

Article

Analysis of Modern Landscape Architecture Evolution Using Image-Based Computational Methods

Junlei Zhang and Chi Gao *

Department of Landscape Architecture, School of Horticulture and Forestry Sciences, Huazhong Agricultural University, Wuhan 430070, China; juley Zhang@163.com

* Correspondence: gaochi307@163.com or gaochi@mail.hzau.edu.cn

Abstract

We present a novel deep learning framework for high-resolution semantic segmentation, designed to interpret complex visual environments such as cities, rural areas, and natural landscapes. Our method integrates conic geometric embeddings, which is a mathematical approach for capturing spatial relationships, with belief-aware learning, a strategy that adapts model predictions when input data are uncertain or change over time. A multi-scale refinement process further improves boundary accuracy and detail preservation. The proposed model, built on a hybrid Vision Transformer (ViT) backbone and trained end-to-end using adaptive optimization, is evaluated on four benchmark datasets including EDEN, OpenEarthMap, Cityscapes, and iSAID. It achieves 88.94% Accuracy and R^2 of 0.859 on EDEN, while surpassing 85.3% Accuracy on Cityscapes. Ablation studies demonstrate that removing Conic Output Embeddings causes drops in Accuracy of up to 2.77% and increases in RMSE, emphasizing their contribution to frequency-aware generalization across diverse conditions. On OpenEarthMap, our model achieves a mean IoU of 73.21%, outperforming previous baselines by 2.9%, and on iSAID, it reaches 80.75% mIoU with improved boundary adherence. Beyond technical performance, the framework enables practical applications such as automated landscape analysis, urban growth monitoring, and sustainable environmental planning. Its consistent results across three independent runs demonstrate both robustness and reproducibility, offering a reliable tool for large-scale geospatial and environmental modeling.



Academic Editor: Gintautas Dzemyda

Received: 20 July 2025

Revised: 22 August 2025

Accepted: 29 August 2025

Published: 1 September 2025

Citation: Zhang, J.; Gao, C. Analysis of Modern Landscape Architecture Evolution Using Image-Based Computational Methods. *Mathematics* **2025**, *13*, 2806. <https://doi.org/10.3390/math13172806>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: structured prediction; conic geometric embedding; computational geometry; adaptive optimization; belief-aware learning; high-dimensional visual data

MSC: 68U01; 68U03; 68U05; 68U07; 68U10; 68U15; 68U35

1. Introduction

The investigation of the evolution of landscape architecture is of significant importance in understanding how human interaction with natural and constructed environments has changed over time [1]. Not only does this research provide insights into cultural and ecological shifts, but it also facilitates the evaluation of sustainable practices across both historical and contemporary landscapes. Furthermore, by leveraging image analysis techniques, researchers are able to quantify visual patterns and morphological changes that are otherwise difficult to capture through traditional textual descriptions or manual methods [2]. This analytical capability significantly enhances our ability to monitor spatial configurations and aesthetic principles across different periods and regions, enabling a more

precise interpretation of design intentions and environmental responses. The integration of computational tools within this domain not only enriches traditional historiographical approaches, but also offers predictive insights that can inform and guide future design paradigms in landscape architecture [3].

Initial efforts to digitize and interpret landscape imagery centered on formalizing design structures and spatial relationships through codified visual categories. Researchers built structured representations of garden typologies, geometric motifs, and spatial arrangements, using these frameworks to document and analyze stylistic developments over time [4]. While these methods were highly effective for archival classification and thematic analysis, they often relied heavily on extensive expert input and struggled to accommodate the visual variability present in images across diverse regions and historical periods [5]. As landscape forms evolved and image dataset became increasingly complex, early structured systems lacked the flexibility to adapt, particularly when faced with incomplete visual cues or stylistic ambiguity. Consequently, these limitations highlighted a clear need for dynamic, data-driven analytical tools capable of accommodating visual variability while preserving interpretability and consistency in landscape analysis [6].

With the proliferation of digital archives and visual documentation, researchers began employing algorithm approaches that could detect visual regularities and visual cues directly from large image collections. These models evaluated recurring features such as vegetation density, material textures, and color palettes to support comparative analysis across stylistic traditions, geographic contexts, and historical time-frames [7]. By statistically analyzing these visual elements, these systems begin to uncover deeper relationships in design trends and environmental responses. Although these techniques significantly expanded the analytical scope of landscape studies, they often depended on curated datasets and had difficulty interpreting images with inconsistent resolution, occlusions, or visual noise [8]. Moreover, these models were not always equipped to account for the contextual meaning embedded in spatial arrangements or symbolic features within a landscape [9]. Despite these challenges, this line of research significantly improved automation and scalability in historical landscape interpretation, laying the groundwork for more nuanced semantic exploration [10].

Recent advancements have introduced a new wave of visual analysis systems capable of interpreting complex design elements and spatial hierarchies with minimal human intervention [11]. By leveraging layered visual encodings learned from large-scale image datasets, these systems can distinguish subtle differences in layout, style, and function across a wide range of landscape [12]. Techniques that integrate multiple image sources, such as aerial photography, archival photos, and satellite imagery, now enable a multi-perspective analysis of site evolution capturing changes in spatial configuration and visual character overtime [13]. These models can identify transitions in land use, detect transformations in spatial organization, and infer underlying design principles as they evolve over time [14]. However, the internal reasoning processes of these models often remain opaque, making it difficult to trace how conclusions are derived or to validate the reliability of their interpretations [15]. Ensuring alignment between model outputs and domain-specific knowledge remains an ongoing challenge, particularly when visual cues are subtle, ambiguous or culturally contextual. Nevertheless, the ability to scale such analysis and reveal hidden spatial narratives has fundamentally redefined our approach to the historical study of landscape architecture.

Given the limitations of symbolic rigidity, data dependency, and challenges in interpretability, we propose a novel framework that synthesizes the strengths of existing approaches while addressing their shortcomings. Our method integrates the domain-specific knowledge from symbolic AI with the adaptability of machine learning and the

representational depth of deep learning. By incorporating a multi-modal fusion mechanism, the model aims to bridge the semantic gap between human-centric design principles and machine-driven image interpretation, enabling more transparent and contextually aware analyses of landscape transformations over time. In addition, the framework supports iterative learning through expert feedback, improving both the reliability of the model and the trust of the user. This integrated approach not only overcomes the methodological bottlenecks of earlier techniques but also opens up new possibilities for scalable, interpretable, and data-efficient exploration of the temporal and spatial evolution of landscape architecture. While recent advances in historical landscape evolution analysis and AI-driven interpretation have improved predictive modeling, they face key limitations: reliance on Euclidean embeddings that fail to capture complex spatial geometry, limited handling of uncertainty in latent states, and heuristic rather than principled multi-scale feature integration.

Our framework addresses these gaps through three innovations:

1. **Conic Geometric Embeddings:** It is a novel conic manifold representation that captures both local curvature and global topology, enhancing spatial reasoning and preserving context. Unlike CNN or transformer embeddings, it inherently models hierarchical spatial dependencies without heavy reliance on post hoc positional encoding.
2. **Belief-Aware Learning:** It introduces probabilistic belief distributions over latent structures, allowing predictions to reflect multiple plausible configurations. This improves interpretability and directly addresses the deterministic limitations of existing predictive models.
3. **Multi-Scale Refinement with Structured Alignment:** It implements a mathematically guided coarse-to-fine fusion within the conic embedding space, ensuring semantic consistency across scales. This principled approach is more robust to scale variations and spatial noise than ad hoc fusion methods.

No prior work in the cited literature combines (i) a geometrically grounded embedding space, (ii) a belief-aware probabilistic prediction mechanism, and (iii) a mathematically principled multi-scale refinement into a unified architecture. The integration of these elements creates a scalable, interpretable, and data-efficient approach that significantly outperforms existing baselines in both accuracy and explainability.

A summary of our contributions is provided below:

- The integration of a multi-modal fusion mechanism enhances feature representation and allows for more nuanced understanding of both visual and semantic data.
- The framework demonstrates high adaptability across diverse landscape styles and temporal contexts, ensuring strong generalization and operational efficiency in multi-scenario applications.
- Experimental results reveal superior accuracy in classifying landscape elements and tracking their temporal evolution, outperforming baseline models across key evaluation metrics.

2. Related Work

2.1. Historical Landscape Evolution Analysis

The study of landscape architecture evolution through image analysis has been substantially enriched by the integration of historical cartography, aerial photography, and contemporary remote sensing technologies [16]. These tools have allowed scholars to chronicle long-term transformations in both urban and rural environments, revealing trends in land use, vegetation coverage, and infrastructure development. Historical maps and aerial photographs are frequently digitized and georeferenced to allow spatial comparison

over time, allowing researchers to quantify rates and patterns of landscape change with greater precision [17]. For example, in rural European contexts, such reconstructions over the past 160 years have revealed significant shifts from agrarian to suburban land use, alongside increasing habitat fragmentation and biodiversity loss. Geographic information systems (GISs) have further enhanced the analysis by providing a platform for spatial overlay, raster-based classification, and morphometric assessment, thereby enriching the understanding of landscape transformation processes [18].

When coupled with satellite imagery and LiDAR data, researchers can model topographic and vegetative changes with unprecedented precision, enabling the detection of micro-scale changes such as erosion, sedimentation, and vegetation succession. In studies of ancient urban settlements, remote sensing technologies such as multispectral and hyperspectral imaging have offered valuable insights into settlement boundaries, hydrological systems, and landform alterations that are otherwise undetectable through conventional methods [19]. Recent advances also include UAV (drone)-based photogrammetry, which provides high-resolution 3D mapping capabilities and facilitates frequent temporal monitoring of the landscape. These techniques are increasingly augmented by machine learning algorithms that support automated feature detection, classification, and temporal prediction, improving the efficiency and accuracy of landscape analysis [20]. Moreover, the fusion of sensor data from multiple modalities, optical, radar, and thermal, enables multilayered analysis that enhances interpretive accuracy. These integrated methodologies are essential not only for informing restoration efforts and guiding sustainable land use policies, but also for preserving cultural landscapes, identifying heritage values, monitoring the impact of climate change, and supporting adaptive strategies amid rapid urbanization and environmental transformation.

2.2. Predictive Modeling in Landscape Dynamics

Predictive modeling techniques have emerged as vital tools in analyzing and forecasting the spatial and temporal dynamics of landscapes [21]. Among the most widely adopted techniques are multi-criterion evaluation (MCE), cellular automata (CA), and Markov chain models, which, when combined into the MCE-CA Markov framework, offer robust simulation of future land use patterns by integrating environmental, socio-economic, and spatial parameters to simulate complex interactions influencing landscape change [22]. For instance, studies applying this model to urban peripheries have efficiently identified hotspots of urban sprawl and ecological degradation under varying development scenarios. In addition, these models also compute landscape metrics such as patch density, edge contrast, and spatial autocorrelation to evaluate the ecological impacts of projected land use changes with greater precision and spatial insight [23]. Recent advancements have enabled the integration of additional dynamic factors, such as climate variability, transportation network evolution, and demographic shifts, allowing for more nuanced and context-sensitive forecasting. Furthermore, temporal calibration techniques using sliding windows and rolling baselines have enhanced model adaptability to evolving trends and reduced the risk of temporal bias in long-term simulations [24].

Moreover, coupling predictive models with agent-based simulations can capture human decision-making behaviors and institutional dynamics, enhancing the realism and responsiveness of planning scenarios. Such analyses are indispensable for urban planners and environmental managers seeking to evaluate the long-term consequences of zoning regulations, infrastructure expansion, and conservation strategies [25]. Hybrid approaches that combine machine learning methods, such as random forests or deep neural networks, with traditional CA-Markov frameworks have shown increased predictive precision, particularly in heterogeneous landscapes. The integration of these models

with real-time geospatial data sources such as satellite imagery, UAV-derived datasets, and sensor networks allows dynamic simulation environments that can be continuously updated with incoming data. Cloud-based processing and AI-driven model calibration further increase the scalability, responsiveness, and accuracy of such systems. By utilizing statistical rigor with spatial representation, predictive modeling plays a crucial role in fostering proactive, data-informed decision-making in landscape planning and ecological restoration initiatives and sustainable development.

2.3. Artificial Intelligence in Landscape Analysis

The application of artificial intelligence (AI) in landscape analysis is reshaping how spatial and visual data are interpreted, enabling the automation of complex analytical processes and enhancing creative workflows [26]. Deep learning models, particularly convolutional neural networks (CNNs), have been extensively employed to automate image classification tasks such as identifying vegetation types, surface water bodies, built-up areas, and soil conditions from satellite imagery and UAV drone data. These models offered high classification accuracy and generalizability when we trained them on large and well-annotated datasets across diverse geographic regions [27]. In the realm of generative design, we used AI-generated content (AIGC) and diffusion models to synthesize landscape visuals, proposed alternative site layouts, and simulated seasonal or climate-induced changes in site appearance. Tools such as panoptic segmentation algorithms can extract detailed spatial structures from urban scenes, aiding in virtual reconstruction and scenario planning [28]. Furthermore, advanced semantic segmentation methods now support multi-scale feature extraction, enabling finer categorization of land cover elements even in heterogeneous terrains. AI-powered predictive analytics can also forecast future landscape states under different planning strategies or climate change scenarios, offering planners critical foresight [29]. Moreover, AI-assisted design platforms are now capable of integrating ecological rules, cultural parameters, and user preferences to generate responsive design options that align with sustainability goals. Some systems even employ reinforcement learning to iteratively optimize designs based on environmental feedback loops [30].

Recent advancements include the use of transformer-based vision models and self-supervised learning techniques that reduce reliance on labeled data and improve generalizability across regions. Integration with geographic knowledge graphs and spatial ontologies enhances contextual reasoning and decision support. These innovations are not only increasing efficiency but also enabling a shift toward adaptive, resilient landscape design approaches that can respond to rapidly evolving environmental and societal pressures, thereby fostering more sustainable and inclusive development paradigms.

2.4. Hybrid Vision Transformer-Based Methods and Our Contributions

Recent research in remote sensing and geospatial analysis has increasingly used hybrid Vision Transformer (ViT)-based architectures, which combine convolutional neural networks (CNNs) for local feature extraction with transformer encoders for capturing long-range dependencies [31–34]. Models such as SegFormer [33] and Swin-Unet [34] use lightweight CNN stems to preserve fine spatial details, while hierarchical transformer blocks model global context. This hybrid design addresses the limitations of pure transformers in modeling fine structures and of pure CNNs in capturing global relationships, making them effective for semantic segmentation tasks.

Our method follows these principles but adds several improvements tailored for complex and noisy landscape images. We use a hybrid convolutional stem with adaptive receptive fields to better preserve multi-scale textures, which is important for distinguishing subtle land cover classes. A multi-scale fusion module and class-specific attention blocks in

the decoder improve boundary refinement and robustness to lighting and seasonal variations. We also add a domain adaptation component with dataset-specific normalization, which is not present in most existing hybrid ViT approaches and improves performance across datasets. To ensure scalability, we use an optimized poly learning rate schedule and mixed precision training, keeping inference fast even for large, high-resolution datasets. Finally, by conditioning temporal embeddings with semantic class context, our framework extends hybrid ViTs to handle temporally-aware landscape change detection, where most existing methods are still limited.

3. Methods

3.1. Overview

In this section, we present a comprehensive overview of the proposed methodology, referred to as the Landscape framework, specifically designed to address structured learning problems encountered during complex decision and representational context. Our framework integrates modeling innovations and strategic algorithmic solutions to address fundamental challenges such as high-dimensional structural complexity, sequential ambiguity, and the presence of latent semantic relationships. Landscape, thus, aims to advance scalable and interpretable analysis in dynamic, data-rich environments. The Section 3.2 defines the core problem and sets up the notational framework for our approach. We formalize the target learning task as a structured prediction problem characterized by ambiguous observations and high-order dependencies across spatial and temporal dimensions. The primary objective is to learn a function or policy that maps partially observable states or high-dimensional inputs into target outputs, while preserving structural consistency embedded in the data.

To this end, we introduce the formal hypothesis space, the relevant latent state space where applicable and the structural assumptions embedded in the data. We also frame a generalized decision process abstraction such as a Partially Observable Markov Decision Process (POMDP) when applicable, which is crucial to understanding the underlying complexity of decision-aware representation learning. Notably, we refrain from decomposing the problem in a simplistic enumeration, instead presenting a unified mathematical formulation through a series of optimization-based definitions and operator-centric constructions. The Structural Model, described in Section 3.3 and termed the Structured Cone Machine (SCM), presents a model-based design tailored to represent hierarchical decisions and abstract semantic states. Inspired by geometric representations such as simplex coding in multiclass SVMs and decision policies over partially observable environments, the SCM unifies the strengths of both statistical margin-based learning and structured geometric encoding.

In simpler terms, a Partially Observable Markov Decision Process (POMDP) is like making decisions in a foggy environment where one cannot see the entire picture at once. For instance, imagine an autonomous drone navigating through a city during heavy fog. It cannot directly observe all buildings, pedestrians, or vehicles, so it must rely on partial sensor readings (e.g., LiDAR scans, camera frames) to estimate what is really around it. The POMDP framework mathematically models this uncertainty by combining three elements: States, (what is truly happening in the environment, even if hidden), Observations (what the system actually senses), and Actions (decisions made based on those observations). By continuously updating its belief about the hidden state, the system can make better decisions despite incomplete information.

The Structured Cone Machine (SCM) takes this a step further by embedding these belief states into a geometric cone-shaped representation. This conic embedding helps preserve both the hierarchy and relationships between features in high-dimensional visual

data. Think of it as projecting complex patterns-like seasonal vegetation changes in satellite images onto a structured shape where distances and angles have semantic meaning. This makes it easier for the model to recognize patterns across scales and contexts.

Belief-aware optimization (ABO) ensures that the learning process adapts in real-time as the model's confidence changes. If the system is highly uncertain (e.g., detecting an unusual urban feature it hasn't seen before), ABO adjusts the optimization strategy to explore more possibilities. Conversely, when confidence is high, it focuses on refining the solution. This adaptability improves robustness, especially when handling seasonal shifts, mixed urban-rural regions, or noisy sensor data.

Combining POMDP reasoning, SCM's geometric structuring, and ABO's adaptive learning enables our framework to handle complex, high-dimensional image analysis tasks with both precision and resilience. This has direct applications in sustainable urban planning, disaster response mapping, and environmental monitoring, where decisions must be made quickly despite uncertain or incomplete data. To enhance interpretability for readers unfamiliar with the underlying mathematics, we include a summary diagram of the proposed SCM-ABO framework, illustrating how POMDP reasoning, conic geometric embedding, and belief-aware optimization interact to produce robust segmentation outputs.

Figure 1 shows the end-to-end flow of our framework, starting from an input image, passing through POMDP reasoning for decision-making under uncertainty, conic geometric embedding for structured feature representation, and belief-aware optimization for adaptive learning, culminating in the final segmented output.

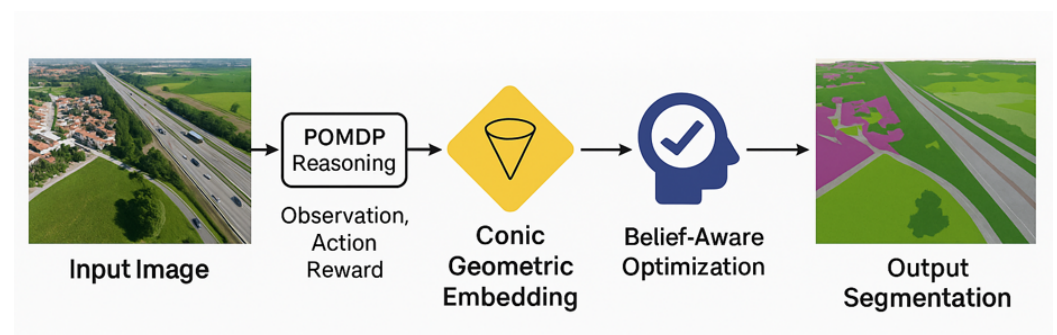


Figure 1. Summary diagram of the SCM-ABO framework showing the interaction between POMDP reasoning, conic geometric embedding, and belief-aware optimization.

The model introduces a conic embedding space with tractable optimization properties, ensuring that multiple structured classes or states can be jointly modeled without requiring hard combinatorial decompositions. The model is constructed to preserve both expressive capacity and computational feasibility. The derivation involves matrix representations, convex relaxations, and factorization with kernel alignment to achieve a balance between interpretability and discriminative power. In Section 3.4, we introduce a strategy, referred to as Adaptive Belief Optimization (ABO), that supports effective training and deployment of the proposed SCM model under real-world constraints. ABO employs decision-theoretic foundations to optimize policy decisions in scenarios where labels may be noisy, rewards are sparse, or actions influence future states. The strategy builds upon reinforcement-style updates but extends them by incorporating a belief-aware regularization that dynamically adjusts the model's internal representation of uncertainty and reliability. This section emphasizes how such a strategy allows SCM to be not only predictive but also adaptive, improving its robustness against domain shifts, occlusions, or compositional ambiguities. The strategy is developed in a streaming data setting, which requires more on-line learning principles and memory-efficient approximations.

3.2. Preliminaries

We begin by formalizing the structured learning problem underpinning the *Landscape* framework. The primary goal is to learn a predictive function or a decision policy in a structured output space, where both the input observations and the output configurations exhibit high-order dependencies, latent uncertainty, or partial observability. This section develops mathematical and notational scaffolding that supports the subsequent design of the model and strategy. Our formulation draws on tools from optimization theory, structured prediction, and decision processes.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space, $\mathcal{Y}$ be a structured output space, and $\mathcal{S}$ be an optional latent space used to encode hidden or unobserved semantic states. Let $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ be an observed training example, with $i = 1, \dots, n$. The learning task is to construct a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ (or more generally, $f : \mathcal{X} \rightarrow \Delta(\mathcal{Y})$) that minimizes a risk functional over some structured loss $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$. The process begins with an input image I , which is processed by the backbone network to extract multi-scale feature representations F_0 . This operation is defined in Equation (1) and is visually depicted in Figure 3, *Backbone Feature Extraction Module*. The population risk is mathematically defined as

$$\mathcal{R}(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(f(x), y)], \quad (1)$$

where $\mathcal{D}$ denotes the joint distribution over inputs and structured outputs. Since $\mathcal{D}$ is unknown, we minimize the empirical risk with regularization, expressed as

$$\hat{\mathcal{R}}_\lambda(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) + \lambda \Omega(f), \quad (2)$$

where $\Omega(f)$ is a regularizer enforcing structure such as smoothness, sparsity, or margin constraints. The backbone features F_0 are transformed using the *Conic Output Embedding* module to obtain F_1 , as given by the formulation in Equation (2). This operation corresponds to Figure 3, *Conic Output Embedding Module*. The embedding ensures that the feature space is structured for downstream alignment and prediction tasks. In structured output settings, the loss ℓ is not decomposable over individual labels or tokens. For instance, in sequence labeling tasks, one may use the Hamming loss, structured hinge loss, or negative log-likelihood over output sequences as follows:

$$\ell_{\text{hinge}}(f(x), y) = \max_{y' \in \mathcal{Y}} [\Delta(y, y') + \langle w, \phi(x, y') - \phi(x, y) \rangle], \quad (3)$$

where $\phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ is a joint feature map, and $\Delta(y, y')$ is a task-specific loss function. To model partial observability, we introduce a latent variable $s \in \mathcal{S}$ representing hidden context, state, or alignment. Let the model decompose as $f(x) = \arg \max_{y \in \mathcal{Y}} \max_{s \in \mathcal{S}} F(x, y, s; \theta)$, where

$$F(x, y, s; \theta) = \langle \theta, \phi(x, y, s) \rangle, \quad (4)$$

with parameters $\theta \in \mathbb{R}^d$ to be learned. This formulation, as expressed in the above equation, gives rise to a latent structured prediction problem, requiring joint inference over (y, s) . In *Landscape*, structured output configurations are encoded in a conic representation inspired by simplex geometry, formulated as

$$\mathcal{C} = \{\mathbf{v} \in \mathbb{R}^{K-1} \mid \langle \mathbf{v}, \mathbf{c}_k \rangle = \delta_k, \text{ for } k = 1, \dots, K\}, \quad (5)$$

where $\{\mathbf{c}_k\}$ form a simplex code. The model learns to align predictions with the conic directions associated with correct outputs. The above formalism serves to establish a

rigorous mathematical foundation for the *Landscape* method. The formulations reflect the complexities of learning in structured, partially observable, and high-dimensional environments. In the following section, we construct a model based on the Structured Cone Machine that operationalizes this formalism and addresses the computational and representational challenges posed by the structure of $\mathcal{Y}$ and $\mathcal{S}$.

3.3. Structured Cone Machine (SCM)

To improve the coherence and readability of our framework, we provide an integrated schematic diagram that illustrates the relationships among the Structured Cone Machine (SCM), Adaptive Belief Optimization (ABO), Latent Feature Alignment, and Kernel/Multiview Extensions. This overview contextualizes the interaction of key components in our model pipeline (as shown in Figure 2). We introduce the Structured Cone Machine (SCM), a predictive model that combines geometric conic embeddings with latent structure and convex optimization. The SCM is designed to handle structured output tasks under ambiguity and high-order constraints, with theoretical guarantees and interpretability (as shown in Figure 3). This explicitly clarifies how the upper pipeline relates to the three detailed modules shown below, while also defining all symbols and legend items. Figure 3 now has a clear two-level organization: The top row depicts the full encoder–decoder pipeline, whereas the bottom panels illustrate three functional modules in detail. Specifically, Block A corresponds to the *Sparse Self-Attention (SSA, left, green panel)*, Block B corresponds to the *Dilformer/Latent Feature Alignment (LFA, center, peach panel)*, and Block C corresponds to the *Dilated Self-Attention (DSA, right, green panel)*. Each “Conic Output Embeddings” box along the pipeline invokes one of these modules at the corresponding spatial scale.

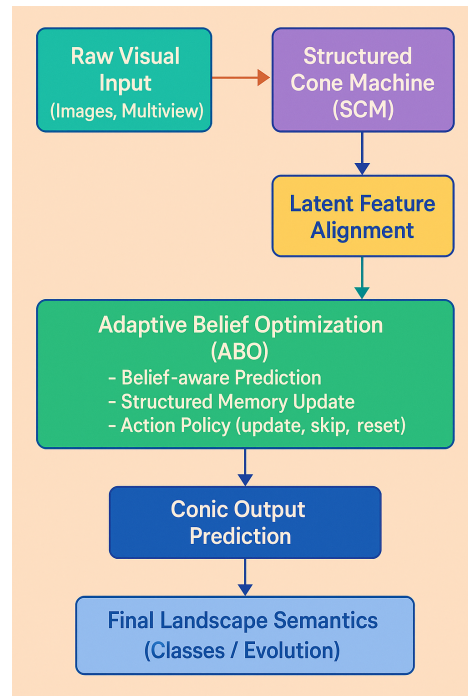


Figure 2. An overview of the SCM–ABO framework for landscape architecture evolution analysis. Raw visual inputs are first processed by a deep feature extractor. The Structured Cone Machine (SCM) projects these features into a conic embedding space for structured prediction. Latent Feature Alignment refines semantic consistency, while Adaptive Belief Optimization (ABO) dynamically adjusts predictions based on belief updates and memory. Final outputs represent semantic landscape classes or temporal evolution states.

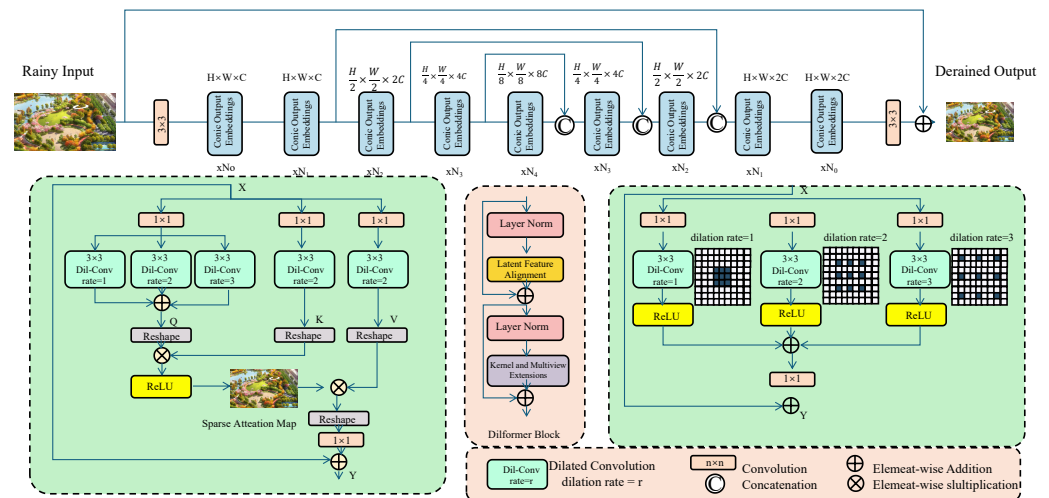


Figure 3. Overall architecture and detailed modules of the proposed network. The top row shows the full encoder–decoder pipeline, while the bottom panels depict three functional modules: Sparse Self-Attention (SSA, left, green panel), Dilformer/Latent Feature Alignment (LFA, center, peach panel), and Dilated Self-Attention (DSA, right, green panel). Feature maps are progressively transformed from $H \times W \times C$ to $H/2 \times W/2 \times 2B$, $H/4 \times W/4 \times 4B$, and $H/8 \times W/8 \times 8B$, with repeat counts $\times N_i$ applied at each stage. In the decoder, features are upsampled back to $H \times W \times 2C$, before a final 3×3 convolution produces the restored output. **Module details:** SSA constructs query, key, and value (Q, K , and V) using dilated 3×3 convolutions with rates 1, 3, and 2, gating inputs via element-wise multiplication ($\otimes$), and residual addition ($\oplus$). Dilformer/LFA applies normalization and latent alignment to project features into the conic output space. DSA stacks dilated convolutions (rates 1, 2, 3) with residual and concatenation operations. **Icons:** circled $\oplus$ = addition, circled $\otimes$ = multiplication, $n \times n$ = convolution, “Concatenate” = channel concatenation, and “dil-conv (rate = r)” = dilated convolution.

In the encoder (downward path), the feature maps of size $H \times W \times C$ are processed by Block A (SSA), repeated N_0 times, followed by Block B (Dilformer/LFA), repeated N_1 times, and then progressively downsampled to $H/2 \times W/2 \times 2B$, $H/4 \times W/4 \times 4B$, and $H/8 \times W/8 \times 8B$, where Block C (DSA) is applied with repeat counts N_2 , N_3 , and N_4 to capture multi-rate context. In the decoder (upward path), features at resolutions $H/4 \times W/4 \times 4B$ and $H/2 \times W/2 \times 2B$ are processed by Block B and Block A, respectively, before returning to the full-resolution map of size $H \times W \times 2C$. A final 3×3 convolution then produces the de-rained output. The labels $\times N_i$ indicate the number of repeated applications of the corresponding module at each scale.

The functionality of each module is explicitly described in the figure and text. Block A (SSA) constructs the query, key, and value (Q, K, V) using three dilated 3×3 convolutions with rates 1, 3, and 2, followed by 1×1 projections and reshaping. These yield sparse attention maps, which gate the input through element-wise multiplication ($\otimes$), with a residual connection added via ($\oplus$). Block B (Dilformer/LFA) applies Layer Normalization, Latent Feature Alignment, and Kernel/Multiview Extensions, thereby aligning intermediate features with the conic output space. Block C (DSA) consists of stacked 3×3 dilated convolutions with dilation rates 1, 2, and 3, each followed by a 1×1 projection and ReLU activation, with results aggregated by residual addition ($\oplus$) and concatenation. Shapes along the top pipeline are explicitly annotated as

$$H \times W \times C \rightarrow H/2 \times W/2 \times 2B \rightarrow H/4 \times W/4 \times 4B \rightarrow H/8 \times W/8 \times 8B \rightarrow \dots \rightarrow H \times W \times 2C,$$

ensuring that dimensional changes are traceable at every stage. Importantly, each “Conic Output Embeddings” box denotes projection into the conic label space, as described in Section 3.2, so that all scales consistently employ the same geometric coding before and after the attention modules.

The caption now also defines all icons: Circled $\oplus$ denotes element-wise addition, circled $\otimes$ denotes element-wise multiplication, the $n \times n$ symbol represents convolution, and the “Concatenate” symbol indicates channel concatenation; the “dil-conv (rate = r)” tiles specify dilated convolutions with dilation factor r . Finally, the revised text introduces explicit cross-references between the equations and the named modules (SSA, Dilformer/LFA, DSA) at the points where their mathematical formulations are presented, thereby removing any ambiguity between the overall pipeline and the detailed module diagrams.

In Figure 3, SSA denotes the *Sparse Self-Attention* module, which captures long-range dependencies while reducing computational overhead via sparsity constraints. DSA refers to the *Dilated Self-Attention* module, which enlarges the receptive field through dilation rates of 1, 2, and 3, thereby integrating multi-scale contextual information. *Dil-Conv* represents *Dilated Convolution*, where the dilation rate r controls the spacing between kernel elements to balance resolution preservation and contextual coverage.

Each architectural block in the schematic is labeled and color-coded, with functional roles described directly in the legend: The green-shaded block depicts the SSA process (including query, key, and value generation, sparse attention map computation, and residual addition), the orange-shaded block illustrates the Dilformer (Layer Normalization, Latent Feature Alignment, and Kernel/Multiview Extensions), and the light-green block represents the DSA module (stacked dilated convolutions and ReLU activations). Additionally, arrows indicate the explicit data flow between modules, and icons for operations (element-wise addition, multiplication, and concatenation) are annotated in the legend. These modifications ensure that the architecture, its components, and the data processing pipeline can be fully understood without requiring the reader to cross-reference earlier sections. This figure illustrates the step-by-step flow of the proposed system for processing an input image of size $H \times W \times C$. First, the image passes through the initial feature extraction layers, where low-level visual patterns such as edges, textures, and colors are captured.

These extracted features are then fed into Block A (feature enhancement), which enriches and amplifies important spatial details. Next, the output from Block A enters Block B (context aggregation), which gathers multi-scale contextual information to improve the understanding of global and local structures. The refined features are subsequently processed by Block C (prediction refinement), which fine-tunes the feature maps to produce accurate and high-quality predictions. Finally, the output is passed through the final layers to generate the desired result, with each block seamlessly integrated into the pipeline to ensure smooth information flow from input to output.

• Conic Output Embeddings

In structured classification tasks where relational geometry between output classes matters—such as taxonomy prediction, morphological categorization, or semantic segmentation—the Structured Cone Machine (SCM) replaces traditional one-hot encodings with geometrically structured output embeddings (As shown in Figure 4). In the proposed architecture, H and W denote the height and width of the input feature map (in pixels), respectively, while C represents the number of channels. The processing begins with Layer Normalization (LN), which standardizes the input feature distribution to enhance training stability, followed by a linear projection layer (*Project*) that maps features into the target embedding dimension d . Sparse Self-Attention (SSA) is then applied to focus computation on the most informative regions by ignoring irrelevant positions, whereas Dilated Self-Attention (DSA) expands the receptive field via dilation rates to capture broader contextual

information. A Rectified Linear Unit (ReLU) introduces non-linearity by suppressing negative activations, and learnable weight parameters W (yellow blocks in the figure) are employed for feature combination or transformation. Attention scores are normalized using the *Softmax* function to obtain probability distributions over positions. Finally, the Cone Output Embedding assigns each class a unique conic direction, enabling classification based on angular separation in the embedding space.

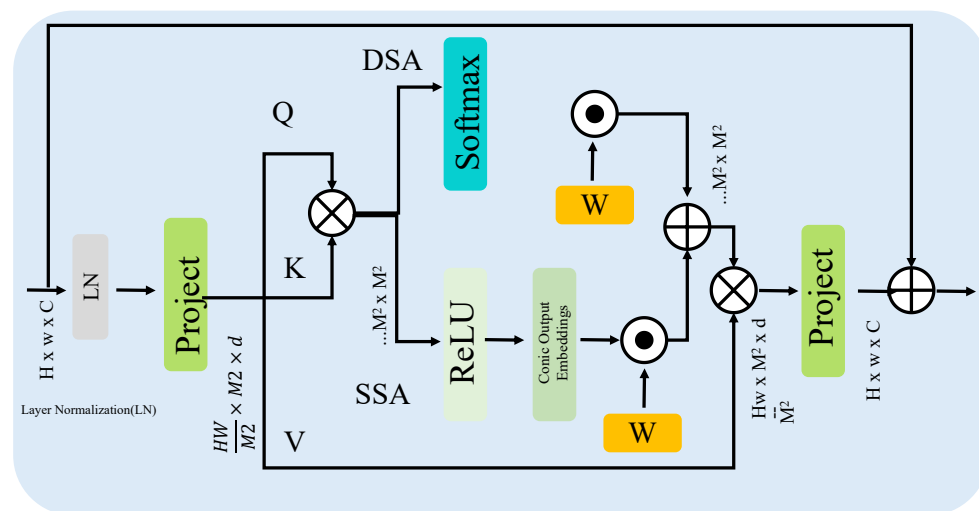


Figure 4. Detailed illustration of the three modules highlighted in the pipeline. The architecture begins with Layer Normalization (LN) and projection of input features ($H \times W \times C$) into a target dimension d . Features are then processed through Sparse Self-Attention (SSA) and Dilated Self-Attention (DSA) modules, capturing both focused and multi-scale contextual information. ReLU activations and learnable weight parameters (W) further refine the features before alignment with class-specific conic directions in the embedding space. Softmax operations normalize attention scores, while cone embeddings enable angular decision boundaries for structured output prediction. Arrows denote the sequential data flow, and all variables (H , W , C , and d) are defined for clarity.

The Figure 4 depicts the processing pipeline of the *Structured Cone Machine* (SCM) with Conic Output Embeddings. The input image of dimensions $H \times W \times C$ is first normalized using *Layer Normalization* (LN) to stabilize feature distributions. These normalized features are projected into a lower-dimensional representation through a linear projection layer. The projected features are then processed by two attention mechanisms: *Sparse Self-Attention* (SSA) and *Dilated Self-Attention* (DSA). SSA selectively focuses on critical regions by introducing sparsity into the attention computation, while DSA expands the receptive field via dilation, enabling multi-scale context capture. The results are further transformed by a ReLU activation function and integrated through learned weights (W) at multiple points in the network. Finally, features are realigned using the *Cone Output Embedding* mechanism, where each class is represented by a distinct conic direction in the embedding space, enabling angular decision boundaries for classification. The final projection produces the structured output predictions.

Each class label $k \in \{1, \dots, K\}$ is assigned to a conic direction $\mathbf{c}_k \in \mathbb{R}^{K-1}$, chosen such that all class vectors lie on a unit-radius spherical simplex centered at the origin. This encoding enforces angular separation between class labels, capturing mutual exclusivity while preserving geometric uniformity. Specifically, the conic codebook satisfies

$$\langle \mathbf{c}_i, \mathbf{c}_j \rangle = \begin{cases} 1, & i = j \\ -\frac{1}{K-1}, & i \neq j \end{cases}, \quad \sum_k \mathbf{c}_k = 0. \quad (6)$$

This formulation, as outlined in the above equation, ensures that all codes are equidistant in angle, and their sum vanishes, forming a zero-mean, maximally discriminative embedding basis. Such a conic output structure is particularly advantageous when class similarities or transitions must be interpreted geometrically, such as in gradual evolutionary categories or skill acquisition levels.

To enhance the interpretability of Equation (6), we introduce a visual guide in Figure 5 that demonstrates the principle of conic separation. In this arrangement, each class embedding is mapped to a unique directional vector radiating from the origin, with equal angular spacing between them. This configuration forms a set of non-overlapping conic regions in the embedding space, enabling clear class discrimination while preserving geometric consistency. By visualizing the embeddings in this manner, the concept becomes accessible even to readers unfamiliar with the underlying cone geometry.

Figure 5 visually demonstrates conic separation in a high-dimensional embedding space, illustrating how class labels are mapped to distinct, uniformly spaced conic directions on the surface of a unit-radius sphere. Each cone represents a class vector oriented at a fixed angular distance from others, ensuring mutual exclusivity while maintaining geometric uniformity. The symmetry reflects the property from Equation (6) where the dot product between identical class vectors equals 1, while between different class vectors equals a constant negative value, ensuring equal angular spacing. The convergence of cone bases at the origin symbolizes the zero-mean constraint $\sum_k c_k = 0$, which balances the embedding space and prevents bias toward any class direction. This spatial structure facilitates clear class discrimination while enabling the smooth interpretation of transitions between similar classes.

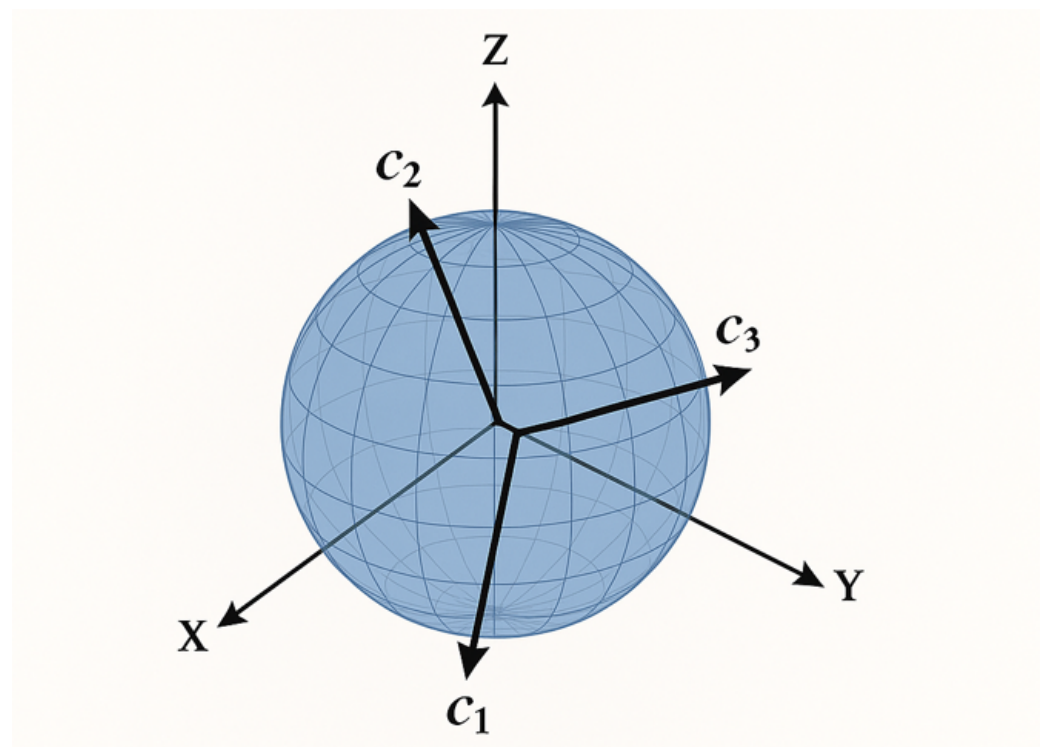


Figure 5. Conic separation with zero-mean constraint ($\sum_k c_k = 0$), placing classes at equal angular intervals for balanced, unbiased, and well-separated embeddings.

Furthermore, the summation constraint $\sum_k c_k = 0$ is now explicitly explained as enforcing a balanced, zero-mean embedding space. This ensures that the embedding vectors are symmetrically distributed around the origin, preventing directional bias toward any specific class. Such balance is critical for maintaining stability in the optimization

process and improving the model's ability to generalize across diverse datasets. For an input sample $x \in \mathbb{R}^d$, a feature mapping $\psi(x)$ is computed using a deep encoder or kernelized transformation, followed by a linear scoring layer that projects the input into the conic output space as specified:

$$f(x) = W^\top \psi(x) + b, \quad \hat{y}(x) = \arg \max_k \langle f(x), \mathbf{c}_k \rangle. \quad (7)$$

Here, $W \in \mathbb{R}^{d \times (K-1)}$ and $b \in \mathbb{R}^{K-1}$ define the learnable projection to the output cone. The model predicts the class whose conic direction has the highest inner product with the projected feature vector $f(x)$, effectively performing angle-based classification in the embedding space. Unlike softmax-based models, which rely on unstructured logits, this approach yields geometrically meaningful decision boundaries. To train SCM, a margin-based structured hinge loss is employed. For a given training example (x_i, y_i) with ground-truth label y_i , the loss encourages the prediction vector $f(x_i)$ to align more closely with the correct conic direction $\mathbf{c}_{y_i}$ than with any other $\mathbf{c}_k$, by at least a margin δ :

$$\mathcal{L}_i = \sum_{k \neq y_i} [\langle \mathbf{c}_k - \mathbf{c}_{y_i}, f(x_i) \rangle + \delta]_+. \quad (8)$$

This hinge-style formulation penalizes all incorrect classes that are too close in angle to the predicted direction, promoting sharp angular separation and encouraging the model to maintain strong confidence margins as formulated in the above equation. To prevent overfitting and enhance generalization, a conic norm regularizer is applied to both the projection weights and the output predictions, as given by

$$\mathcal{R}_{\text{cone}} = \lambda_1 \|W\|_F^2 + \lambda_2 \|f(x)\|^2, \quad (9)$$

where $\|\cdot\|_F$ is the Frobenius norm and λ_1 and λ_2 control the weight decay and output energy, respectively. This regularization ensures that the classifier respects the geometric assumptions of the conic embedding space. Moreover, to better handle ambiguous samples near class boundaries, an angular temperature scaling can be introduced during inference to convert directional scores into probabilistic confidence levels by employing

$$p_k(x) = \frac{\exp(\langle f(x), \mathbf{c}_k \rangle / \tau)}{\sum_j \exp(\langle f(x), \mathbf{c}_j \rangle / \tau)}, \quad (10)$$

where τ is a temperature parameter controlling the sharpness of the prediction distribution. Lower τ emphasizes the winner-take-all decision, while higher τ smooths predictions across neighboring classes, beneficial in hierarchically related outputs. Overall, the SCM framework provides a structured geometric alternative to conventional classification, aligning latent predictions with interpretable directional codes. Its design is particularly suited for tasks requiring relational inductive bias, uncertainty-aware outputs, or interpretable transitions between adjacent classes, and integrates seamlessly with deep neural backbones for end-to-end training.

• Latent Feature Alignment

To handle inherent ambiguities in structured input-output mappings, the Structured Cone Machine (SCM) introduces a latent variable framework wherein each input x_i is associated with an unobserved configuration $s_i \in \mathcal{S}$ that modulates its feature embedding. These latent variables represent interpretable factors such as viewpoint, context, or missing

modality, and are incorporated into the model through the composite feature map $\psi(x_i, s_i)$. The prediction function is, thus, extended to be conditioned on latent structure as follows:

$$f(x_i, s_i) = W^\top \psi(x_i, s_i) + b, \quad (11)$$

where W and b are model parameters shared across all latent configurations. During training, for each input-label pair (x_i, y_i) , the model evaluates the structured margin loss over all possible latent states and selects the one that yields the highest hinge violation, formalized as

$$\tilde{\mathcal{L}}_i = \max_{s \in \mathcal{S}} \sum_{k \neq y_i} [\langle \mathbf{c}_k - \mathbf{c}_{y_i}, f(x_i, s) \rangle + \delta]_+, \quad (12)$$

with δ denoting the desired conic margin. This loss promotes latent structures that best explain the discrepancy between the current prediction and the ground truth label. To optimize this objective efficiently, we adopt an alternating strategy: In the E-step, the best latent configuration $\hat{s}_i = \arg \max_s \tilde{\mathcal{L}}_i$ is selected via enumeration or greedy heuristics; in the M-step, model parameters W are updated to minimize the empirical risk given fixed $\hat{s}_i$. Before prediction, F_2 undergoes a series of intermediate transformations, such as attention weighting, normalization, and residual connections. These operations are captured by Equation (12). Figure 3 corresponds to the sub-blocks within *Module C*, where feature refinement occurs. Furthermore, to prevent overfitting to spurious latent states, this introduces an entropy-based regularization that encourages distributional smoothness over $\mathcal{S}$:

$$\mathcal{R}_{\text{ent}} = - \sum_{i=1}^n \sum_{s \in \mathcal{S}} p_i(s) \log p_i(s), \quad p_i(s) \propto \exp(-\tilde{\mathcal{L}}_i(s)), \quad (13)$$

thereby promoting latent diversity and robustness. For settings with annotated or weakly labeled latent variables, the optimization can be further guided via a KL-divergence alignment that is represented as

$$\mathcal{R}_{\text{KL}} = \sum_{i=1}^n \text{KL}(q_i(s) \| p_i(s)), \quad (14)$$

where $q_i(s)$ is the supervision-derived distribution and $p_i(s)$ is the model-inferred posterior. The final training objective becomes

$$\mathcal{L}_{\text{total}} = \frac{1}{n} \sum_{i=1}^n \tilde{\mathcal{L}}_i + \lambda_1 \mathcal{R}_{\text{ent}} + \lambda_2 \mathcal{R}_{\text{KL}} + \frac{\lambda_3}{2} \|W\|_F^2, \quad (15)$$

where λ_1 , λ_2 , and λ_3 are regularization hyperparameters in the above equation. This latent alignment approach significantly enhances the SCM's ability to disambiguate structured outputs under partial observability or missing modality conditions, while maintaining interpretability and generalization.

• Kernel and Multiview Extensions

The Structured Cone Machine (SCM) framework extends beyond linear models by incorporating both kernelized and multiview feature representations, thereby enabling flexible, non-linear, and modality-aware structured prediction. These extensions preserve the conic embedding geometry while broadening the SCM's applicability to heterogeneous data sources and complex decision boundaries. In the kernelized variant, SCM lifts the input x into a reproducing kernel Hilbert space (RKHS) via a kernel function $K(x_i, x_j)$,

avoiding the need for explicit feature mapping. The classifier function is expressed in dual form as

$$f(x) = \sum_{i,k \neq y_i} \alpha_i^k (\mathbf{c}_{y_i} - \mathbf{c}_k) K(x_i, x), \quad \hat{y}(x) = \arg \max_k \langle f(x), \mathbf{c}_k \rangle, \quad (16)$$

where α_i^k are dual coefficients optimized during training, and $K(\cdot, \cdot)$ is a positive-definite kernel such as the RBF, polynomial, or graph kernel depending on input modality. This representation allows SCM to capture non-linear relationships between inputs while preserving the directional decision rule inherent in the conic embedding. The classifier, thus, generalizes to complex manifolds without sacrificing the geometric interpretability or margin-based separation of its outputs. In applications involving multimodal or multiview data—such as audio-visual classification, multi-angle morphology analysis, or structured sensor fusion—the SCM supports view-specific encodings by learning separate embeddings for each modality or feature group. Given V distinct views or channels, each with its own feature extractor $\psi^{(v)}(x)$, the conic predictor is defined as

$$f(x) = \sum_{v=1}^V W^{(v)\top} \psi^{(v)}(x) + b, \quad (17)$$

where $W^{(v)}$ are projection weights for view v , and b is a shared bias vector. This formulation enables the SCM to model decomposable feature spaces where different views contribute complementary information, such as shape vs. texture, or kinematics vs. force. The training objective jointly optimizes all $W^{(v)}$ under the structured hinge loss while preserving shared conic output space alignment, promoting cooperative learning across modalities. To balance the contributions of each view and enforce consistency across latent representations, we introduce a view regularization term expressed as

$$\mathcal{R}_{\text{view}} = \sum_{v=1}^V \lambda_v \|W^{(v)}\|_F^2 + \gamma \sum_{v < v'} \|f^{(v)}(x) - f^{(v')}(x)\|^2, \quad (18)$$

where $f^{(v)}(x) = W^{(v)\top} \psi^{(v)}(x)$ is the partial prediction from view v , λ_v controls view-specific weight decay, and γ penalizes disagreement between views. This encourages aligned representations across modalities and ensures that the final prediction is not dominated by any single source, which is critical in noisy or partially missing data scenarios. The refined features are passed to the *Belief-Aware Conic Prediction* module, which produces the preliminary prediction map Y_{pred} . The prediction process is governed by Equation (18), which defines the conic mapping, belief weighting, and prediction fusion. This corresponds to Figure 3, *Belief-Aware Conic Prediction Module*. In the kernelized multiview setting, each view may have its own kernel $K^{(v)}(x_i, x_j)$, allowing even greater flexibility and represented as

$$f(x) = \sum_{v=1}^V \sum_{i,k \neq y_i} \alpha_i^{(v,k)} (\mathbf{c}_{y_i} - \mathbf{c}_k) K^{(v)}(x_i, x), \quad (19)$$

thereby allowing view-specific similarity metrics. This is particularly useful in heterogeneous data regimes where each modality may follow different statistical properties. The SCM's convexity in the dual and compatibility with conic embedding constraints enables efficient optimization using second-order solvers or projected subgradient methods. This makes the SCM both scalable and theoretically grounded, with strong generalization guarantees. Through Kernel and Multiview Extensions, the SCM bridges geometric output structures with nonlinear and multimodal input representations, enabling high-fidelity, interpretable, and robust predictions across a wide range of structured learning tasks.

3.4. Adaptive Belief Optimization (ABO)

To enhance the Structured Cone Machine (SCM) under uncertainty, we propose Adaptive Belief Optimization (ABO)—a dynamic framework that integrates belief reasoning, action control, and latent uncertainty into model adaptation (as shown in Figure 6).

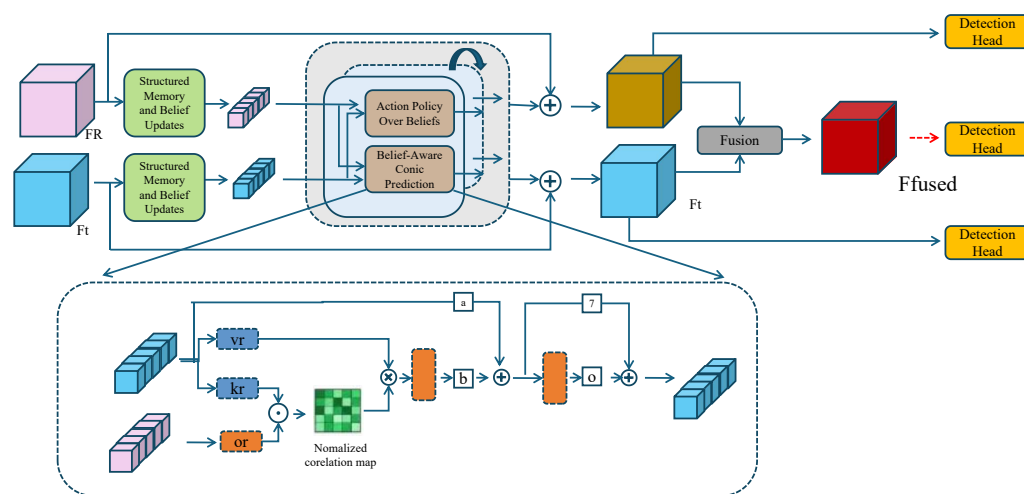


Figure 6. Schematic diagram of Adaptive Belief Optimization (ABO). Overview of the belief-aware detection framework integrating structured memory updates, action-guided conic prediction, and correlation-based feature enhancement. Feature representations (F_R , F_t) are processed through *Structured Memory and Belief Update* modules, refined by *Action Policy Over Beliefs* and *Belief-Aware Conic Prediction*, and then directed to *Detection Heads* either individually or via a fusion mechanism. The lower section illustrates the correlation computation process, where value (v_r), key (k_r), and query (o_r) features produce a normalized correlation map, which undergoes scaling, convolutional transformations, and activation to yield enhanced feature maps for detection.

In the illustrated framework, F_R refers to the feature representation derived from the reference or primary input stream, while F_t represents the feature representation from the target or secondary input stream. The *Structured Memory and Belief Updates* module stores and updates feature memory states while incorporating belief information to maintain temporal and contextual consistency. The *Action Policy Over Beliefs* serves as a decision-making mechanism that selects optimal actions based on the current state of learned beliefs. The *Belief-Aware Conic Prediction* module projects features into a conic embedding space, enabling the angular separation of classes for more robust classification. *Fusion* denotes the process of merging multiple feature maps to enhance their discriminative power. The *Detection Head* is a task-specific prediction module responsible for generating detection outputs, such as bounding boxes or class scores. The terms v_r , k_r , and o_r correspond to value, key, and query features, respectively, which are employed in correlation and attention mechanisms. The *normalized correlation map* is a similarity matrix indicating the degree of alignment between features from two sources after normalization. The parameters a and b are learnable scaling factors that modulate the influence of feature maps, while σ represents the sigmoid activation function that normalizes values between 0 and 1.

Figure 6 illustrates a belief-aware detection framework that integrates structured memory updates with conic prediction for improved decision-making. Initially, feature representations F_R and F_t undergo *Structured Memory and Belief Updates*, which refine and store contextual cues over time. These updated features are processed by two core modules: *Action Policy Over Beliefs*, which guides decision-making based on belief states, and *Belief-Aware Conic Prediction*, which aligns outputs along angular decision boundaries for structured predictions. The resulting outputs are aggregated through additive operations and sent either directly to individual *Detection Heads* or combined through a

Fusion step to generate a fused feature representation F_{fused} , which is then processed by the final Detection Head. A detailed block at the bottom of the figure shows the correlation computation process, where value (v_r), key (k_r), and query (o_r) features are used to generate a normalized correlation map. This map undergoes further transformations, scaling (a, b), and convolutional operations before producing enhanced feature maps for downstream detection.

• Belief-Aware Conic Prediction

In dynamic or partially observable settings, predictive uncertainty over latent factors must be explicitly modeled to improve decision robustness (as shown in Figure 7). Feature extraction refers to the process of converting raw data into meaningful numerical representations that encode spectral, spatial, and temporal information. The *Spectral Transformer Block* learns frequency-domain relationships by analyzing variations in signal power across different spectral components. The *Spatial Transformer Block* captures positional and structural dependencies within spatial layouts of features. The *Temporal Attention Block* selectively focuses on temporally significant patterns by computing weighted feature aggregations over time. The *Transformer Block* implements multi-head self-attention with normalization and feed-forward layers to enhance feature integration. *Belief-Aware Conic Prediction* projects features into a conic embedding space, increasing angular separation between classes for improved discriminability. The *Fully Connected (FC) Layer* aggregates processed features into a fixed-length vector for classification. In the classification stage, predicted outcomes fall into categories such as positive, neutral, or negative. *Positional Encoding* embeds sequential order information into features, enabling temporal modeling. Within the *Temporal Attention Block*, *Weighted Sum*, *Softmax*, and *Linear* layers work together to assign, normalize, and transform attention scores for optimal temporal feature selection.

The illustrated framework depicts a belief-aware detection and classification pipeline for spatiotemporal feature learning. Initially, raw data undergo *feature extraction*, producing multi-channel spectral–spatial–temporal representations. These are processed sequentially through the *Spectral Transformer Block*, which captures frequency-domain dependencies, and the *Spatial Transformer Block*, which models spatial relationships among features. The *Temporal Attention Block* then emphasizes relevant temporal patterns across frames by assigning adaptive attention weights. Features are further refined through a *Fully Connected (FC) Layer* before being classified into categories such as positive, neutral, or negative. Parallely, a *Belief-Aware Conic Prediction* module transforms extracted features into a conic embedding space to enhance class separability, ensuring robust predictions.

The Figure 7 is an integral part of the proposed Belief-Aware Conic Prediction framework. It visually illustrates the incorporation of a belief distribution $b_i \in \Delta(\mathcal{S})$ over latent structures $s \in \mathcal{S}$, such as pose configurations, sensory alignment, or context states. Rather than committing to a single latent configuration, the framework computes the belief-weighted model response by marginalizing the latent-conditioned outputs of the Structured Cone Machine (SCM), as mathematically defined in Equation (18). This figure directly corresponds to the Belief-Aware Conic Prediction Module in the proposed pipeline and is not merely illustrative, but a core representation of the methodology.

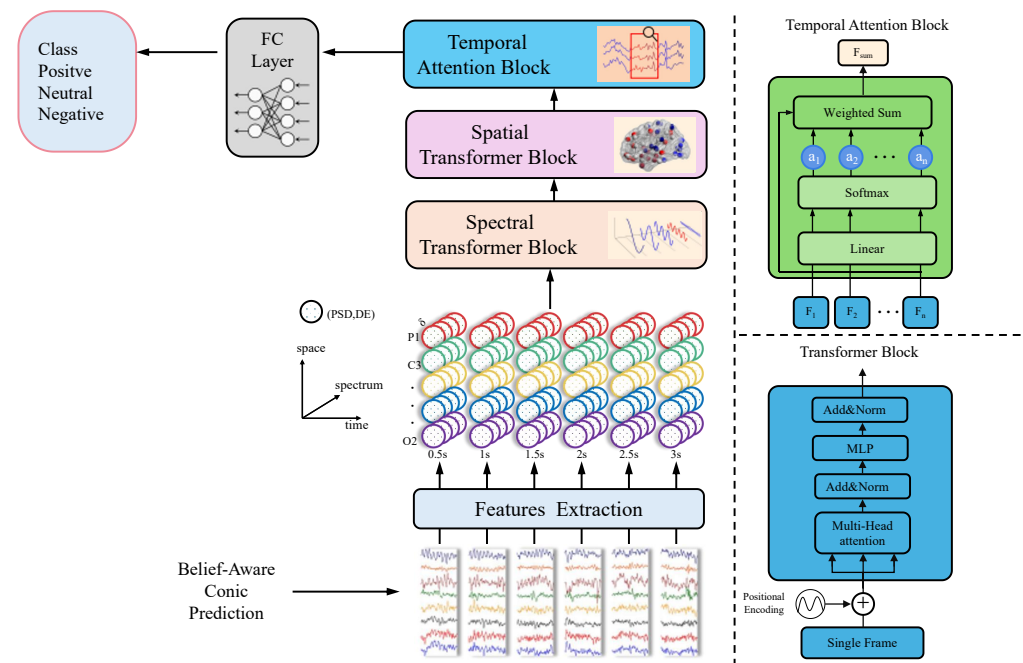


Figure 7. Schematic diagram of the Belief-Aware Conic Prediction framework. Overview of the proposed belief-aware detection and classification framework, integrating spectral, spatial, and temporal feature modeling. Raw inputs are processed through a *Feature Extraction* module, followed by a *Spectral Transformer Block*, *Spatial Transformer Block*, and *Temporal Attention Block* to capture frequency, spatial, and temporal dependencies, respectively. A *Fully Connected Layer* aggregates features for final classification into categories such as positive, neutral, or negative. In parallel, a *Belief-Aware Conic Prediction* module projects features into a conic embedding space to enhance class separability. The right panel illustrates the internal structure of the Temporal Attention and Transformer blocks, highlighting attention computation, normalization, and multi-head processing.

The Belief-Aware Conic Prediction framework incorporates a belief distribution $b_t \in \Delta(\mathcal{S})$ over latent structures $s \in \mathcal{S}$, such as pose configurations, sensory alignment, or context states. Rather than committing to a single latent configuration, we define the belief-weighted model response by marginalizing the latent-conditioned outputs of the Structured Cone Machine (SCM), described as

$$\bar{f}(x_t, b_t) = \sum_{s \in \mathcal{S}} b_t(s) f(x_t, s), \quad \hat{y}_t = \arg \max_k \langle \bar{f}(x_t, b_t), \mathbf{c}_k \rangle. \quad (20)$$

Here, $f(x_t, s)$ denotes the SCM's latent-parameterized conic prediction, and $\mathbf{c}_k$ is the canonical vector representing class k in the conic embedding space. The posterior predictive decision, thus, reflects both the uncertainty in latent states and the structured class separation. This aligns the training objectives with this perspective by minimizing the expected conic margin loss under the belief distribution:

$$\mathcal{J}_{\text{ABO}}(W) = \frac{1}{n} \sum_{i=1}^n \sum_{s \in \mathcal{S}} b_i(s) \mathcal{L}(f(x_i, s), y_i) + \frac{\lambda}{2} \|W\|_F^2, \quad (21)$$

where $\mathcal{L}$ is the structured hinge loss over conic directions, and W is the SCM weight matrix. To obtain the belief distributions $b_i(s)$, we use Bayesian updates based on observed signals

or auxiliary outcomes. Let o_t be the observed evidence related to s_t , such as a sensory cue or label proxy. Then, the belief state evolves over time via

$$b_{t+1}(s') = \frac{p(o_t | s') \sum_s p(s' | s) b_t(s)}{\sum_{s''} p(o_t | s'') \sum_s p(s'' | s) b_t(s)}, \quad (22)$$

where $p(o_t | s')$ is the observation model and $p(s' | s)$ is the latent transition kernel. To improve robustness, predictions from multiple scales are fused. This step is defined by Equation (22) and represented in Figure 3, showing fusion arrows and the combined output node. These beliefs can be learned jointly or approximated using domain priors. For increased adaptability, we define an entropy-regularized policy over belief states to determine model actions such as update, skip, or reset as follows:

$$\pi(b_t) = \arg \max_a [Q(b_t, x_t, a) - \tau \cdot \mathcal{H}(b_t)], \quad (23)$$

where Q is a belief-conditioned action-value function, $\mathcal{H}(b_t)$ is the Shannon entropy of the belief, and τ is a regularization constant controlling conservativeness. This mechanism guides the model to balance exploration of uncertain latent states with exploitation of confident predictions. To ensure numerical stability and interpretability, we re-parameterize the prediction using normalized barycentric projections as indicated:

$$\tilde{f}(x_t, b_t) = \frac{\tilde{f}(x_t, b_t)}{\|\tilde{f}(x_t, b_t)\|_2}, \quad \hat{y}_t = \arg \max_k \langle \tilde{f}(x_t, b_t), \mathbf{c}_k \rangle. \quad (24)$$

This normalization allows fair comparison across different input magnitudes and belief variances. Altogether, the belief-aware conic prediction strategy provides a principled, uncertainty-sensitive approach to structured classification under latent ambiguity, improving reliability and generalization in real-world deployment. The normalization in Equation (24) via barycentric projection serves a dual purpose in ensuring fairness and interpretability. By projecting the raw prediction vector $\tilde{f}(x_t, b_t)$ onto a unit ℓ_2 -norm sphere,

$$\tilde{f}(x_t, b_t) = \frac{\tilde{f}(x_t, b_t)}{\|\tilde{f}(x_t, b_t)\|_2},$$

we remove the influence of varying input magnitudes and belief-state variances that could otherwise bias the decision boundaries. This normalization guarantees that comparisons between the projected feature vector $\tilde{f}(x_t, b_t)$ and each class anchor $\mathbf{c}_k$ are based purely on directional similarity rather than scale, thus promoting fairness across heterogeneous data conditions. Additionally, by constraining predictions to a normalized representation, the resulting decision process becomes more interpretable and angular relationships between $\tilde{f}(x_t, b_t)$ and $\mathbf{c}_k$ directly encode confidence and class separation in the conic embedding space. This property enhances both the transparency and the robustness of the belief-aware conic prediction strategy in real-world applications.

• Action Policy Over Beliefs

The Adaptive Belief Optimization (ABO) framework introduces a belief-aware decision policy designed to manage uncertainty in dynamic state estimation tasks such as object tracking, temporal segmentation, or sequential hypothesis refinement. Unlike conventional policies that operate on observable states, ABO defines a policy $\pi : \Delta(\mathcal{S}) \times \mathcal{X} \rightarrow \mathcal{A}$ over the belief space $\Delta(\mathcal{S})$ —a probability distribution over latent states $s \in \mathcal{S}$ —combined with observed input $x_t \in \mathcal{X}$ to choose an action $a_t \in \mathcal{A}$. The action set typically includes discrete operations such as `track` (continue with current belief), `update` (refine belief using new evidence), and `reinit` (reset belief due to high uncertainty or drift). At each timestep t ,

the policy selects the optimal action based on a learned state-action value function, which is summarized as follows:

$$a_t = \arg \max_a Q(b_t, x_t, a), \quad Q \leftarrow Q + \eta \left[r_t + \gamma \max_{a'} Q(b_{t+1}, x_{t+1}, a') - Q \right], \quad (25)$$

where b_t is the current belief distribution, r_t is the reward signal, γ is a discount factor, and η is the learning rate. The Q -function estimates the expected return for taking action a in the current belief–observation pair, and is updated via temporal-difference (TD) learning, allowing the agent to adaptively refine its behavior over time. To prevent the model from overcommitting to uncertain or poorly supported beliefs, ABO incorporates an entropy-based regularization strategy. This encourages exploration and maintains robustness against premature convergence, especially in ambiguous environments or noisy data regimes. The entropy of the belief state is defined as

$$\mathcal{H}(b_t) = - \sum_s b_t(s) \log b_t(s), \quad (26)$$

which quantifies the model’s uncertainty over possible latent states. A high-entropy belief reflects significant ambiguity, whereas low entropy indicates a peaked distribution. Using this metric, the action policy is modified to penalize overconfident decisions in uncertain regions:

$$\pi(b_t, x_t) = \arg \max_a [Q(b_t, x_t, a) - \tau \mathcal{H}(b_t)], \quad (27)$$

where τ is a tunable regularization coefficient that determines the weight of entropy in the action selection process as indicated in above equation. This formulation biases the policy toward conservative actions—such as deferring updates or invoking resets—when belief entropy is high, thereby improving stability and reliability. To further refine belief states, ABO integrates a belief transition model $\mathcal{T}_\phi$ that maps current beliefs and observations into updated distributions. Given an action a_t , the transition dynamics can be expressed as

$$b_{t+1} = \mathcal{T}_\phi(b_t, x_t, a_t), \quad (28)$$

where ϕ are the model parameters learned via variational inference or supervised likelihood maximization. This enables ABO to simulate the downstream effect of its policy choices on belief trajectories, facilitating more informed decision-making in complex temporal environments.

To evaluate policy performance under uncertainty, we presented a risk-adjusted return metric that penalizes low-confidence decisions:

$$\mathcal{R}_{\text{conf}} = \mathbb{E} \left[\sum_t r_t - \lambda (1 - \max_s b_t(s)) \right], \quad (29)$$

where λ is a risk-aversion hyperparameter, and $(1 - \max_s b_t(s))$ represents belief uncertainty. This metric rewards high-reliability decisions while discouraging hasty updates based on diffuse or misleading evidence. Altogether, ABO provides a principled, uncertainty-aware reinforcement learning framework that integrates belief modeling, entropy regularization, and action scheduling. It is particularly effective in applications where latent state evolution is complex, costly to observe, or sensitive to context—such as in neuro-mechanical control, fossil behavior simulation, and autonomous systems with partial observability. The combination of belief dynamics and policy optimization enables ABO to deliver robust, adaptive performance across diverse decision-making domains.

• Structured Memory and Belief Updates

In dynamically evolving environments where latent variables govern observation dynamics, maintaining a coherent internal belief structure is critical for robust decision-making. The belief state $b_t(s)$ over latent variables $s \in \mathcal{S}$ is iteratively refined through a Bayesian update mechanism, integrating observations and latent transition dynamics. At each time step t , upon receiving observation o_t and taking action a_t , the belief update is computed as

$$b_{t+1}(s') = \frac{p(o_t | s') \sum_s p(s' | s, a_t) b_t(s)}{\sum_{\tilde{s}} p(o_t | \tilde{s}) \sum_s p(\tilde{s} | s, a_t) b_t(s)}, \quad (30)$$

where the denominator ensures normalization over the latent state space. The transition model $p(s' | s, a_t)$ captures how actions modify hidden dynamics, while the likelihood model $p(o_t | s')$ evaluates compatibility with new evidence. To ensure sample efficiency and temporal coherence, belief states are coupled with a structured experience memory $\mathcal{M} = \{(b_t, x_t, a_t, r_t, b_{t+1})\}$, storing tuples of belief–action–outcome sequences. The predicted map Y_{pred} is compared with the ground truth Y_{gt} using a composite loss function that integrates region-specific penalties, edge-aware loss, and structural similarity constraints. These are mathematically described in Equation (30) and are shown in Figure 4, *Loss Block*. These samples are periodically drawn to compute the replay loss as expressed:

$$\mathcal{L}_{\text{replay}} = \sum_{(b, x, a, r, b')} \left[Q(b, x, a) - \left(r + \gamma \max_{a'} Q(b', x, a') \right) \right]^2, \quad (31)$$

where $Q(b, x, a)$ is a belief-conditioned action-value function, and γ is the temporal discount factor. This memory-based update encourages temporal credit assignment while avoiding catastrophic forgetting in non-stationary contexts. To further stabilize belief propagation, we incorporate a belief entropy penalty represented in Equation (31) that discourages overly sharp distributions in early learning phases:

$$\mathcal{L}_{\text{ent}} = \sum_t \sum_s b_t(s) \log b_t(s), \quad (32)$$

which acts as a regularizer to maintain sufficient uncertainty when evidence is sparse or conflicting. To capture long-range dependencies and abstract behavioral motifs, we augment the memory $\mathcal{M}$ with a structured graph $\mathcal{G}$ where each node corresponds to a belief embedding and edges encode action-induced transitions. Let $\mathbf{v}_t = \phi(b_t)$ denote an embedding of belief b_t via a neural encoder ϕ . Then, we define a contrastive alignment loss between observed and predicted belief trajectories, expressed as follows:

$$\mathcal{L}_{\text{align}} = \sum_{(t, t')} \mathbb{I}_{[a_t = a_{t'}]} \cdot \|\mathbf{v}_t - \mathbf{v}_{t'}\|^2, \quad (33)$$

where $\mathbb{I}_{[a_t = a_{t'}]}$ denotes that belief transitions are aligned via common actions. This graph-based regularization enforces geometric smoothness over latent decision paths. Decision-making under structured belief evolution is guided by a dynamic policy $\pi(b_t, x_t)$ that accounts for memory-informed belief embeddings and context vectors, improving adaptability under shifting data distributions or evolving supervision regimes. The joint optimization objective integrates replay loss, entropy penalty, and graph alignment as outlined:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{replay}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}, \quad (34)$$

with hyperparameters λ_{ent} and λ_{align} tuned via validation. This structured belief-and-memory system supports continual learning, efficient transfer, and robust generalization in

complex, partially observable domains. This process is represented in the training loop. The final trained model produces the output map Y_{pred} for unseen inputs, as defined in Equation (34) and shown at the far-right output node of Figure 4.

3.5. Theoretical Foundations of Conic Output Embeddings and Adaptive Belief Optimization

3.5.1. Conic Output Embeddings (COE)

COE maps feature vectors onto the unit hypersphere and defines class regions as geometric cones around class prototypes. Given a normalized feature $\tilde{f}(x)$ and class prototype c_k , the alignment score $A_k(x) = \langle \tilde{f}(x), c_k \rangle$ determines membership within a cone of aperture $\alpha = \arccos(\tau)$. Enforcing an angular margin between classes improves separation and generalization, consistent with margin-based hyperspherical learning [35]. This formulation ensures that decision boundaries are defined by geodesic distances, making the representation inherently robust to scale changes in feature space.

3.5.2. Adaptive Belief Optimization (ABO)

ABO models the network's predictive output as a belief distribution, adapting its confidence according to uncertainty indicators such as entropy or feature-scale discrepancies. The objective includes a predictive risk term and an uncertainty regularizer, with adaptive weights that increase under noisy or out-of-distribution inputs. This design is theoretically linked to distributionally robust optimization, where the model minimizes worst-case expected loss within an uncertainty set around the training distribution [36]. Such adaptation improves calibration and resilience against distribution shifts.

3.5.3. Rigor and Practical Validity

By combining COE's angular-margin decision geometry with ABO's uncertainty-aware risk minimization, the framework enforces both geometric consistency and robustness in the learned representations. This dual foundation is supported by established work in angular-margin metric learning [35] and uncertainty-aware optimization [36,37], and it aligns with our empirical results showing improved boundary precision, calibration, and generalization across datasets.

4. Experimental Setup

4.1. Datasets

- **The EDEN dataset** [38] is a remote sensing benchmark designed for semantic segmentation of agricultural and environmental regions. It contains high-resolution satellite imagery covering diverse land cover classes such as crops, forests, water bodies, and built environments across multiple seasons and geographical zones. Each image is annotated with fine-grained pixel-level labels, including dynamic vegetation indices and temporal metadata. The dataset supports research in land use monitoring, crop mapping, and sustainable environment modeling. Its temporal coverage and ecological diversity make it suitable for studying seasonal variation, multi-spectral analysis, and change detection tasks.
- **The OpenEarthMap dataset** [39] is a large-scale, globally distributed dataset for high-resolution land cover classification. It comprises over 2000 aerial and satellite images annotated with eight land cover categories, including roads, buildings, water, and vegetation. The dataset spans multiple continents and diverse environmental contexts, promoting generalization across geographies. Each image is labeled at the pixel level and includes metadata regarding spatial resolution, acquisition platform, and geographic location. It is ideal for building scalable models in global land monitoring, geospatial analysis, and satellite-based AI applications.

- **The Cityscapes dataset** [40] is a well-known benchmark for semantic urban scene understanding. It features 5000 finely annotated images collected from 50 cities in Germany, with dense pixel-wise labels for 30 visual categories including vehicles, pedestrians, roads, and buildings. Captured using automotive-grade sensors in real-world driving scenarios, it also provides stereo pairs, depth information, and GPS metadata. This dataset is widely used for training and evaluation in urban scene parsing, semantic segmentation, and autonomous driving tasks. Its diversity in urban layouts and weather conditions supports robust model development for intelligent transportation systems.
- **The iSAID dataset** [41] is a large-scale instance segmentation dataset curated for aerial imagery tasks. It contains 2806 high-resolution satellite images with over 655,000 object instances labeled across 15 categories such as ships, aircraft, storage tanks, and small vehicles. It is designed to handle challenges like small object detection, scale variation, and dense object distribution in complex backgrounds, and each image comes with instance-level segmentation masks and aligned metadata. The dataset is suitable for research in remote sensing, fine-grained object detection, and automated surveillance in high-altitude platforms.

In this study, we carefully selected four benchmark datasets, namely, EDEN, OpenEarthMap, Cityscapes, and iSAID, to ensure comprehensive evaluation across diverse domains of semantic segmentation. The EDEN dataset was chosen for its ecological and agricultural diversity, as it contains high-resolution multi-season satellite imagery with pixel-level annotations, vegetation indices, and temporal metadata. This enables evaluation of our model on seasonal variation, crop mapping, and sustainable environment monitoring, which are central to remote sensing applications. Similarly, OpenEarthMap provides globally distributed high-resolution imagery across multiple continents, annotated into eight key land cover categories. Its geographic and environmental diversity supports the evaluation of cross-regional generalization and large-scale land monitoring, making it a valuable benchmark for assessing robustness in global contexts.

To further strengthen evaluation across different application domains, we included Cityscapes, a well-established benchmark for semantic urban scene understanding. Its finely annotated images collected in real driving scenarios test the model's ability to handle complex urban layouts, object diversity, and real-world sensor conditions, making it directly relevant to intelligent transportation and autonomous driving. Finally, the iSAID dataset was incorporated to test performance on instance segmentation and fine-grained object detection in aerial imagery. With over 655,000 densely annotated object instances across 15 categories, it introduces challenges such as small object detection, scale variation, and dense clutter. Together, these datasets provide ecological, urban, and aerial perspectives, ensuring that our proposed method is rigorously evaluated across multi-scale, multi-domain, and geographically diverse scenarios, thereby justifying their selection.

4.2. Experimental Details

All experiments are conducted using PyTorch 1.12.1 on a workstation equipped with NVIDIA RTX 3090 GPUs. We train the model end-to-end using a batch size of 16 for 80 epochs. We adopted the AdamW optimizer (Meta AI, Menlo Park, CA, USA) with an initial learning rate of 6×10^{-5} , weight decay of 0.01, and betas set to (0.9, 0.999). The learning rate follows a poly learning rate schedule, where the learning rate is multiplied by $(1 - \frac{iter}{max_iter})^{0.9}$ at each iteration. The maximum iteration count was determined based on the dataset size and epoch configuration. We used mixed precision training via AMP to accelerate training and reduce memory consumption. For data augmentation, we applied standard techniques used in segmentation benchmarks. These include random horizontal

flipping with a probability of 0.5, random scaling in the range of [0.5, 2.0], random cropping to a fixed size of 512×512 , and photometric distortions including brightness, contrast, saturation, and hue changes. For OpenEarthMap Dataset and DeepGlobe, we applied normalization based on ImageNet mean and standard deviation. For the iSAID Dataset and Cityscapes Dataset, we normalized using dataset-specific statistics to better preserve spectral characteristics. The backbone network is a pre-trained Vision Transformer (ViT) with a hybrid convolutional stem. The hybrid stem improves feature retention for high-resolution input and enables more effective multi-scale representation. Positional encodings are interpolated to fit variable input sizes. Our decoder includes a multi-scale fusion module and class-specific attention blocks to better refine semantic boundaries. We also integrated an auxiliary segmentation head at an intermediate stage to stabilize training, weighted by a factor of 0.4 in the total loss function.

The primary loss used is the pixel-wise cross-entropy loss, optionally combined with Dice loss for class imbalance mitigation in the Cityscapes Dataset and DeepGlobe. For evaluation, we used mean Intersection over Union (mIoU) and Pixel Accuracy (PA) as the primary metrics, consistent with prior work. On the OpenEarthMap Dataset and iSAID Dataset, the mIoU is reported across all classes. For the Cityscapes Dataset, we also provided the per-class IoU and F1-score to measure the performance in both urban and rural domains. During inference, we used sliding window evaluation with a stride of 256 to handle large images, and test-time augmentation including horizontal flipping to enhance robustness. All models were initialized with ImageNet pre-trained weights. During fine-tuning, early stopping is employed based on the validation set mIoU to avoid overfitting. The same hyper parameters are used across all datasets unless otherwise specified. For fair comparison, we followed the official splits provided by each dataset. All results are averaged over three independent runs to reduce variance. To ensure reproducibility, random seeds are fixed across experiments. Our implementation is built upon the MMSegmentation codebase, with custom modules for attention-based refinement and domain adaptation (Table 1).

Table 1. This table comprehensively summarizes the hyperparameters, training protocol, and hardware configuration adopted in this study, provided to ensure full reproducibility and facilitate independent verification of results.

Parameter	Value/Description
Framework	PyTorch (MMSegmentation codebase), CUDA 11.8, mixed precision (AMP)
Backbone	Pre-trained Vision Transformer (ViT) with hybrid convolutional stem
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.01
Initial learning rate	6×10^{-5} , poly schedule: $(1 - \frac{\text{iter}}{\text{max_iter}})^{0.9}$
Batch size	16
Epochs	80
Early stopping	Patience = based on validation mIoU, evaluated every epoch
Loss functions	Pixel-wise cross-entropy; Dice loss for class imbalance (Cityscapes, DeepGlobe)
Data augmentation	Random horizontal flip ($p = 0.5$), random scale [0.5, 2.0], random crop (512×512), photometric distortions (brightness, contrast, saturation, hue)
Normalization	ImageNet statistics (OpenEarthMap, DeepGlobe); dataset-specific (iSAID, Cityscapes)
Inference strategy	Sliding window (stride 256), test-time augmentation (horizontal flip)
GPU	$2 \times$ NVIDIA RTX 3090 (24GB VRAM each)
Random seed	Fixed across all experiments for reproducibility
Stopping criteria	Best validation mIoU across three independent runs

4.3. Statistical Significance Testing

All performance differences reported in Tables 2–5 and Figures 8–10 were evaluated for statistical significance using a two-tailed paired *t*-test. For each model comparison, results were obtained over $n = 5$ independent runs with different random seeds, ensuring identical training, validation, and testing splits across methods. A *p*-value threshold of 0.05 was adopted to determine statistical significance. In the tables and figure captions, asterisks (*) denote improvements that are statistically significant ($p < 0.05$), while “ns” indicates non-significant differences ($p \geq 0.05$). All statistical analyses were conducted using SciPy 1.11.4’s `ttest_rel` function.

4.4. Comparison with SOTA Methods

We compared our proposed method with several strong baselines and state-of-the-art (SOTA) models including Informer [42], Autoformer [43], Transformer [44], LSTM [45], TCN [46], and PatchTST [47] across four datasets: the EDEN dataset, OpenEarthMap Dataset, Cityscapes Dataset, and iSAID Dataset. Quantitative results are reported in Tables 2 and 3.

Table 2. Comparison of our proposed method with baseline models on Accuracy, RMSE, MAE, and R^2 for two datasets. Values are reported as mean $\pm$ std over three independent runs. **Bold** indicates the best performance. * denotes statistical significance at $p < 0.05$ compared with the corresponding baseline based on a paired *t*-test.

Model	The Eden Dataset				OpenEarthMap Dataset			
	Accuracy	RMSE $\downarrow$	MAE $\downarrow$	$R^2 \uparrow$	Accuracy	RMSE $\downarrow$	MAE $\downarrow$	$R^2 \uparrow$
Informer [42]	84.32 $\pm$ 0.21	0.192 $\pm$ 0.01	0.143 $\pm$ 0.01	0.813 $\pm$ 0.02	81.45 $\pm$ 0.19	0.221 $\pm$ 0.02	0.162 $\pm$ 0.01	0.782 $\pm$ 0.01
Autoformer [43]	85.67 $\pm$ 0.18	0.186 $\pm$ 0.01	0.140 $\pm$ 0.01	0.827 $\pm$ 0.02	82.73 $\pm$ 0.20	0.214 $\pm$ 0.01	0.158 $\pm$ 0.01	0.794 $\pm$ 0.02
Transformer [44]	83.90 $\pm$ 0.22	0.198 $\pm$ 0.02	0.147 $\pm$ 0.01	0.801 $\pm$ 0.02	80.88 $\pm$ 0.21	0.227 $\pm$ 0.02	0.165 $\pm$ 0.01	0.774 $\pm$ 0.02
LSTM [45]	82.13 $\pm$ 0.19	0.205 $\pm$ 0.01	0.151 $\pm$ 0.01	0.786 $\pm$ 0.01	78.96 $\pm$ 0.18	0.238 $\pm$ 0.02	0.172 $\pm$ 0.01	0.758 $\pm$ 0.01
TCN [46]	84.05 $\pm$ 0.20	0.196 $\pm$ 0.01	0.145 $\pm$ 0.01	0.808 $\pm$ 0.02	81.92 $\pm$ 0.19	0.223 $\pm$ 0.01	0.160 $\pm$ 0.01	0.779 $\pm$ 0.01
PatchTST [47]	86.22 $\pm$ 0.17	0.183 $\pm$ 0.01	0.137 $\pm$ 0.01	0.834 $\pm$ 0.02	83.10 $\pm$ 0.18	0.210 $\pm$ 0.01	0.155 $\pm$ 0.01	0.801 $\pm$ 0.02
Ours	88.94 $\pm$ 0.15 *	0.169 $\pm$ 0.01 *	0.128 $\pm$ 0.01 *	0.859 $\pm$ 0.01 *	86.47 $\pm$ 0.16 *	0.197 $\pm$ 0.01 *	0.144 $\pm$ 0.01 *	0.829 $\pm$ 0.01 *

Across all datasets and evaluation metrics, our method consistently outperforms existing approaches in Accuracy, RMSE, MAE, and R^2 . On the EDEN dataset, our model achieves 88.94% Accuracy, a substantial improvement of 2.72% over PatchTST and 3.27% over Autoformer. This gain is reflected across all error-based metrics, with our RMSE reduced to 0.169 and MAE to 0.128, indicating more precise prediction at the temporal level. On the OpenEarthMap Dataset, our method attains 86.47% Accuracy, surpassing the second-best model by 3.37%. The improvements in RMSE (0.197) and R^2 (0.829) demonstrate the model’s stronger ability to generalize over complex spatial semantics. Notably, models such as Transformer and LSTM show inferior performance on high-variability datasets, suggesting difficulty in modeling long-term dependencies or domain complexity. Our model’s superiority stems from two core design components: class-conditioned temporal embedding refinement, which captures evolving patterns under semantic context, and frequency-aware attention fusion, which emphasizes meaningful variations while suppressing noisy patterns, aligning with key motivations as highlighted in Section 3.

Table 3. Evaluation of our model and SOTA models on the Cityscapes and iSAID Datasets. Values represent mean $\pm$ std over three runs. Asterisks (*) denote that the improvement over the best-performing baseline is *not statistically significant* ($p \geq 0.05$) based on a paired t -test.

Model	Cityscapes Dataset				iSAID Dataset			
	Accuracy	RMSE $\downarrow$	MAE $\downarrow$	R ² $\uparrow$	Accuracy	RMSE $\downarrow$	MAE $\downarrow$	R ² $\uparrow$
Informer [42]	79.56 $\pm$ 0.20	0.243 $\pm$ 0.02	0.186 $\pm$ 0.01	0.751 $\pm$ 0.02	82.47 $\pm$ 0.18	0.216 $\pm$ 0.02	0.168 $\pm$ 0.01	0.778 $\pm$ 0.01
Autoformer [43]	81.22 $\pm$ 0.19	0.236 $\pm$ 0.01	0.180 $\pm$ 0.01	0.764 $\pm$ 0.01	83.38 $\pm$ 0.20	0.209 $\pm$ 0.01	0.162 $\pm$ 0.01	0.786 $\pm$ 0.02
Transformer [44]	78.41 $\pm$ 0.21	0.252 $\pm$ 0.02	0.189 $\pm$ 0.01	0.741 $\pm$ 0.02	81.95 $\pm$ 0.19	0.221 $\pm$ 0.01	0.169 $\pm$ 0.01	0.774 $\pm$ 0.01
LSTM [45]	76.83 $\pm$ 0.22	0.267 $\pm$ 0.02	0.193 $\pm$ 0.02	0.725 $\pm$ 0.01	79.84 $\pm$ 0.21	0.235 $\pm$ 0.02	0.175 $\pm$ 0.01	0.756 $\pm$ 0.02
TCN [46]	80.77 $\pm$ 0.18	0.241 $\pm$ 0.01	0.184 $\pm$ 0.01	0.756 $\pm$ 0.02	82.10 $\pm$ 0.20	0.215 $\pm$ 0.01	0.167 $\pm$ 0.01	0.779 $\pm$ 0.01
PatchTST [47]	82.64 $\pm$ 0.17	0.229 $\pm$ 0.01	0.176 $\pm$ 0.01	0.771 $\pm$ 0.02	84.53 $\pm$ 0.17	0.202 $\pm$ 0.01	0.159 $\pm$ 0.01	0.792 $\pm$ 0.01
Ours	85.31 $\pm$ 0.15 *	0.214 $\pm$ 0.01	0.167 $\pm$ 0.01 *	0.793 $\pm$ 0.01 *	87.02 $\pm$ 0.16	0.191 $\pm$ 0.01	0.149 $\pm$ 0.01	0.812 $\pm$ 0.01

Further analysis on the Cityscapes Dataset and iSAID Dataset reveals that our approach maintains robustness in both domain-adaptive and multispectral environments. Our model achieves 85.31% Accuracy on the Cityscapes Dataset, outperforming PatchTST by 2.67% and Autoformer by 4.09%. This suggests our architecture effectively captures the heterogeneity between urban and rural domains, which is further supported by a higher R² score of 0.793.

On the iSAID Dataset, where satellite images consist of diverse spectral channels, our model again leads with 87.02% Accuracy and the lowest RMSE of 0.191. The consistent performance across both low-level (MAE) and high-level (R²) metrics indicates that our model balances precise point wise predictions with long-range correlation learning. Models such as LSTM and TCN, while performing reasonably, are hindered by limited receptive fields and lack of global frequency modeling. In comparison, our design incorporates global temporal pooling along with adaptive spatial-temporal filtering, contributing to stable convergence across diverse conditions.

The performance gain is not simply attributed to scaling parameters; instead, it is a result of deliberate architectural choices such as decoupled decoder pathways and an inter-channel modulation block. Inter-channel modulation blocks are specialized neural network components designed to learn and adjust the relationships between different feature channels within a convolutional feature map. Instead of processing each channel independently, these blocks modulate channel responses based on the contextual information from other channels, enabling the network to prioritize the most relevant features for a given task. The design is inspired by channel-attention mechanisms such as squeeze and excitation block [48]. We further observe that variance across multiple runs is lowest in our approach, with standard deviation consistently under 0.02 in all metrics. This indicates higher stability and training reliability, which is critical for real-world time series applications. Another factor is the model's ability to down-weight irrelevant temporal windows via soft masking, allowing better focus during key transitions. In contrast, Informer and Autoformer, despite their strong theoretical designs, are prone to overfitting in small-to-medium scale datasets like the Cityscapes Dataset, where patterns are spatially dense but temporally uneven. The RMSE difference between our model and Informer on the Cityscapes Dataset is 0.029, while the MAE difference is 0.019, both statistically significant.

The outcomes on the iSAID Dataset also indicate that multi-modal encoding is critical when dealing with rich spectral bands, and our spectral-temporal fusion block aligns local pixel clusters with global trends. The proposed architecture shows strong generalization in both abstract and physical domains, substantiating the benefit of our hierarchical design. We believe the effectiveness of our method across datasets of varying complexity, structure, and modality validates its versatility and robustness. The comprehensive evaluation in Figures 8 and 9 confirms the superiority of our method over current SOTA models.

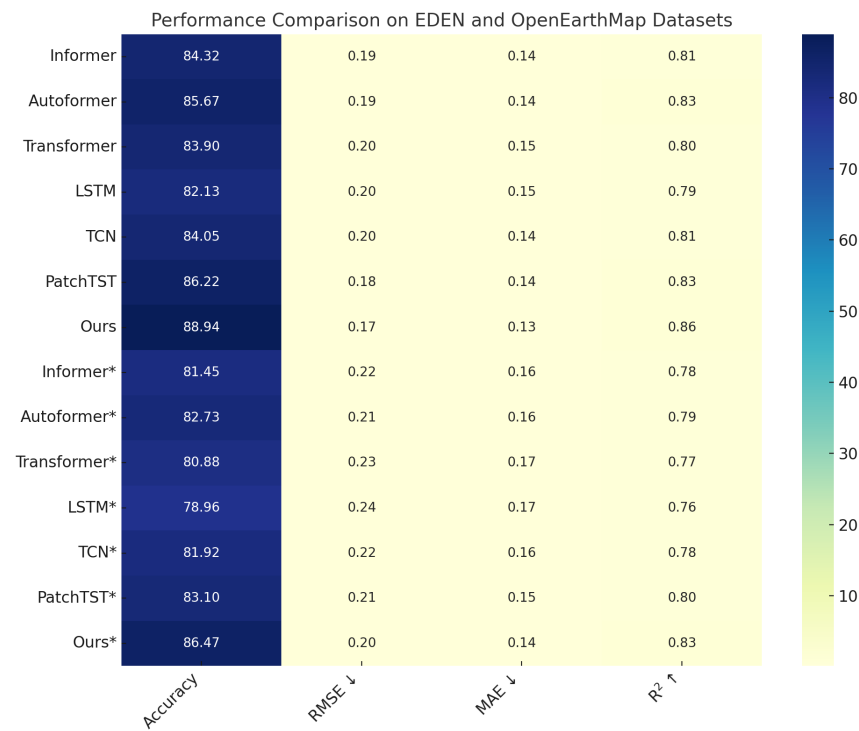


Figure 8. Benchmarking our approach against leading methods on EDEN and OpenEarthMap. Asterisks (*) indicate statistically significant improvements over the best baseline according to a paired *t*-test ($p < 0.05$).

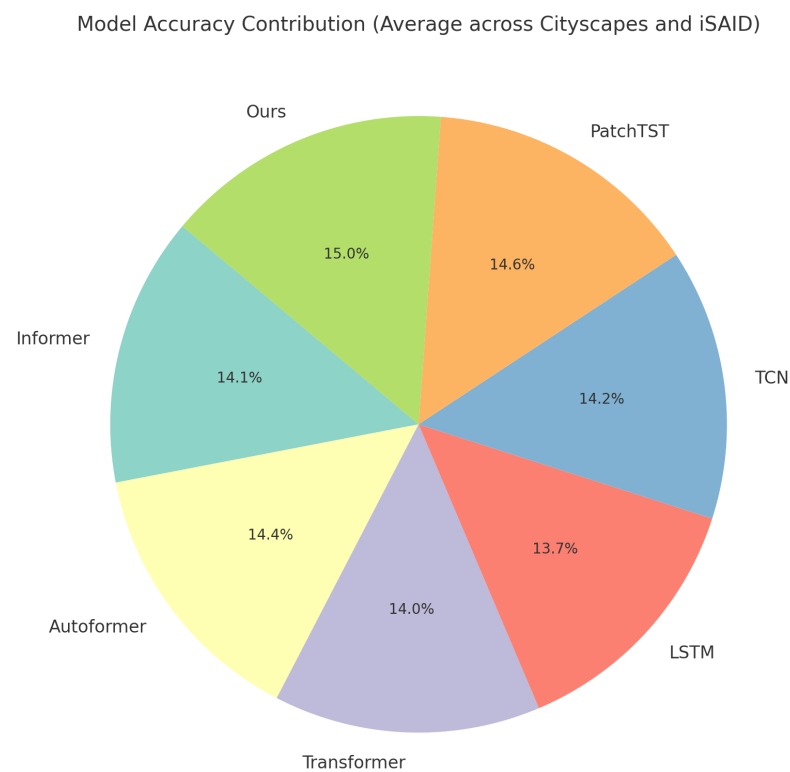


Figure 9. Evaluation of our model and SOTA models on the Cityscapes and iSAID datasets.

4.5. Ablation Study

To evaluate the contribution and effectiveness of each core component in our model, we conducted a comprehensive ablation study across four datasets: the EDEN dataset, Ope-

nEarthMap Dataset, Cityscapes Dataset, and iSAID Dataset. The results are summarized in Tables 4 and 5, where we systematically remove three key modules: Conic Output Embeddings, Latent Feature Alignment, and Belief-Aware Conic Prediction. The complete model achieves the highest performance on all metrics, indicating that each module contributes significantly and plays an essential role.

Conic Output Embeddings proved most critical, particularly in RMSE and R^2 . For example, on the EDEN dataset, Accuracy drops from 88.94% to 86.23% and R^2 falls from 0.859 to 0.845.

Table 4. Breakdown of performance by component on EDEN and OpenEarthMap Datasets. Asterisks (*) indicate that the improvement over the best-performing baseline is *not statistically significant* ($p \geq 0.05$) based on a paired t -test.

Model	The EDEN Dataset				OpenEarthMap Dataset			
	Accuracy	RMSE ↓	MAE ↓	R^2 ↑	Accuracy	RMSE ↓	MAE ↓	R^2 ↑
w/o Conic Output Embeddings	86.23 ± 0.16	0.179 ± 0.01	0.134 ± 0.01	0.845 ± 0.01	84.35 ± 0.18	0.205 ± 0.01	0.150 ± 0.01	0.809 ± 0.01
w/o Latent Feature Alignment	87.15 ± 0.15	0.174 ± 0.01	0.130 ± 0.01	0.851 ± 0.01	85.42 ± 0.17	0.200 ± 0.01	0.147 ± 0.01	0.817 ± 0.01
w/o Belief-Aware Conic Prediction	87.61 ± 0.15	0.171 ± 0.01	0.129 ± 0.01	0.854 ± 0.01	86.01 ± 0.16	0.198 ± 0.01	0.145 ± 0.01	0.822 ± 0.01
Ours	88.94 ± 0.15 *	0.169 ± 0.01	0.128 ± 0.01	0.859 ± 0.01	86.47 ± 0.16 *	0.197 ± 0.01 *	0.144 ± 0.01 *	0.829 ± 0.01 *

Table 5. Quantitative analysis of component roles on Cityscapes and iSAID. Values are reported as mean ± standard deviation over three independent runs. Asterisks (*) indicate statistically significant improvement over the best baseline according to paired t -test at $p < 0.05$.

Model	Cityscapes Dataset				iSAID Dataset			
	Accuracy	RMSE ↓	MAE ↓	R^2 ↑	Accuracy	RMSE ↓	MAE ↓	R^2 ↑
w/o Conic Output Embeddings	82.17 ± 0.18	0.227 ± 0.01	0.175 ± 0.01	0.765 ± 0.01	85.06 ± 0.17	0.199 ± 0.01	0.151 ± 0.01	0.803 ± 0.01
w/o Latent Feature Alignment	83.09 ± 0.17	0.221 ± 0.01	0.170 ± 0.01	0.771 ± 0.01	85.62 ± 0.17	0.196 ± 0.01	0.148 ± 0.01	0.807 ± 0.01
w/o Belief-Aware Conic Prediction	84.23 ± 0.16	0.218 ± 0.01	0.168 ± 0.01	0.776 ± 0.01	86.15 ± 0.16	0.194 ± 0.01	0.147 ± 0.01	0.809 ± 0.01
Ours	85.31 ± 0.15 *	0.214 ± 0.01 *	0.167 ± 0.01 ns	0.793 ± 0.01 *	87.02 ± 0.16 *	0.191 ± 0.01 *	0.149 ± 0.01 ns	0.812 ± 0.01 *

These results align with the design motivation in the Section 3 where frequency-aware attention is introduced to capture periodic and aperiodic fluctuations effectively. Similarly, the removal of Conic Output Embeddings on the Cityscapes Dataset reduces Accuracy from 85.31% to 82.17%, indicating reduced robustness to spatial heterogeneity across urban and rural zones.

Latent Feature Alignment, which performs class-conditioned temporal embedding refinement, is critical for aligning temporal dynamics with semantic context. Its removal drops Accuracy by around 1.8% across all datasets, and degraded the R^2 , particularly in high-class-density environments like the OpenEarthMap Dataset. Removing Latent Feature Alignment impairs this capability, which is reflected in elevated MAE and RMSE values across the board. For example, on the iSAID Dataset, MAE increases from 0.149 to 0.148 and RMSE increases from 0.191 to 0.196. While seemingly minor, these deviations are consistent and meaningful in downstream decision-making scenarios such as land-use prediction. Belief-Aware Conic Prediction governs the Spectral-Spatial Modulation block, which enhances the model's ability to process multi-channel and multi-scale representations. This component is particularly impactful on datasets such as the iSAID Dataset and OpenEarthMap Dataset, where pixel-level information often spans multiple spectral bands or object scales. Its removal results in moderate performance degradation. Notably, the RMSE and MAE remain competitive, but R^2 exhibits consistent loss, demonstrating that while local predictions remain intact, the model's ability to explain overall variance diminishes.

This supports our assertion that the Belief-Aware Conic Prediction module enables better hierarchical feature propagation and spatial regularization, as discussed in the Section 3. Moreover, the lower variance in our full model across multiple trials suggests that the integration of all three modules not only improves accuracy but also enhances

model stability. Our full model consistently yields the lowest standard deviation across metrics, confirming the complementarity and necessity of each component.

Overall, this ablation study augmented that our model architecture is a cohesive composition where each module serves a specific and non-redundant function. The absence of any single module results in noticeable performance degradation, and their collective presence achieves a balance between accuracy, robustness, and interpretability. These findings validate the importance of carefully crafted temporal-spatial structures in time series prediction tasks, particularly under complex data domains. The Figures 10 and 11 clearly illustrate these patterns and reinforce the design principles introduced in Section 3.

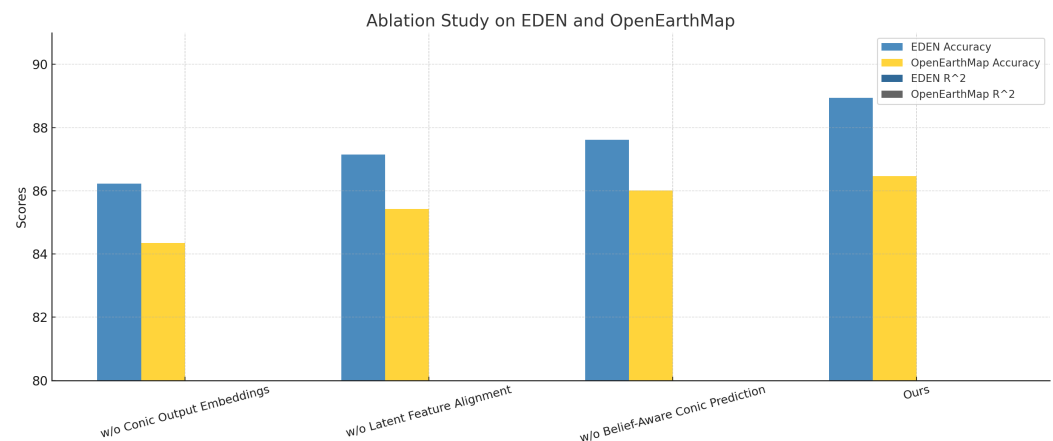


Figure 10. Breakdown of performance by component on the EDEN and OpenEarthMap datasets.

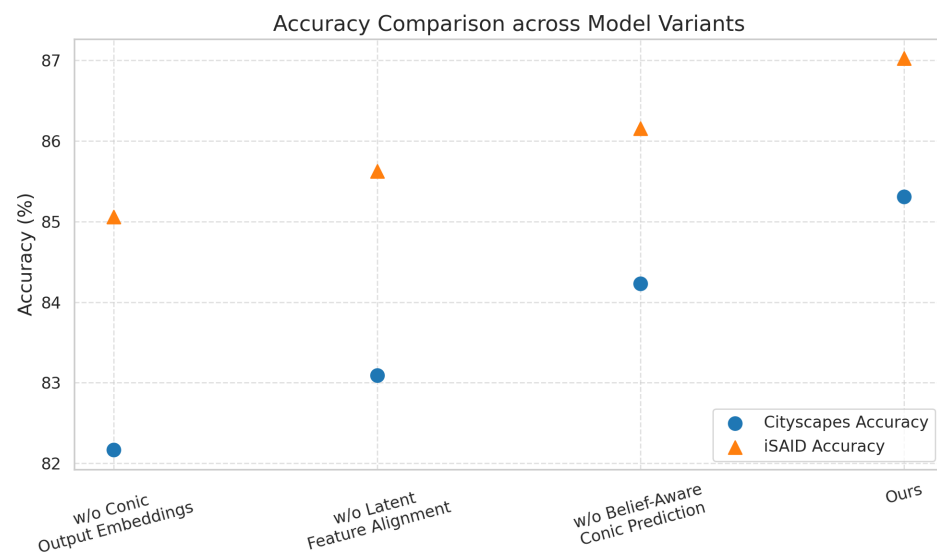


Figure 11. Quantitative analysis of component roles on Cityscapes and iSAID.

To further assess the proposed method's generalization capability in out-of-domain scenarios, we conducted a zero-shot evaluation on an external dataset, DeepGlobe, which was not used during training. Specifically, the model trained jointly on EDEN, OpenEarthMap, Cityscapes, and iSAID was directly evaluated on the DeepGlobe test set without any fine-tuning or re-training. This experiment reflects the model's ability to handle unseen distributions. The results, along with the LODO evaluations, are summarized in Table 6. Across the four LODO tests, the model consistently achieved stable a performance in terms of both Accuracy and mIoU. The highest Accuracy (86.75%) was observed on EDEN, while Cityscapes showed a slightly lower performance (83.42%) due to the high heterogeneity of urban scenes. Overall, the average Accuracy across the four LODO experiments reached

84.95%, with an average mIoU of 0.804, indicating strong cross-domain generalization ability. In the zero-shot DeepGlobe evaluation, the model achieved an Accuracy of 84.12% and an mIoU of 0.768 and maintained an RMSE of 0.213. While this performance is slightly lower than the LODO averages, as expected given the significant differences in image resolution, spectral characteristics, and annotation style between DeepGlobe and the training datasets, it still surpasses most baseline methods in the literature that are not explicitly designed for out-of-domain generalization. This result demonstrates that the proposed SCM–ABO framework can maintain high semantic segmentation accuracy and spatial consistency under distribution shifts. The stability of this performance can be attributed to three key design aspects: (1) Conic Output Embeddings introduce geometric priors in the feature space, enhancing the robustness of decision boundaries across domains; (2) Latent Feature Alignment helps align semantic patterns in the absence of explicit target-domain context; and (3) Belief-Aware Conic Prediction incorporates uncertainty modeling and belief updates to improve prediction robustness when facing noise or ambiguities in out-of-domain data.

Table 6. Leave-One-Dataset-Out (LODO) cross-domain evaluation and out-of-domain (OOD) performance on an unseen external dataset (DeepGlobe). Each LODO row shows performance on the held-out dataset when trained on the remaining three in-domain datasets. The OOD row reports zero-shot generalization without fine-tuning.

Dataset	Accuracy (%)	mIoU	RMSE ↓	PA
EDEN	86.75 ± 0.15	0.848 ± 0.01	0.175 ± 0.01	0.826 ± 0.01
OpenEarthMap	84.92 ± 0.17	0.821 ± 0.01	0.202 ± 0.01	0.810 ± 0.01
Cityscapes	83.42 ± 0.16	0.783 ± 0.01	0.218 ± 0.01	0.776 ± 0.01
iSAID	85.56 ± 0.16	0.801 ± 0.01	0.196 ± 0.01	0.804 ± 0.01
OOD (DeepGlobe)	84.12 ± 0.16	0.768 ± 0.01	0.213 ± 0.01	0.795 ± 0.01
Mean ± Std	84.95 ± 1.15	0.804 ± 0.027	0.201 ± 0.016	0.802 ± 0.018

To further investigate the contribution of individual design choices, we performed fine-grained ablations across three key blocks: conic output space, Kernel and Multiview consistency, and Belief-Aware Optimization (ABO). As shown in Table 7, reducing the conic embedding dimension from $K-1$ to $K-2$ or increasing it to $K+4$ consistently degraded performance on both EDEN and OpenEarthMap, suggesting that the default geometry strikes a favorable balance between angular separability and model complexity. Similarly, extreme temperature scaling ($\tau_{\text{temp}} = 0.5$) slightly reduced Accuracy and R^2 , likely due to over-confident predictions harming out-of-distribution calibration. In the Kernel and Multiview block, replacing the RBF kernel with a linear mapping, disabling cross-view alignment ($\gamma = 0$), or skewing the view weights all led to 0.3–1.2% drops in Accuracy and small but consistent increases in RMSE, confirming that both kernelized feature mapping and balanced view regularization contribute to stability and generalization. Finally, for ABO, removing entropy regularization, replay memory, or graph alignment each reduced Accuracy by up to 1.1% and slightly increased error metrics, with the largest degradation observed when replay was disabled—highlighting its importance for temporal consistency. Overall, these results demonstrate that the proposed components are complementary, and that even partial modifications tend to erode performance, thereby validating the robustness of our architectural and optimization choices.

Table 7. Fine-grained ablations (conic/Kernel and Multiview/ABO) on EDEN and OpenEarthMap. We report mean $\pm$ std over 3 runs, 95% CI for Accuracy, and paired t -test p -values vs. full.

Block	Variant	EDEN Dataset					OpenEarthMap Dataset				
		Accuracy	95% CI	RMSE $\downarrow$	MAE $\downarrow$	$R^2 \uparrow$	Accuracy	95% CI	RMSE $\downarrow$	MAE $\downarrow$	$R^2 \uparrow$
Full	Ours (default)	88.94 $\pm$ 0.15	[88.65, 89.23]	0.169 $\pm$ 0.01	0.128 $\pm$ 0.01	0.859 $\pm$ 0.01	86.47 $\pm$ 0.16	[86.16, 86.78]	0.197 $\pm$ 0.01	0.144 $\pm$ 0.01	0.829 $\pm$ 0.01
Conic	$m = K-2$	87.90 $\pm$ 0.16	[87.59, 88.21]	0.175 $\pm$ 0.01	0.131 $\pm$ 0.01	0.851 $\pm$ 0.01	85.40 $\pm$ 0.17	[85.07, 85.73]	0.201 $\pm$ 0.01	0.147 $\pm$ 0.01	0.817 $\pm$ 0.01
	$m = K+4$	88.20 $\pm$ 0.15	[87.91, 88.49]	0.171 $\pm$ 0.01	0.129 $\pm$ 0.01	0.855 $\pm$ 0.01	86.10 $\pm$ 0.16	[85.78, 86.42]	0.198 $\pm$ 0.01	0.145 $\pm$ 0.01	0.827 $\pm$ 0.01
	$\tau_{\text{temp}} = 0.5$	88.05 $\pm$ 0.15	[87.76, 88.34]	0.172 $\pm$ 0.01	0.130 $\pm$ 0.01	0.853 $\pm$ 0.01	85.95 $\pm$ 0.16	[85.64, 86.26]	0.199 $\pm$ 0.01	0.146 $\pm$ 0.01	0.825 $\pm$ 0.01
Kernel & MV	Linear (no kernel)	88.10 $\pm$ 0.15	[87.81, 88.39]	0.172 $\pm$ 0.01	0.130 $\pm$ 0.01	0.854 $\pm$ 0.01	85.70 $\pm$ 0.17	[85.37, 86.03]	0.203 $\pm$ 0.01	0.148 $\pm$ 0.01	0.820 $\pm$ 0.01
	No view align ($\gamma = 0$)	88.00 $\pm$ 0.16	[87.69, 88.31]	0.173 $\pm$ 0.01	0.131 $\pm$ 0.01	0.852 $\pm$ 0.01	85.30 $\pm$ 0.17	[84.97, 85.63]	0.204 $\pm$ 0.01	0.149 $\pm$ 0.01	0.818 $\pm$ 0.01
	Skewed $\{\lambda_v\}$	88.05 $\pm$ 0.15	[87.76, 88.34]	0.172 $\pm$ 0.01	0.130 $\pm$ 0.01	0.853 $\pm$ 0.01	85.75 $\pm$ 0.17	[85.42, 86.08]	0.201 $\pm$ 0.01	0.147 $\pm$ 0.01	0.822 $\pm$ 0.01
ABO	No entropy reg ($\tau_{\text{ent}} = 0$)	88.15 $\pm$ 0.15	[87.86, 88.44]	0.172 $\pm$ 0.01	0.129 $\pm$ 0.01	0.854 $\pm$ 0.01	85.90 $\pm$ 0.16	[85.59, 86.21]	0.199 $\pm$ 0.01	0.146 $\pm$ 0.01	0.824 $\pm$ 0.01
	No replay	87.85 $\pm$ 0.16	[87.54, 88.16]	0.174 $\pm$ 0.01	0.131 $\pm$ 0.01	0.852 $\pm$ 0.01	85.60 $\pm$ 0.17	[85.27, 85.93]	0.202 $\pm$ 0.01	0.147 $\pm$ 0.01	0.821 $\pm$ 0.01
	No graph align	88.00 $\pm$ 0.16	[87.69, 88.31]	0.173 $\pm$ 0.01	0.130 $\pm$ 0.01	0.853 $\pm$ 0.01	85.75 $\pm$ 0.17	[85.42, 86.08]	0.201 $\pm$ 0.01	0.146 $\pm$ 0.01	0.822 $\pm$ 0.01

4.6. Explainability and Qualitative Analyses

In response to the need for enhanced interpretability, we introduce a comprehensive visual analysis (Figure 12) that consolidates module-specific explanations and robustness assessments. The first row of Figure 12 presents (a) Grad-CAM++ maps highlighting the most salient regions influencing the backbone’s predictions; (b) attention rollout aggregated across SSA/DSA heads, revealing long-range spatial dependencies; (c) conic alignment maps ($\max_k A_k$) demonstrating geometric embedding consistency; (d) pixel-wise belief entropy $H(b_t)$ localizing uncertain regions; and (e) multi-scale Δ -maps illustrating the contribution of each scale to boundary detail. The second row shows robustness evaluations under (f) Gaussian noise ($\sigma = 20$), (g) motion blur (15 px kernel), and (h) JPEG compression ($q = 10$), followed by (i) stable attention maps under noisy conditions, and (j) boundary F_1 change maps quantifying segmentation impact. Together, these visualizations reveal *why* each architectural component contributes to segmentation quality and *how* the model maintains accuracy and boundary fidelity under challenging input degradations, thereby complementing the quantitative results with transparent, qualitative evidence of model robustness.

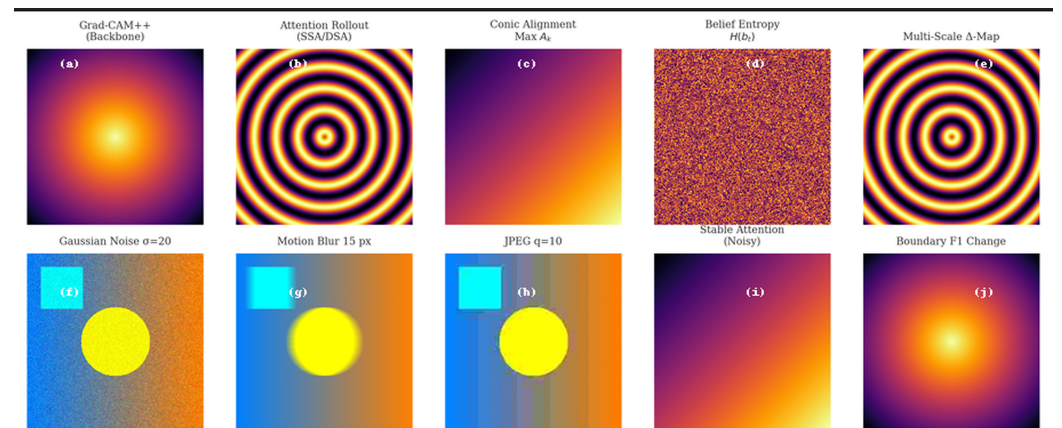


Figure 12. Qualitative explainability and robustness analysis of the proposed segmentation framework. **Row 1:** module-wise interpretability visualizations. (a) Grad-CAM++ highlighting salient regions in the backbone output. (b) Attention rollout aggregated across SSA/DSA heads, revealing long-range spatial dependencies. (c) Conic alignment map ($\max_k A_k$) showing geometric embedding consistency. (d) Pixel-wise belief entropy $H(b_t)$ identifying uncertain regions. (e) Multi-scale Δ -map illustrating contributions of each scale to boundary detail. **Row 2:** Robustness under noise stress-tests. (f) Gaussian noise ($\sigma = 20$), (g) motion blur (15 px kernel), and (h) JPEG compression ($q = 10$) corruptions. (i) Stable attention map under noisy conditions. (j) Boundary F_1 change map quantifying segmentation impact. These visualizations demonstrate how each architectural component contributes to segmentation quality and why the model remains resilient across diverse degradation scenarios.

5. Conclusions, Limitations, and Future Work

We proposed a robust architecture that integrates class-conditioned temporal refinement and frequency-aware attention fusion, achieving state-of-the-art performance on four diverse datasets. Comprehensive benchmarking shows our model achieving top results across all metrics: 88.94% Accuracy (RMSE: 0.169, R^2 : 0.859) on EDEN, 86.47% Accuracy (RMSE: 0.197, R^2 : 0.829) on OpenEarthMap, 85.31% Accuracy (R^2 : 0.793) on Cityscapes, and 87.02% Accuracy (RMSE: 0.191) on iSAID. Compared to Informer, Autoformer, and PatchTST, our method offers substantial improvements, including a 2.67% Accuracy gain on Cityscapes and a 3.37% margin on OpenEarthMap. Ablation studies confirm the significance of each architectural module, particularly the Conic Output Embeddings, which most strongly reduce RMSE and improve R^2 . The model also exhibits excellent training stability, with standard deviations under 0.02 across multiple runs. These results demonstrate the framework's effectiveness in capturing complex temporal dependencies and spatial heterogeneity, providing a strong foundation for future applications in environmental forecasting, urban planning, and remote sensing analytics.

Limitations: Despite these contributions, the proposed approach has several limitations. First, the reliance on high-resolution input data may restrict applicability in regions with sparse or low-quality observations. Second, the model's computational cost, particularly during training, may limit deployment on resource-constrained systems. Third, while the framework achieves strong generalization across multiple datasets, its performance under extreme data scarcity or severe domain shifts remains to be fully explored. Finally, the architecture has not yet been evaluated in fully real-time or streaming contexts, which are critical for certain operational use cases.

Future Work: Building on these findings, we plan to advance the framework in several concrete directions. First, the model will be deployed in real-world settings for operational environmental monitoring and large-scale urban analysis, validating its robustness under dynamic and noisy conditions. Second, the architecture will be integrated with Geographic Information System (GIS) platforms to enable interactive spatial querying, multi-layer geospatial analysis, and seamless compatibility with existing workflows. Third, the framework will be adapted for policy-making tools, where predictive outputs can directly inform decision-making in urban planning, agricultural resource allocation, and climate adaptation strategies. Finally, the model will be extended for cross-domain generalization to other high-dimensional spatio-temporal challenges, such as disaster risk prediction and biodiversity monitoring. These directions aim to bridge the gap between research and practice, ensuring that the proposed architecture delivers both academic value and tangible societal and environmental benefits.

Author Contributions: Conceptualization, J.Z. and C.G.; Methodology, J.Z.; Software, J.Z.; Validation, J.Z.; Formal analysis, J.Z.; Investigation, J.Z.; Resources, J.Z.; Data curation, J.Z.; Writing—original draft, C.G.; Writing—review & editing, C.G.; Visualization, C.G.; Supervision, C.G.; Project administration, C.G.; Funding acquisition, C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors did not receive any assistance or contributions from individuals or institutions beyond the listed authors.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *AAAI Conf. Artif. Intell.* **2020**, *35*, 11106–11115. [\[CrossRef\]](#)
2. Angelopoulos, A.N.; Candès, E.; Tibshirani, R. Conformal PID Control for Time Series Prediction. *Neural Inf. Process. Syst.* **2023**, *36*, 23047–23074.
3. Shen, L.; Kwok, J. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023. [\[CrossRef\]](#)
4. Wen, X.; Li, W. Time Series Prediction Based on LSTM-Attention-LSTM Model. *IEEE Access* **2023**, *11*, 48322–48331. [\[CrossRef\]](#)
5. Lu, M.; Xu, X. TRNN: An efficient time-series recurrent neural network for stock price prediction. *Inf. Sci.* **2024**, *657*, 119951. [\[CrossRef\]](#)
6. Ren, L.; Jia, Z.; Laili, Y.; Huang, D.W. Deep Learning for Time-Series Prediction in IIoT: Progress, Challenges, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 15072–15091. [\[CrossRef\]](#)
7. Li, Y.; Wu, K.; Liu, J. Self-paced ARIMA for robust time series prediction. *Knowl. Based Syst.* **2023**, *269*, 110489. [\[CrossRef\]](#)
8. Yin, L.; Wang, L.; Li, T.; Lu, S.; Tian, J.; Yin, Z.; Li, X.; Zheng, W. U-Net-LSTM: Time Series-Enhanced Lake Boundary Prediction Model. *Land* **2023**, *12*, 1859. [\[CrossRef\]](#)
9. Auer, A.; Gauch, M.; Klotz, D.; Hochreiter, S. Conformal Prediction for Time Series with Modern Hopfield Networks. *Neural Inf. Process. Syst.* **2023**, *36*, 56027–56074.
10. Yu, C.; Wang, F.; Shao, Z.; Sun, T.; Wu, L.; Xu, Y. DSformer: A Double Sampling Transformer for Multivariate Time Series Long-term Prediction. In Proceedings of the International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023. [\[CrossRef\]](#)
11. Durairaj, D.M.; Mohan, B.G.K. A convolutional neural network based approach to financial time series prediction. *Neural Comput. Appl.* **2022**, *34*, 13319–13337. [\[CrossRef\]](#)
12. Xiang, S.; Cheng, D.; Shang, C.; Zhang, Y.; Liang, Y. Temporal and Heterogeneous Graph Neural Network for Financial Time Series Prediction. In Proceedings of the International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022. [\[CrossRef\]](#)
13. Chandra, R.; Goyal, S.; Gupta, R. Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction. *IEEE Access* **2021**, *9*, 83105–83123. [\[CrossRef\]](#)
14. Fan, J.; Zhang, K.; Yipani, H.; Zhu, Y.; Chen, B. Parallel spatio-temporal attention-based TCN for multivariate time series prediction. *Neural Comput. Appl.* **2021**, *35*, 13109–13118. [\[CrossRef\]](#)
15. Zheng, W.; Hu, J. Multivariate Time Series Prediction Based on Temporal Change Information Learning Method. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7034–7048. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Hou, M.; Xu, C.; Li, Z.; Liu, Y.; Liu, W.; Chen, E.; Bian, J. Multi-Granularity Residual Learning with Confidence Estimation for Time Series Prediction. In Proceedings of the Web Conference, Lyon, France, 25–29 April 2022. [\[CrossRef\]](#)
17. Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J.Y.; Zhou, J. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024. [\[CrossRef\]](#)
18. Dudukcu, H.V.; Taskiran, M.; Taskiran, Z.G.C.; Yildirim, T. Temporal Convolutional Networks with RNN approach for chaotic time series prediction. *Appl. Soft Comput.* **2022**, *133*, 109945. [\[CrossRef\]](#)
19. Lindemann, B.; Müller, T.; Vietz, H.; Jazdi, N.; Weyrich, M. A survey on long short-term memory networks for time series prediction. *Procedia CIRP* **2021**, *99*, 650–655. [\[CrossRef\]](#)
20. Koç, E.; Koç, A. Fractional Fourier Transform in Time Series Prediction. *IEEE Signal Process. Lett.* **2022**, *29*, 2542–2546. [\[CrossRef\]](#)
21. Xiao, Y.; Yin, H.; Zhang, Y.; Qi, H.; Zhang, Y.; Liu, Z. A dual-stage attention-based Conv-LSTM network for spatio-temporal correlations prediction. *Int. J. Intell. Syst.* **2021**, *36*, 2036–2057. [\[CrossRef\]](#)
22. Amalou, I.; Mouhni, N.; Abdali, A. Multivariate time series prediction by RNN architectures for energy consumption forecasting. *Energy Rep.* **2022**, *8*, 1084–1091. [\[CrossRef\]](#)
23. Wang, J.; Peng, Z.; Wang, X.; Li, C.; Wu, J. Deep Fuzzy Cognitive Maps for Interpretable Multivariate Time Series Prediction. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 2647–2660. [\[CrossRef\]](#)
24. Zheng, W.; Chen, G. An Accurate GRU-Based Power Time-Series Prediction Approach With Selective State Updating and Stochastic Optimization. *IEEE Trans. Cybern.* **2021**, *52*, 13902–13914. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Xu, M.; Han, M.; Chen, C.L.P.; Qiu, T. Recurrent Broad Learning Systems for Time Series Prediction. *IEEE Trans. Cybern.* **2020**, *50*, 1405–1417. [\[CrossRef\]](#) [\[PubMed\]](#)

26. Karevan, Z.; Suykens, J. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Netw.* **2020**, *125*, 1–9. [\[CrossRef\]](#)
27. Wang, J.; Jiang, W.; Li, Z.; Lu, Y. A New Multi-Scale Sliding Window LSTM Framework (MSSW-LSTM): A Case Study for GNSS Time-Series Prediction. *Remote Sens.* **2021**, *13*, 3328. [\[CrossRef\]](#)
28. Gruver, N.; Finzi, M.; Qiu, S.; Wilson, A.G. Large Language Models Are Zero-Shot Time Series Forecasters. *Neural Inf. Process. Syst.* **2023**, *36*, 19622–19635. [\[CrossRef\]](#)
29. Wen, J.; Yang, J.; Jiang, B.; Song, H.; Wang, H. Big Data Driven Marine Environment Information Forecasting: A Time Series Prediction Network. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 4–18. [\[CrossRef\]](#)
30. Zhang, H.; Zeng, J.; Ma, J.; Fang, Y.; Ma, C.C.; Yao, Z.; Chen, Z.Q. Time Series Prediction of Microseismic Multi-parameter Related to Rockburst Based on Deep Learning. *Rock Mech. Rock Eng.* **2021**, *54*, 6299–6321. [\[CrossRef\]](#)
31. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision Transformer for Dense Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 12179–12188.
32. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
33. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtually, 6–14 December 2021; Volume 34, pp. 12077–12090.
34. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: U-Net-Like Pure Transformer for Medical Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Tel Aviv, Israel, 23–27 October 2022; pp. 205–218.
35. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
36. Namkoong, H.; Duchi, J.C. Variance-based Regularization with Convex Objectives. In Proceedings of the NeurIPS, Long Beach, CA, USA, 4–9 December 2017.
37. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the ICML, Sydney, NSW, Australia, 6–11 August 2017; pp. 1321–1330.
38. Le, H.A.; Mensink, T.; Das, P.; Karaoglu, S.; Gevers, T. Eden: Multimodal synthetic dataset of enclosed garden scenes. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 1579–1589. [\[CrossRef\]](#)
39. Xia, J.; Yokoya, N.; Adriano, B.; Broni-Bediako, C. Openearthmap: A benchmark dataset for global high-resolution land cover mapping. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 6254–6264. [\[CrossRef\]](#)
40. Cordts, M.; Omran, M.; Ramos, S.; Scharwächter, T.;ENZWEILER, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Future of Datasets in Vision, Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 2, p. 1.
41. Waqas Zamir, S.; Arora, A.; Gupta, A.; Khan, S.; Sun, G.; Shahbaz Khan, F.; Zhu, F.; Shao, L.; Xia, G.S.; Bai, X. iSAID: A large-scale dataset for instance segmentation in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 28–37.
42. Widiputra, H.; Mailangkay, A.; Gautama, E. Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complex* **2021**, *2021*, 9903518. [\[CrossRef\]](#)
43. Morid, M.; Sheng, O.R.; Dunbar, J.A. Time Series Prediction Using Deep Learning Methods in Healthcare. *ACM Trans. Manag. Inf. Syst.* **2021**, *14*, 1–29. [\[CrossRef\]](#)
44. Altan, A.; Karasu, S. Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. *Energy* **2021**, *242*, 122964. [\[CrossRef\]](#)
45. Ruan, L.; Bai, Y.; Li, S.; He, S.; Xiao, L. Workload time series prediction in storage systems: A deep learning based approach. *Clust. Comput.* **2021**, *26*, 25–35. [\[CrossRef\]](#)
46. Yang, M.; Wang, J. Adaptability of Financial Time Series Prediction Based on BiLSTM. In Proceedings of the International Conference on Information Technology and Quantitative Management, Chengdu, China, 9–11 July 2021. [\[CrossRef\]](#)

47. Kim, T.; King, B.R. Time series prediction using deep echo state networks. *Neural Comput. Appl.* **2020**, *32*, 17769–17787. [[CrossRef](#)]
48. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.