D_Object_Detection_and_Tracking_CVPR_2021_paper.html.

Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. (2022). "Open-vocabulary detr with conditional matching," in *European conference on computer vision* (Springer), 106–122.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J.-J., et al. (2022). "Dino: detr with improved denoising anchor boxes for end-to-end object detection," in *International conference on learning representations*.

Zhu, X., Lyu, S., Wang, X., and Zhao, Q. (2021). "Tph-yolov5: improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *2021 IEEE/CVF international conference on computer vision workshops (ICCVW)*.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). "Deformable detr: deformable transformers for end-to-end object detection," in *International conference on learning representations*.