Neurocomputing 700 (2026) 134509

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey paper

Leveraging autoencoders for GeoAI: A survey of methods and applications

Andreas Karathanasis [a], John Violos [b,*], Iraklis Varlamis [a], Aris Leivadeas [c], Konstantinos Tserpes [d]

[a] Dept. of Informatics & Telematics, Harokopio University of Athens, Omirou 9, Tavros, 17778, Greece
[b] Dept. of Geography, Harokopio University of Athens, El. Venizelou Ave. 70, Kallithea, 17676, Greece
[c] Dept. of Software & Information Technology Engineering, École de Technologie Supérieure, 1100 Notre-Dame St W, Montreal, H3C1K3, Quebec, Canada
[d] School of Electrical and Computer Engineering, National Technical University of Athens, Heroon Polytechniou 9, Zografou, 15773, Greece

HIGHLIGHTS

- Comprehensive survey of autoencoders for GeoAI trajectory analysis.
- Unified taxonomy across architectures, tasks, domains, and datasets.
- Latent representations as a unifying basis for trajectory tasks.
- Applications: urban mobility, maritime, aviation, and interaction systems.
- Challenges: scalability, privacy, generalization, and interpretability.

ARTICLE INFO

Communicated by B. Hu

Keywords:
GeoAI
Autoencoders
Trajectory representation learning
Spatio-temporal data mining
Geospatial artificial intelligence
Latent space learning

ABSTRACT

The rapid growth of spatio-temporal data from sources such as GPS, remote sensing, and Internet of Things (IoT) devices has established trajectory data as a core component of modern Geospatial Artificial Intelligence (GeoAI). However, the high dimensionality, noise, irregular sampling, and heterogeneous nature of trajectory data pose significant challenges for traditional analytical methods. In recent years, autoencoders have emerged as a powerful representation learning framework for modeling complex trajectory patterns by encoding spatio-temporal data into compact and expressive latent spaces. This survey provides a comprehensive review of autoencoder-based methods for trajectory representation learning within the GeoAI domain. Moving beyond the traditional focus on trajectory compression, we present a unified perspective that encompasses a broad range of tasks, including clustering, anomaly detection, similarity computation, recovery and reconstruction, prediction, classification, and trajectory generation. We systematically categorize existing approaches based on autoencoder architectures such as feed-forward, recurrent, convolutional, variational, transformer-based, graph-based, and hybrid models, and analyze their suitability for different trajectory representations and application scenarios. Furthermore, we examine commonly used evaluation metrics across tasks and highlight how latent representations serve as a unifying abstraction for diverse analytical objectives. The survey also reviews real-world applications in domains such as urban mobility, maritime navigation, aviation safety, and interaction-aware modeling. Finally, we identify critical challenges related to limited labeled data, robustness under noisy, incomplete, degraded, or heterogeneous geospatial observations, privacy and ethical data usage, scalability and deployment constraints, lack of standardized evaluation protocols, limited cross-region and cross-domain generalization, multimodal representation alignment, uncertainty estimation, and the interpretability of latent representations. We outline future research directions toward robust and uncertainty-aware autoencoders, degradation-aware reconstruction, physics-guided and foundation model-enhanced GeoAI, multimodal and cross-domain representation learning, privacy-aware methods, scalable deployment, and semantically meaningful and explainable latent spaces.

* Corresponding author.
Email address: violos@hua.gr (J. Violos).

https://doi.org/10.1016/j.neucom.2026.134509
Received 6 May 2026; Received in revised form 6 July 2026; Accepted 13 July 2026
Available online 14 July 2026

1. Introduction

Trajectory data, representing the movement of objects through space and time, have become a fundamental data modality in modern geospatial analytics. The proliferation of sensing technologies, including Global Positioning Systems (GPS), Automatic Identification Systems (AIS), remote sensing platforms and Internet of Things (IoT) devices, has led to an unprecedented increase in the volume, variety and velocity of spatio-temporal data [1]. These developments have enabled a wide range of applications, from urban mobility analysis and intelligent transportation systems to environmental monitoring, aviation safety, and maritime surveillance [2]. However, the high dimensionality, irregular sampling, noise, and heterogeneity inherent in trajectory data pose significant challenges for traditional analytical methods [3,4]. In many real-world settings, trajectory data extend beyond basic spatial-temporal coordinates to include additional attributes such as velocity, heading, acceleration, transportation mode, semantic labels (e.g., activity or land-use context), and sensor-derived signals, effectively increasing the dimensionality and complexity of the data representation.

To address these challenges, the emerging field of Geospatial Artificial Intelligence (GeoAI) integrates machine learning and spatial data science to enable scalable and data-driven modeling of complex geospatial phenomena [5,6]. Within this context, trajectory data mining has evolved from reliance on handcrafted features and distance-based similarity measures toward representation learning paradigms, where models automatically learn meaningful abstractions from raw data [7–10]. This shift is particularly important for capturing the joint spatial and temporal dependencies that characterize real-world movement patterns.

Among representation learning approaches, autoencoders have emerged as a powerful and flexible framework for modeling trajectory data [11–13]. By learning compact latent representations through reconstruction-based objectives, autoencoders enable efficient encoding of complex spatio-temporal structures while preserving essential information. Early work in this area primarily focused on trajectory compression, aiming to reduce storage and transmission costs while maintaining reconstruction fidelity [14,15]. However, recent advances have significantly expanded the role of autoencoders beyond compression, positioning them as a general-purpose tool for a wide range of GeoAI tasks.

This broader role is especially timely as GeoAI increasingly moves toward large-scale, weakly labeled, and multimodal geospatial analysis. Contemporary geospatial applications often combine trajectory records with road-network context, semantic annotations, remote sensing observations, environmental variables, and sensor-derived attributes, creating heterogeneous data representations that are difficult to model using fixed handcrafted features alone. In this setting, autoencoders are important not only for dimensionality reduction and compression, but also for reconstruction-based self-supervised learning, recovery of incomplete or degraded observations, anomaly-sensitive modeling, and the construction of latent spaces that can support downstream GeoAI tasks across different data modalities.

In particular, the latent representations learned by autoencoders provide a unifying foundation for diverse trajectory analysis problems. These include clustering and similarity computation [11,13], where latent embeddings facilitate more robust distance estimation; anomaly detection [16], where deviations from learned normal patterns can be identified through reconstruction errors or latent space density; trajectory recovery and reconstruction [17,18], where missing, sparse, or degraded observations can be inferred from learned spatio-temporal regularities; trajectory prediction [19,20], where encoded temporal dynamics support forecasting; classification [21], where latent features enable discrimination between behavioral patterns; and trajectory generation [22], where structured latent spaces allow the synthesis of realistic movement data. This versatility stems from the ability of autoencoders to adapt their architecture, ranging from feed-forward and recurrent models to convolutional, variational, and transformer-based designs, according to the structural properties of trajectory data. Despite the growing body of work in this area, existing studies are often fragmented across specific tasks, data modalities, or architectural choices, making it difficult to obtain a unified understanding of how autoencoders are applied within GeoAI. Moreover, prior surveys have typically focused on individual problems such as trajectory prediction [23], outlier detection [24], clustering [25] or compression [26], without systematically examining the broader landscape of autoencoder-based methods and their interconnections across tasks.

1.1. Scope and contributions of this survey

This survey aims to provide a comprehensive and structured overview of autoencoder-based approaches for trajectory data representation learning within the GeoAI domain. Unlike prior work that focuses on isolated tasks, we adopt a unifying perspective centered on representation learning, highlighting how autoencoders serve as a common backbone across diverse trajectory analysis problems. Accordingly, this survey should be read as a trajectory-focused review of autoencoder-based representation learning within GeoAI, rather than as a general survey of autoencoder applications across all GeoAI data modalities and tasks.

The main contributions of this survey are summarized as follows:

- GIScience-Grounded Motivation for Autoencoder-Based GeoAI: We clarify the importance and timeliness of autoencoder-based representation learning for GeoAI and connect trajectory modeling to key Geographic Information Science (GIScience) principles, including spatial dependence, spatial heterogeneity, scale and representation dependence, and space-time constraints.
- Unified Perspective on GeoAI Trajectory Tasks: We present a taxonomy of autoencoder-based trajectory analysis tasks, including compression, clustering, anomaly detection, similarity computation, recovery/reconstruction, prediction, classification, and generation, through the lens of latent representation learning.
- Comprehensive Review of Autoencoder Architectures: We systematically categorize and analyze a wide range of autoencoder variants, including feed-forward, recurrent, convolutional, variational, transformer-based, graph-based, and hybrid architectures, emphasizing their suitability for different trajectory data types, representations, and task requirements.
- Integration of Data Representations, Loss Formulations, and Modeling Choices: We examine how trajectory representations, including raw point sequences, enriched feature vectors, grid-based encodings, graph structures, and high-dimensional sensor data, influence architectural design, reconstruction objectives, regularization strategies, and downstream learning behavior.
- Cross-Architecture and Task-Specific Comparative Synthesis: We provide a comparative analysis of architecture usage across tasks and domains, discuss task-specific representation requirements, and summarize the strengths, limitations, and practical suitability of major autoencoder families for different GeoAI trajectory problems.
- Dataset, Metric, and Application-Oriented Analysis: We review real-world applications across urban mobility, maritime navigation, aviation safety, synthetic validation, and interaction-aware systems, while also summarizing public datasets, software resources, evaluation metrics, and the limits of direct quantitative comparison across studies.
- Positioning Autoencoders within Emerging Trajectory GeoAI Paradigms: We discuss how recent approaches such as diffusion models, flow matching, and LLM-based trajectory methods relate to autoencoder-based representation learning, highlighting whether they complement, extend, or differ from autoencoder-centered approaches.

Table 1
Taxonomy of autoencoder type, task formulation, application domain and dataset (Part I).

Paper	AE Type	Task	Domain	Dataset
Feed-Forward Autoencoder Architectures				
Chu et al. [27]	Stacked AE	Clustering	Aviation	custom
Song et al. [28]	Dense AE	Compression	Other Applications	custom
Zeng et al. [29]	Deep AE	Clustering	Aviation	custom
Wang et al. [30]	Deep AE	Anomaly Detection	Synthetic data	custom
Zhang et al. [31]	Deep AE	Anomaly Detection	Aviation	custom
Olive et al. [32]	Deep AE	Clustering/Anomaly Detection	Aviation	custom
Olive et al. [33]	Deep AE	Anomaly Detection	Aviation	custom
Peralta et al. [34]	Stacked AE	Anomaly Detection	Urban & Human Mobility	custom
Yue et al. [35]	Stacked AE	Clustering	Urban & Human Mobility	GeoLife [36–38]
Yu et al. [39]	Deep AE	Anomaly Detection	Urban & Human Mobility	custom
Murray et al. [40]	Dual Linear AE	Prediction	Maritime	custom
Xu et al. [41]	Contrastive Masked AE	Prediction	Interaction-Aware Systems (Vehicles)	INTERACTION [42], Argoverse 2 [43,44], Waymo Open Motion [45]
Gajamannage et al. [46]	Hadamard Deep AE	Recovery/reconstruction	Other Applications	custom/synthetic
Convolutional Autoencoder Architectures				
Wang et al. [47]	Multi-feature Fusion AE	Clustering	Maritime	custom
Liang et al. [48]	Convolutional AE	Similarity	Maritime	custom
Olive et al. [49]	Deep Convolutional AE	Clustering	Aviation	custom
Rákos et al. [50]	Convolutional VAE	Compression	Interaction-Aware Systems (Vehicles)	NGSIM [51]
Li et al. [52]	Convolutional AE	Similarity	Urban & Human Mobility	custom
Qi et al. [53]	Multi-scale Convolutional AE	Anomaly Detection	Maritime	custom
Memarzadeh et al. [54]	Convolutional VAE	Anomaly Detection	Aviation	custom
Lu et al. [21]	Dual Convolutional AE	Classification	Urban & Human Mobility	GeoLife [36–38], SHL [55]
Dabiri et al. [56]	Convolutional AE	Classification	Urban & Human Mobility	GeoLife [36–38]
Rakos et al. [57]	Convolutional VAE	Prediction	Interaction-Aware Systems (Vehicles)	NGSIM [51]
Krauth et al. [58]	Temporal Convolutional VAE	Generation	Aviation	custom
Cheng et al. [59]	Self-attention Convolutional VAE	Prediction	Urban & Human Mobility	TrajNet++ [60]
Liu et al. [61]	Convolutional VAE	Prediction	Interaction-Aware Systems (Vehicles)	highD [62]
Li et al. [63]	Deep Convolutional AE	Similarity	Maritime	custom
Wang et al. [64]	Multimodal Convolutional AE with attention	Similarity	Maritime	custom

- Identification of Open Challenges and Future Directions: We identify key limitations and future research directions, including scalability, privacy, robustness, interpretability, multimodal learning, cross-domain generalization, standardized evaluation, and real-time deployment in dynamic geospatial environments (Tables 1–3).

1.2. Structure of the survey

The remainder of this survey is organized as follows. Section 2 establishes the geographic and data-oriented background of the review. It first clarifies the importance and timeliness of autoencoder-based methods for GeoAI, then introduces key GIScience principles relevant to trajectory representation learning, including spatial dependence, spatial heterogeneity, scale and representation dependence, and space-time constraints. The section also discusses GeoAI and trajectory representation learning, defines the main trajectory data modalities and representations, and summarizes evaluation metrics and infrastructure considerations.

Section 3 presents the high-level taxonomy of autoencoder-based trajectory representation learning. It introduces the main autoencoder principles and architectural variants and organizes the major trajectory-analysis tasks addressed in the reviewed literature, including compression, clustering, anomaly detection, similarity computation, recovery/reconstruction, prediction, classification, and generation. Section 4 then provides an architecture-oriented review of representative autoencoder methods, covering feed-forward, recurrent, convolutional, variational, transformer-based, graph-based, and hybrid models.

Section 5 reviews the main application domains in which autoencoder-based trajectory methods have been used, including urban mobility and smart-city analytics, maritime navigation, aviation flow analysis, interaction-aware trajectory modeling, and synthetic validation. Section 6 provides a comparative synthesis of the reviewed literature, including architecture distributions across tasks and domains, task-specific representation requirements, formal cross-architecture analysis, architecture-task suitability, dataset-based metric comparisons, and practitioner-oriented guidelines for architecture selection.

Section 7 summarizes public datasets and software frameworks that support trajectory analysis and autoencoder-based GeoAI research. Section 8 discusses emerging trajectory modeling paradigms beyond conventional autoencoder-based approaches, including diffusion models, flow matching, and LLM-based trajectory modeling, and analyzes how these directions relate to autoencoder-centered representation learning. Section 9 synthesizes the main open challenges and future research directions identified from the reviewed trajectory-focused corpus and from adjacent GeoAI developments discussed in Section 8. It first summarizes the remaining limitations affecting robustness, transferability, interpretability, evaluation, and deployment and then outlines promising directions for developing more robust, uncertainty-aware, multimodal, explainable, and operational autoencoder-based GeoAI. Finally, Section 10 concludes the survey.

2. Geographic foundations and trajectory data context

This section establishes the conceptual background for understanding autoencoder-based trajectory modeling in GeoAI. We first clarify the importance and timeliness of autoencoders for modern geospatial representation learning, before grounding trajectory analysis in

Table 2
Taxonomy of autoencoder type, task formulation, application domain and dataset (Part II).

RNN/LSTM Autoencoder Architectures				
Liu et al. [65]	Multi-scale Attention AE	Clustering	Maritime	not public
Kölle et al. [14]	Seq2Seq AE	Compression	Interaction-aware Systems (Vehicles)	T-Drive [66]
Horovitz et al. [67]	Symmetric AE	Compression	Interaction-Aware Systems (Vehicles)	NGSIM [51]
Yao et al. [11]	Seq2Seq AE	Clustering	Maritime	custom
Liatsikou et al. [16]	Denoising Sequential AE	Anomaly Detection	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68]
Suo et al. [69]	GRU	Prediction	Maritime	–
Dong et al. [70]	Adaptive Forgetting-Controlled RNN (AFC-RNN)	Prediction	Interaction-Aware Systems (Human)	ETH [71], UCY [72], SDD [73]
Li et al. [13]	Seq2Seq AE	Similarity	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68]
Alhussein et al. [74]	LSTM AE	Anomaly Detection	Aviation	custom
Quan et al. [75]	Holistic LSTM	Prediction	Interaction-Aware Systems (Human)	JAAD [76,77], PIE [78]
Bhujel et al. [79]	RNN	Prediction	Interaction-Aware Systems (Human)	ETH [71], UCY [72]
Xue et al. [80]	LSTM	Prediction (via clustering)	Interaction-Aware Systems (Human)	Edinburgh Informatics Forum (Pedestrian dataset) [81]
Miguel et al. [82]	Recurrent VAE	Prediction	Interaction-aware Systems (Vehicles)	highD [62]
Huang et al. [83]	Sequential VAE	Generation	Urban & Human Mobility	custom
Liu et al. [84]	Gaussian Mixture Sequential VAE	Anomaly Detection	Urban & Human Mobility	custom
Song et al. [20]	Convolutional LSTM	Prediction	Urban & Human Mobility	–
Demetriou et al. [85]	Recurrent AE	Generation	Synthetic Validation & Latent Pattern Intelligence	Non public dataset provided by Volvo Cars Corporation (multiple datasets, not found)
Ma et al. [86]	RNN-based AE	Anomaly Detection	Urban & Human Mobility	custom
Capobianco et al. [87]	Attention-based Variational LSTM	Prediction	Maritime	custom
Jiang et al. [88]	LSTM, GRU, and Stacked AE (multiple models)	Prediction	Interaction-Aware Systems (Vehicles)	NGSIM [51]
Sheng et al. [9]	Siamese Denoising LSTM AE	Classification	Urban & Human Mobility	–
Wang et al. [89]	Seq2Seq AE	Clustering	Urban & Human Mobility	–
Zhang et al. [90]	Seq2Seq AE	Similarity	Urban & Human Mobility	custom
Yao et al. [12]	Seq2Seq AE	Clustering	Maritime	custom
Feng et al. [91]	Conditional VAE + LSTM	Prediction	Interaction-Aware Systems (Vehicles)	NGSIM [51]
Wang et al. [18]	Recurrent encoder-decoder with attention	Recovery/reconstruction	Urban & Human Mobility	GeoLife [36–38]
Ren et al. [92]	Seq2Seq	Recovery/reconstruction	Urban & Human Mobility	custom
Ye et al. [17]	RNN AE	Recovery/reconstruction	Urban & Human Mobility	not found

Table 3
Taxonomy of autoencoder type, task formulation, application domain and dataset (Part III).

Transformer-based Architectures				
Chen et al. [93]	Transformer-VAE	Clustering	Aviation	custom
Hou et al. [94]	Transformer–VAE	Anomaly Detection	Maritime	custom
Chen et al. [95]	Transformer-based Masking AE	Prediction	Interaction-aware Systems	INTERACTION [42], Argoverse [96], TrajNet++ [60]
Wu et al. [97]	U-type Sparse Attention AE	Clustering	General/Cross-Domain	GeoLife [36–38] + other synthetic custom datasets
Shi et al. [98]	Trajectory Unified Transformer AE	Prediction	Interaction-Aware Systems (Human)	ETH [71], UCY [72], SDD [73]
Chen et al. [99]	Spatial-temporal Transformer	Recovery/reconstruction	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68]
VAE Architectures				
Yue et al. [100]	VAE	Clustering	Urban & Human Mobility	GeoLife [36–38]
Zhang et al. [101]	VAE	Anomaly Detection	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68]
Cho et al. [10]	VAE	Anomaly Detection	Urban & Human Mobility	custom
Lu et al. [102]	VAE	Anomaly Detection	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68]
Neumeier et al. [103]	Descriptive VAE	Prediction	Interaction-aware Systems (Vehicles)	highD [62]
Chen et al. [22]	VAE	Generation	Synthetic Validation & Latent Pattern Intelligence	–
Xie et al. [104]	Gaussian Mixture VAE	Anomaly Detection	Maritime	MarineCadastre AIS [105]
Cheng et al. [106]	Conditional VAE	Prediction	Interaction-aware Systems	inD [107], TrajNet++ [60]
Xu et al. [108]	Context-aware VAE	Prediction	Interaction-aware Systems	nuScenes [109], Waymo Open Motion [45], (+Lyft not found)
Lee et al. [110]	Conditional VAE-based framework	Prediction	Interaction-aware Systems	SDD [73], TrajNet++ [60], nuScenes [109]
Ivanovic et al. [111]	Conditional VAE	Prediction	Interaction-Aware Systems (Human)	ETH [71], UCY [72]
Xu et al. [112]	VAE	Prediction	Interaction-Aware Systems (Human)	ETH [71], UCY [72], SDD [73], (+NBA not public)
Graph Architectures				
Wiederer et al. [113]	Spatio-Temporal Graph AE	Anomaly Detection	Interaction-Aware Systems (Vehicles)	custom
Zhao et al. [114]	Masked Graph AE	Prediction	General/Cross-Domain	–
Wei et al. [115]	Graph-based encoder-decoder	Recovery/reconstruction	Urban & Human Mobility	Porto Taxi (ECML PKDD 2015) [68] +other not found

key GIScience principles. We then introduce the GeoAI trajectory representation-learning paradigm, define the main trajectory data modalities and representations, and summarize the evaluation and infrastructure considerations that shape the design, assessment, and deployment of autoencoder-based GeoAI methods.

2.1. Importance and timeliness of autoencoders for GeoAI

The importance of autoencoders for Geospatial Artificial Intelligence (GeoAI) stems from a broader shift in geospatial analysis. Modern geographic data are increasingly large-scale, heterogeneous, high-dimensional, weakly labeled, and generated from multiple sensing and observation systems. Trajectories, remote sensing imagery, road-network data, environmental observations, and sensor-derived mobility attributes often contain complex spatial, temporal, semantic, and contextual dependencies that are difficult to represent using manually designed features or fixed distance measures. In this setting, autoencoders provide a general representation-learning mechanism that can transform complex geospatial observations into compact latent spaces while preserving task-relevant structure through reconstruction-based objectives. This is particularly relevant for GeoAI because recent work increasingly emphasizes self-supervised learning as a way to exploit large geospatial datasets when labeled data are sparse, unevenly distributed, or costly to obtain [116].

The timeliness of this review is also connected to the changing nature of geospatial data and GeoAI workflows. Contemporary GeoAI is no longer limited to single-modality spatial records or isolated coordinate sequences. Instead, it increasingly requires the integration of heterogeneous data sources, including trajectories, remote sensing imagery, semantic attributes, environmental variables, road-network context, and multi-agent interaction information. Recent surveys on multimodal remote sensing data fusion highlight this broader movement toward deep representation learning methods capable of handling heterogeneous Earth observation and geospatial data sources [117]. Within this landscape, autoencoders are important not only as compression models, but as flexible encoder–decoder frameworks that support dimensionality reduction, latent representation learning, reconstruction of incomplete or degraded data, anomaly-sensitive modeling, and downstream task adaptation.

This relevance has been further reinforced by recent developments in self-supervised and foundation model-based GeoAI. In remote sensing, masked image modeling and masked autoencoding have become important reconstruction-based pretraining strategies for learning transferable representations from large unlabeled datasets, including settings affected by cloud cover, occlusions, sensor limitations, and missing observations [118]. Similarly, recent surveys on remote sensing foundation models emphasize the role of self-supervised pretraining, including contrastive learning and masked autoencoders, in improving the robustness and transferability of large-scale models [119]. At the same time, the domain-specific characteristics of remote sensing and geospatial data continue to motivate specialized foundation models rather than direct transfer from natural-image models [120].

Therefore, autoencoders remain timely because they connect several active directions in GeoAI: compact representation learning, self-supervised pretraining, multimodal fusion, reconstruction under missing or degraded observations, uncertainty-aware modeling, and generative spatial-temporal analysis. Broader surveys of generative techniques for spatial-temporal data mining further show that masked autoencoders, sequence-to-sequence models, diffusion models, and large language model-based approaches are increasingly being discussed within a common spatial-temporal modeling landscape [121]. From this perspective, autoencoders are not only a family of models for individual tasks, but also a methodological bridge between heterogeneous geospatial data representations and the downstream analytical needs of modern GeoAI. This makes a dedicated survey of autoencoder-based methods, variants, applications, and limitations both relevant and timely.

2.2. Geographic information science principles for trajectory analysis

This subsection introduces the GIScience principles that are most relevant to autoencoder-based trajectory analysis. Rather than providing a general overview of GIScience, the discussion focuses on concepts that directly affect how trajectory data are represented, modeled, interpreted, and evaluated in GeoAI. Specifically, we highlight spatial dependence, spatial heterogeneity, scale and representation dependence, and space-time constraints as core principles that shape the design and interpretation of latent trajectory representations. These principles provide the geographic foundation for the autoencoder architectures and task taxonomy discussed in the following section.

GIScience foundations and GeoAI. GIScience provides the conceptual foundation for understanding why trajectory analysis in GeoAI cannot be treated as a generic sequence-modeling problem. GIScience has long emphasized that geographic information involves scientific questions that extend beyond the technical handling of spatial data, including how geographic phenomena are represented, measured, modeled, analyzed, and interpreted [122]. This perspective is particularly important for trajectory data, where observations are not independent records but movement traces embedded in geographic space, constrained by spatial relations, temporal order, environmental context, and domain-specific mobility processes. Accordingly, GeoAI should not be understood merely as the application of artificial intelligence to geospatial datasets, but as the development of spatially explicit learning methods that incorporate geographic representations, spatial relations, and contextual knowledge into data-driven models [123]. In the context of data-driven GeoAI, these principles are closely linked to the broader representation learning problem, where spatial entities such as points, trajectories, or regions must be encoded into learning-friendly representations for downstream tasks [124]. Rather than operating directly on raw geographic data, modern models rely on latent embeddings that aim to preserve meaningful spatial structure while enabling efficient learning. Consequently, grounding autoencoder-based trajectory modeling in GIScience principles is essential not only for understanding how such representations are constructed, but also for clarifying which geographic properties they should retain.

Spatial dependence and Tobler's first law. A central principle of GIScience is spatial dependence, commonly expressed through Tobler's First Law of Geography, according to which geographic entities are generally more related when they are closer in space [125]. This principle implies that movement observations should not be interpreted as isolated records, since nearby locations, neighboring road segments, adjacent regions, or interacting moving objects often exhibit related spatial and behavioral patterns. In trajectory analysis, spatial dependence is reflected in several modeling assumptions, including the use of local neighborhoods, distance-based similarity, map-matched network proximity, spatial interpolation, and density-based descriptions of movement behavior. Goodchild further emphasizes that Tobler's First Law provides a conceptual foundation for many GIScience operations, including spatial interpolation, resampling, generalization, contouring, and spatial data modeling [126]. For autoencoder-based trajectory modeling, this principle suggests that latent representations should preserve meaningful spatial proximity and neighborhood relationships, so that trajectories that are geographically or behaviorally similar remain close in the learned embedding space. This is particularly important for downstream tasks such as clustering, similarity search, anomaly detection, and prediction, where distorted spatial relations in the latent space can lead to misleading interpretations of movement patterns.

Spatial heterogeneity and context dependence. A second principle is spatial heterogeneity, often discussed in spatial analysis in relation to non-stationarity, context dependence, and spatially varying relationships. While spatial dependence emphasizes relationships among nearby observations, spatial heterogeneity emphasizes that geographic processes

and data-generating mechanisms may vary across locations. Spatial dependence and spatial heterogeneity have been identified as distinctive properties of spatial data that complicate the direct application of standard statistical assumptions, particularly those based on independence, homogeneity, or globally fixed model parameters [127]. This principle is directly relevant to trajectory analysis because movement patterns are shaped by local geographic context: the same speed, direction change, stop duration, or route deviation may have different meanings in an urban street network, a maritime corridor, an airport terminal area, or a pedestrian environment. More generally, studies of spatial heterogeneity show that averaging or globally smoothing heterogeneous spatial structures can obscure important local variation and connectivity patterns [128]. For autoencoder-based trajectory modeling, spatial heterogeneity suggests that learned representations should be sensitive to domain, region, and context, rather than assuming that a single global latent structure is equally valid across all geographic settings. This issue is particularly important for cross-domain generalization, transfer learning, anomaly detection, and evaluation, where models trained in one spatial context may not preserve the same semantic meaning when applied to another.

Scale, granularity, and representation dependence. A further GIScience principle concerns scale, granularity, and the representation-dependence of trajectory analysis. Movement data are inherently multi-scale; their interpretation can change according to the spatial resolution, temporal sampling interval, segmentation strategy, and movement attributes selected for analysis. In movement-pattern analysis, trajectories are commonly described through spatial, temporal, and spatio-temporal parameters such as position, distance, direction, duration, speed, velocity, and acceleration, while the scale and granularity of these parameters strongly influence the patterns that can be detected and interpreted [129]. This issue also appears in trajectory similarity analysis, where different comparison methods may preserve different aspects of movement depending on whether spatial and temporal information are modeled separately or jointly [130]. In trajectory representation learning, this creates a modifiable representation problem. The same movement process may be encoded as raw points, line segments, fixed-length sequences, grid images, network paths, or semantic trajectories, each exposing different spatial and temporal properties to the model, as illustrated in Fig. 2. Consequently, autoencoder-based methods should be interpreted and compared with attention to the scale and granularity of the input representation, since changes in representation can alter both the learned latent space and the conclusions drawn from downstream tasks.

Space-time integration and movement constraints. Finally, trajectory analysis requires the explicit integration of space and time, since movement is not merely an ordered sequence of locations but a process unfolding under spatial, temporal, and contextual constraints. Time-geographic theory conceptualizes movement as a space-time path, where possible actions are limited by mobility capabilities, coordination requirements with other individuals or activities, and constraints imposed by institutions or the surrounding environment [131]. Space-time prism models further operationalize this perspective by showing that feasible movement and accessibility depend jointly on location, travel time, velocity, transportation network structure, and activity schedules [132]. These principles clarify why trajectory representations should preserve more than geometric displacement: depending on the task, they may also need to retain temporal ordering, duration, speed, recurrence, interaction, accessibility, and contextual constraints. Therefore, autoencoder-based trajectory models should be understood as geographic representation-learning systems, whose latent spaces are expected to preserve not only compactness and reconstruction fidelity, but also the spatio-temporal structure and movement semantics that make trajectory data meaningful within GeoAI.

2.3. GeoAI and trajectory representation learning

GeoAI is an emerging interdisciplinary field that integrates advances in artificial intelligence, machine learning, and spatial data science to extract meaningful patterns from large-scale geospatial data [123,133–135]. By combining domain knowledge from geography with data-driven modeling techniques, GeoAI enables the analysis of complex spatial phenomena, including urban mobility, environmental dynamics, and infrastructure usage. The increasing availability of high-resolution spatial-temporal data from sources such as GPS, remote sensing platforms, and Internet of Things (IoT) devices has further accelerated the adoption of deep learning methods within geospatial analytics [136]. In this context, GeoAI provides a unifying framework where spatial reasoning and learning-based approaches are jointly leveraged to model both the geometric structure and semantic meaning of geospatial data.

Within this paradigm, trajectory data representing the movement of objects such as vehicles, vessels, or individuals, over time has become a central data modality in GeoAI applications. Trajectory data mining focuses on uncovering patterns such as frequently traveled routes, mobility behaviors, and anomalous movements, often under challenging conditions including irregular sampling, noise, and high dimensionality [137]. Traditional approaches have relied heavily on handcrafted similarity measures and rule-based models [7,8,15,93]. However, these methods often struggle to scale and generalize across heterogeneous datasets. Recent advances in deep learning have shifted this paradigm toward representation learning [4,8,138,139], where models automatically extract latent features that capture both spatial and temporal dependencies, significantly improving performance in large-scale trajectory analysis tasks.

Autoencoders constitute a powerful tool within GeoAI for learning compact and expressive representations of trajectory data. By encoding high-dimensional spatial-temporal inputs into lower-dimensional latent spaces, autoencoders enable efficient compression while preserving essential structural and behavioral characteristics of movement patterns. These learned representations have been successfully applied across a wide range of GeoAI tasks, including trajectory similarity search [13], mobility pattern discovery [35], and large-scale clustering of transportation data [89].

This general workflow in which raw trajectory data is encoded in a latent representation and subsequently used for downstream tasks is illustrated in Fig. 1. The pipeline consists of a sequence of processing steps, starting from raw geospatial data, which are first transformed into trajectories through a trajectory construction process. These trajectories are then converted into suitable representations that can be ingested by the autoencoder model. The encoder maps the input data into a latent space, which serves as a compact intermediate abstraction of the underlying spatio-temporal patterns. This latent representation is subsequently utilized to support a range of downstream tasks, as shown in the figure. In addition, some approaches incorporate an optional task-specific refinement step, represented by the dashed feedback loop, where the learned latent space is further adapted using task-driven objectives or supervision signals to improve performance for particular applications. The figure therefore highlights both the forward processing pipeline and the potential interaction between downstream tasks and representation learning. As demonstrated throughout this review, the flexibility of autoencoder architectures allows them to adapt to various trajectory modalities and learning paradigms, positioning them as a foundational component in modern data-driven geospatial analysis [24].

2.4. Trajectory data: Definitions, modalities, and representations

Trajectory data represent the movement of objects through space over time and are typically modeled as ordered sequences of spatio-temporal observations. Formally, a trajectory can be defined as a sequence $T = \{p_1, p_2, \ldots, p_n\}$, where each point $p_i = (x_i, y_i, t_i)$ encodes the spatial location and timestamp of the moving object at a given time step [140]. Depending on the application, trajectories may

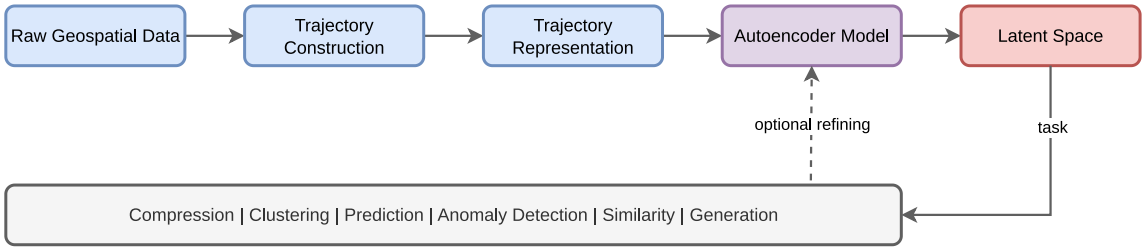

Fig. 1. General pipeline of autoencoder usage in GeoAI.

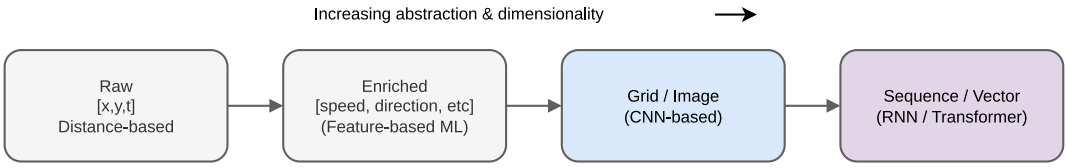

Fig. 2. Progression of data representations from simple structured formats to more complex forms. The structure of the input data influences the choice of autoencoder architecture.

extend beyond simple 2D spatial coordinates as shown in Fig. 2 to include additional dimensions such as altitude, resulting in 3D or even higher-dimensional representations. This sequential structure inherently captures both spatial continuity and temporal dependencies, which are essential characteristics of real-world movement processes. As a result, trajectory data provide a natural representation for modeling dynamic phenomena, making them a fundamental component in GeoAI applications [137,141], involving mobility analysis, transportation systems, and dynamic environment modeling.

Beyond their basic structure, trajectories often include multiple complementary attributes that enrich their descriptive power [142]. In addition to spatial coordinates and timestamps, trajectory points may incorporate kinematic features such as velocity, acceleration, and heading direction, either directly recorded or derived from raw data. Furthermore, semantic and contextual information, such as transportation mode, road network constraints, environmental conditions, or interaction with other agents, can be associated with trajectories to support higher-level analysis [143]. These heterogeneous attributes significantly increase the complexity and dimensionality of trajectory data, posing challenges for traditional analytical methods while motivating the use of latent representation learning techniques.

Trajectory data can be broadly categorized into several types based on their structure and modality. The most common form is raw point-based trajectories, obtained from sources such as GPS, AIS, or ADS-B systems, which consist of irregularly sampled spatio-temporal points and are often affected by noise and missing data. To address these limitations, structured or enriched trajectories incorporate preprocessing steps such as map-matching, interpolation, or semantic annotation, enabling more meaningful comparisons and behavioral analysis [144]. Alternatively, trajectories can be transformed into grid-based or rasterized representations, such as images, heatmaps, or occupancy grids, to leverage convolutional architectures for spatial pattern extraction [48]. Finally, high-dimensional trajectory representations arise in sensor-rich environments, including LiDAR point sequences, skeletal motion capture data, and multivariate time series, where each trajectory point encodes complex spatial or physical measurements [4,145]. These diverse trajectory types necessitate flexible modeling approaches, as different representations emphasize distinct spatial, temporal, and semantic characteristics.

2.5. Evaluation metrics for autoencoder-based trajectory analysis

The evaluation of autoencoder-based models for trajectory data is inherently task-dependent, as different applications require assessing distinct aspects of performance. While reconstruction-based metrics are commonly used to evaluate the quality of learned representations, downstream tasks such as clustering, prediction, and anomaly detection rely on specialized evaluation criteria. Furthermore, trajectory data introduce additional challenges due to their spatio-temporal nature, variable length, and potential irregular sampling. In this subsection, we provide an overview of commonly used evaluation metrics, organized according to the primary GeoAI tasks discussed in this review. A summary of the evaluation metrics used across different trajectory analysis tasks is provided in Table 4.

Compression and reconstruction metrics. For trajectory compression tasks, evaluation focuses on the trade-off between data reduction and information preservation. Reconstruction accuracy is typically measured using metrics such as Mean Squared Error (MSE) and Signal-to-Noise Ratio (SNR) [28], which quantify the difference between the original and reconstructed trajectories. In some cases, Peak Signal-to-Noise Ratio (PSNR) is also employed as a normalized variant of SNR. Compression efficiency is assessed using the Compression Ratio (CR) [67], which reflects the degree of size reduction achieved by the model. Several works also report incremental changes (e.g., Δ_{MSE}, Δ_{SNR}) to evaluate improvements over baseline methods. These metrics collectively capture the balance between compact representation and reconstruction fidelity.

Clustering metrics. Trajectory clustering is commonly evaluated using both internal and external validation metrics. Internal metrics, such as the Sum of Squared Errors (SSE), Silhouette Coefficient (SC), Davies–Bouldin Index (DBI), and Calinski–Harabasz Index (CH) [93], assess the compactness and separation of clusters without relying on ground truth labels. External metrics, including Normalized Mutual Information (NMI) [89,146], Rand Index (RI), Adjusted Rand Index (ARI) [65], and Fowlkes–Mallows Index (FMI) [35], measure the agreement between predicted clusters and known labels when available. Additionally, clustering accuracy and mean distance-based measures are sometimes used to evaluate the quality of learned trajectory representations.

Prediction metrics. Trajectory prediction tasks require evaluating the accuracy of forecasted positions over time. Standard regression-based metrics such as Mean Squared Error (MSE) [75], Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) [82] are widely used, commonly computed separately for spatial dimensions (e.g., longitudinal and lateral components). More specialized metrics, often employed in the studies analyzed in this review, include Average Displacement

Table 4
Metrics for trajectory data tasks.

Metric	Task	Description
Mean Squared Error (MSE)	Compression, Prediction	Measures average squared difference between original and reconstructed/predicted trajectories
Root Mean Squared Error (RMSE)	Prediction	Square root of MSE, provides error in original units
Mean Absolute Error (MAE)	Prediction	Measures average absolute difference between predicted and actual values
Signal-to-Noise Ratio (SNR)	Compression	Measures signal quality relative to reconstruction noise
Peak Signal-to-Noise Ratio (PSNR)	Compression	Normalized version of SNR for comparing reconstruction quality
Compression Ratio (CR)	Compression	Ratio of original data size to compressed representation size
Silhouette Coefficient (SC)	Clustering	Measures how well trajectories fit within their assigned cluster
Davies–Bouldin Index (DBI)	Clustering	Evaluates cluster separation and compactness (lower is better)
Calinski–Harabasz Index (CH)	Clustering	Ratio of between-cluster to within-cluster variance
Sum of Squared Errors (SSE)	Clustering	Measures intra-cluster compactness
Normalized Mutual Information (NMI)	Clustering	Measures agreement between clustering and ground truth labels
Rand Index (RI)	Clustering	Measures the agreement between predicted and true cluster assignments
Adjusted Rand Index (ARI)	Clustering	Adjusted version of the Rand Index that accounts for agreement occurring by chance
Fowlkes-Mallows Index (FMI)	Clustering	Measures similarity between predicted and true clusters
Average Displacement Error (ADE)	Prediction	Average distance between predicted and ground-truth trajectories over all time steps
Final Displacement Error (FDE)	Prediction	Distance between predicted and true final positions
Negative Log-Likelihood (NLL)	Prediction	Measures likelihood of predicted trajectory under probabilistic models
Miss Rate (MR)	Prediction	Percentage of predictions exceeding a predefined error threshold
Dynamic Time Warping (DTW)	Similarity, Generation	Measures similarity between trajectories with temporal alignment
Hausdorff Distance	Similarity, Generation	Measures maximum deviation between two trajectories
Euclidean Distance (ED)	Similarity	Measures direct point-wise distance in space or latent space
k-nearest neighbor (k-NN)	Similarity	Evaluates similarity via nearest neighbor search
Accuracy (ACC)	Anomaly Detection	Measures overall correctness of anomaly classification
Clustering Accuracy	Clustering	Percentage of correctly assigned cluster labels
Precision	Anomaly Detection	Fraction of correctly identified anomalies among detected ones
Recall	Anomaly Detection	Fraction of actual anomalies correctly identified
F1-score	Anomaly Detection	Harmonic mean of precision and recall
PR-AUC	Anomaly Detection	Precision-recall trade-off across thresholds; robust to class imbalance.
Detection Rate (DR)	Anomaly Detection	Proportion of correctly detected anomalies
False Alarm Rate (FAR)	Anomaly Detection	Proportion of normal trajectories incorrectly flagged
Reconstruction Error	Anomaly Detection	Measures deviation between input and reconstructed trajectory
Mean Distance Error (MDE)	Generation	Measures average deviation between generated and real trajectories
Kullback–Leibler Divergence (KL)	Generation	Measures difference between generated and real data distributions
Coverage	Generation	Measures how well generated trajectories represent real data diversity
Matching Score	Generation	Measures similarity between generated and real trajectory sets

Error (ADE) and Final Displacement Error (FDE) [147], which measure the average and final deviation between predicted and ground-truth trajectories, respectively. Probabilistic approaches may additionally employ likelihood-based metrics such as Negative Log-Likelihood (NLL) [112,148]. Variants of MSE computed over specific trajectory components or prediction horizons are also commonly reported [75]. These metrics collectively assess both short-term accuracy and long-term prediction consistency.

Similarity metrics. Trajectory similarity is typically evaluated using distance-based measures that quantify the resemblance between movement patterns. Common approaches include Euclidean distance (ED) [86] in the learned latent space and more specialized sequence alignment techniques such as Dynamic Time Warping (DTW) [86,149], which account for temporal misalignment. Other measures, such as Hausdorff distance [86,150], capture the maximum deviation between trajectories. In practical applications, similarity is often assessed through retrieval-based evaluations, such as k-nearest neighbor (k-NN) queries [90], where the quality of the learned representation is reflected in the ability to retrieve similar trajectories.

Anomaly/outlier detection metrics. Anomaly detection in trajectory data is typically formulated as a binary classification problem, where the objective is to distinguish normal from abnormal patterns. As such, standard classification metrics including Accuracy (ACC), Precision, Recall, and F1-score are widely used [39]. Additionally, Precision–Recall Area Under the Curve (PR-AUC) [102] is often preferred in imbalanced settings. Domain-specific metrics such as Detection Rate (DR) and False Alarm Rate (FAR) are also employed to assess detection performance. In autoencoder-based approaches, reconstruction error is frequently used as an anomaly score, with abnormal trajectories exhibiting significantly higher reconstruction deviations.

Generation metrics. Trajectory generation tasks focus on evaluating the realism, diversity, and accuracy of synthesized trajectories. Distance-based metrics such as Mean Distance Error (MDE) [83] are used to assess similarity to real trajectories. Distribution-based measures, including Kullback–Leibler (KL) divergence [83,151], evaluate how well the generated data matches the underlying data distribution. Additionally, similarity-based approaches using Dynamic Time Warping (DTW) [85] or Hausdorff distance [22] as mentioned above, are employed to compare generated and real trajectory sets. Some works further assess generation quality using matching and coverage-based evaluation schemes, which quantify how well generated samples represent the diversity of real trajectories. These measures are typically implemented using distance-based comparisons and are often tailored to the specific dataset and application.

General evaluation considerations. Beyond task-specific metrics, several broader evaluation aspects are important in assessing autoencoder-based models for trajectory data. In compression settings, a key consideration is the trade-off between compression ratio and reconstruction accuracy [67]. Qualitative evaluation, including trajectory visualizations [75], is often used to complement quantitative metrics and provide insights into spatial and temporal consistency. Furthermore, the ability of models to generalize to unseen trajectories is critical, particularly in real-world applications where data distributions may vary. These considerations highlight that a comprehensive evaluation should combine multiple metrics and perspectives to fully capture model performance.

2.6. Infrastructure and deployment considerations

The computational infrastructure used for training and evaluating autoencoder-based trajectory models in the analyzed literature is predominantly centered around conventional computing environments.

Most studies rely on desktop/laptop computers [30,47,53], workstations [48,93], or server-grade machines equipped with multi-core CPUs and, in many cases, powerful dedicated GPUs [54,100]. The use of GPUs is particularly common for training deep architectures such as convolutional, recurrent, and transformer-based autoencoders, reflecting the computational demands of large-scale spatio-temporal data processing [84,95]. In contrast, several studies report experiments conducted without GPU acceleration, although detailed information regarding training duration and computational efficiency is often not provided, making it difficult to assess the practical feasibility of such configurations [11,29,101]. Overall, the experimental setups observed across the reviewed studies suggest a strong emphasis on offline training and evaluation under controlled computational settings.

The reliance on centralized and resource-rich hardware highlights an important gap between research practices and real-world deployment requirements. While several works demonstrate efficient inference or near real-time performance under specific conditions [58,112], the majority of approaches do not explicitly consider constraints related to latency, memory usage, or energy efficiency. As a result, the practical deployment of these models in dynamic environments, such as transportation systems, autonomous vehicles, or urban sensing platforms, remains challenging. These limitations are closely related to the broader scalability and deployment issues discussed in Section 9.1.6.

Beyond the currently adopted infrastructure, emerging computing paradigms such as cloud [152], edge [153], and fog computing [154] present promising directions for deploying GeoAI trajectory models. Cloud-based solutions offer scalable resources for training and large-scale data processing, while edge and on-device deployment enable real-time inference closer to data sources, reducing latency and communication overhead which is an essential requirement in time-critical applications such as autonomous driving [108] and real-time trajectory prediction [155]. However, these deployment settings, along with their associated hardware and system-level requirements, are only sparsely addressed in the existing literature. As discussed in the future directions (Section 9.2.7), the development of lightweight models, model compression techniques, and distributed learning frameworks is essential for supporting efficient deployment of autoencoder-based trajectory analysis systems across diverse computational environments.

3. High-level taxonomy of autoencoder-based trajectory representation learning

This section introduces the high-level taxonomy used to organize autoencoder-based trajectory representation learning in GeoAI. We first present the fundamental principles of autoencoders and summarize their main architectural variants (Section 3.1), establishing the basis for comparing different encoder–decoder designs. We then outline the primary trajectory-analysis tasks addressed by autoencoder-based approaches (Section 3.2), highlighting how the learned latent space supports objectives such as compression, clustering, anomaly detection, similarity computation, recovery, prediction, classification, and generation.

3.1. Autoencoders: Principles and variants

Autoencoders are a class of unsupervised neural network models designed to learn compact representations of data through reconstruction-based training. A standard autoencoder demonstrated in Fig. 3 consists of two main components: an encoder function $f(\cdot)$ that maps an input x to a latent representation z, and a decoder function $g(\cdot)$ that reconstructs the input from this latent code. The model is trained to minimize a reconstruction loss between the original input and its reconstruction, enabling the learning of low-dimensional embeddings that preserve the most informative characteristics of the data. Deep autoencoders were popularized by Hinton and Salakhutdinov [156], who demonstrated their effectiveness for nonlinear dimensionality reduction.

Building upon this basic architecture, numerous variants have been proposed to improve robustness, incorporate domain-specific structure, or enable probabilistic and generative modeling. As illustrated in Fig. 4, autoencoders can be organized along three main dimensions. The first dimension concerns the structural backbone, which defines the overall architecture of the encoder–decoder model. This dimension is explored in detail through the architectural variants presented later in this subsection. The second dimension concerns latent space constraints, which impose specific properties on the learned representation. The third dimension concerns training strategies, which modify the learning process to guide representation learning.

Together, these dimensions form a flexible design space for domain-specific adaptations, where architectural backbones, latent constraints, and training strategies are jointly selected based on the properties of the data and the desired characteristics of the learned representations. In the following, we focus primarily on the structural backbone dimension, providing a detailed analysis of the main architectural variants, while the remaining dimensions are discussed briefly as complementary design components.

The second dimension of the taxonomy concerns constraints imposed on the latent representation. These approaches aim to regularize the encoding space, improving properties such as sparsity, robustness, and structure. For example, sparse autoencoders encourage compact

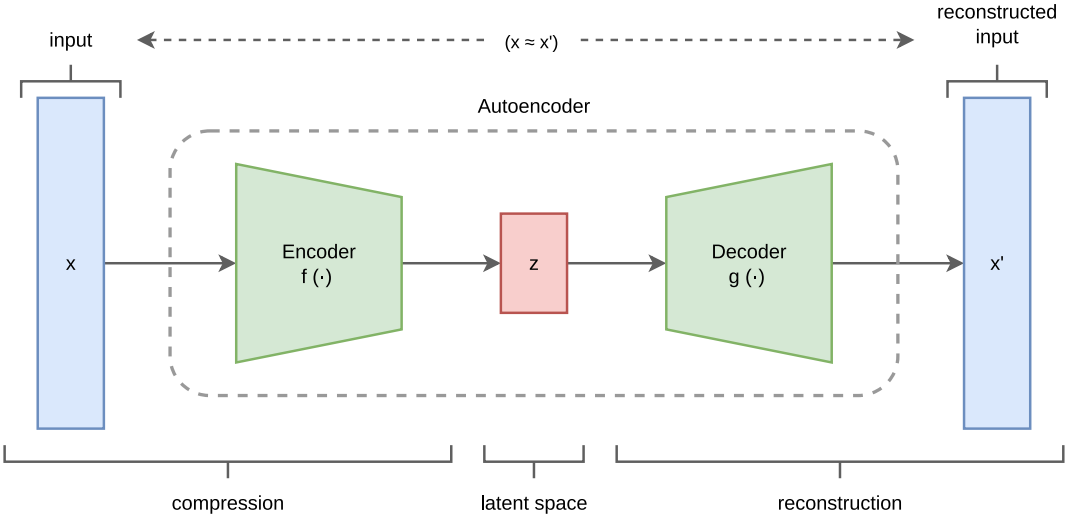

Fig. 3. Basic autoencoder architecture.

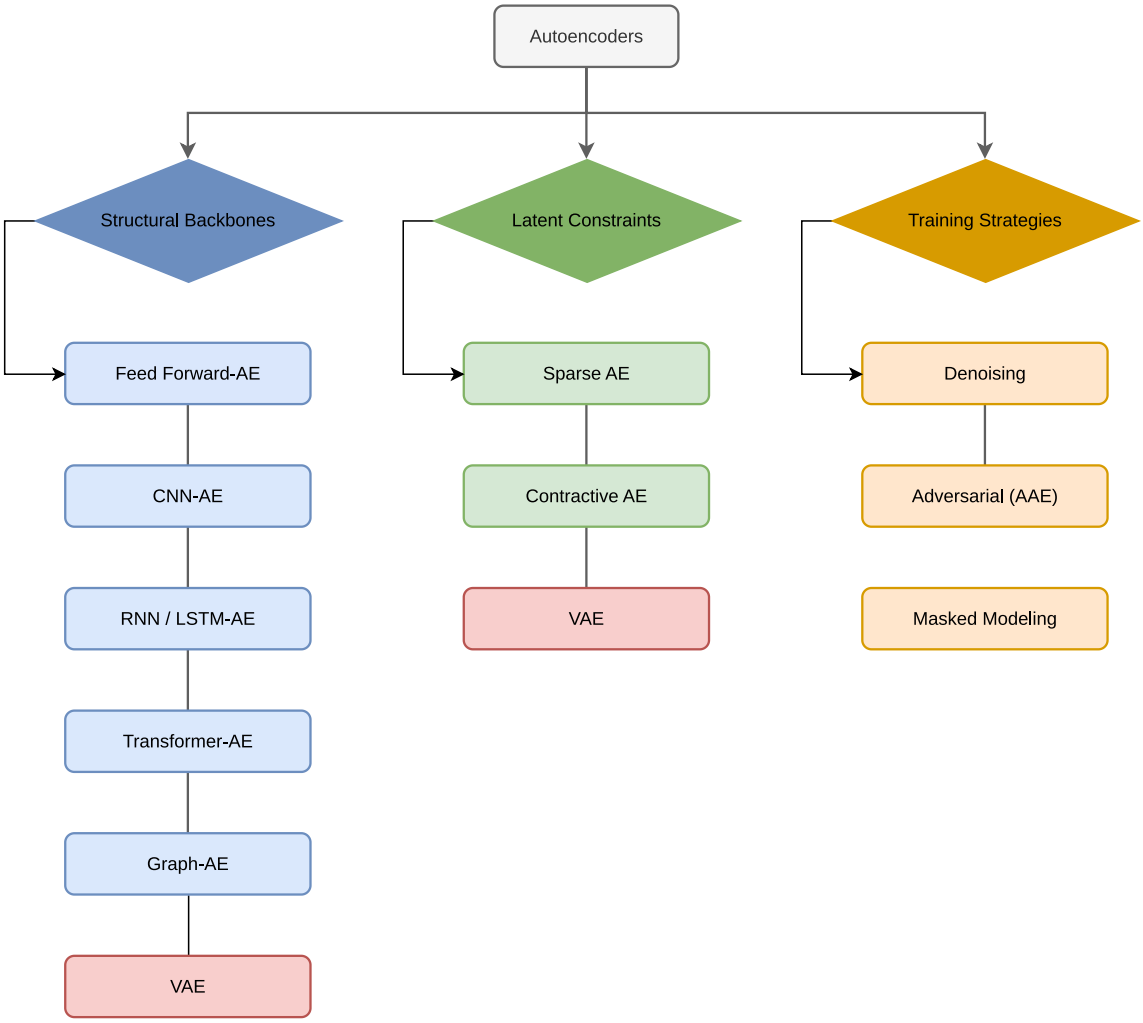

Fig. 4. Taxonomy of autoencoder models organized along three dimensions: structural backbones, latent space constraints, and training strategies. These components can be combined to design models tailored to specific data types and tasks.

representations by limiting neuron activation, while contractive autoencoders promote stability by enforcing local invariance to input perturbations. Probabilistic formulations, such as variational autoencoders, impose an explicit prior distribution over the latent space, enabling smooth latent structures and generative capabilities. Importantly, these methods do not fundamentally alter the encoder–decoder architecture, but instead shape the geometry and statistical properties of the learned representation.

The third dimension of the taxonomy relates to the training strategy, where the objective function or input transformation is modified to guide representation learning. Denoising autoencoders introduce controlled corruption to the input data and train the model to reconstruct the original signal, thereby improving robustness. Adversarial autoencoders incorporate a discriminator to align the latent distribution with a desired prior, enabling more structured representations. More recent approaches, such as masked modeling, rely on partial input reconstruction to encourage contextual understanding of the data. These strategies are largely architecture-agnostic highlighting that the learning objective plays a critical role alongside model design.

We now turn our focus to the first dimension of the taxonomy, namely the structural backbone of autoencoders, which defines the architectural design of the encoder–decoder model. The following paragraphs present the most commonly used architectural variants in the GeoAI literature and discuss their suitability for different types of trajectory data. It is worth noting that variational autoencoders span multiple dimensions of the taxonomy; although previously introduced from the perspective of latent space constraints, they are also discussed here due to their distinct architectural characteristics.

Feed-forward autoencoders. Feed-forward autoencoders represent the basic form of the architecture [156], consisting of fully connected layers in both the encoder and decoder. They operate on fixed-length vector inputs and learn global nonlinear transformations for dimensionality reduction. Despite their simplicity, they remain effective for structured tabular or flattened data representations.

Recurrent autoencoders (LSTM/GRU). Recurrent autoencoders [157] extend the architecture to sequential data by incorporating recurrent neural networks such as LSTMs or GRUs. These models capture temporal dependencies by encoding sequences into latent representations that preserve dynamic information.

Convolutional autoencoders. Convolutional autoencoders (CAEs) [158] replace fully connected layers with convolutional operations, enabling the model to exploit local spatial correlations and parameter sharing. They are particularly effective for grid-structured data such as images or spatial trajectory representations.

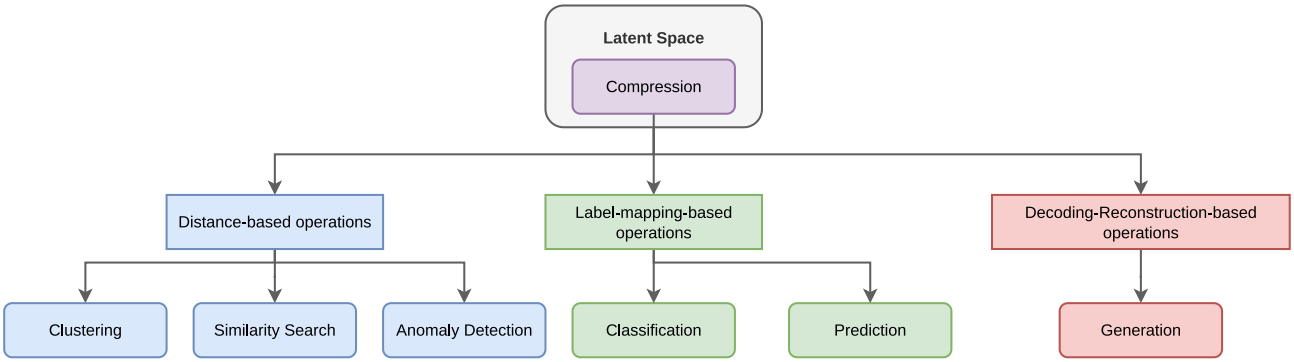

Fig. 5. Relationship between the latent space representation, the operations performed on it, and downstream tasks. Different tasks generally rely on specific mechanisms such as distance computation, label mapping and decoding.

Variational autoencoders. Variational Autoencoders (VAEs) [159] introduce a probabilistic framework in which inputs are encoded as distributions rather than fixed vectors. The model is trained using a combination of reconstruction loss and Kullback–Leibler divergence, enabling structured latent spaces and generative capabilities.

Transformer-based autoencoders. Transformer-based autoencoders leverage self-attention mechanisms [160] to model global dependencies within data. Unlike feed-forward or recurrent models, they process sequences in parallel and capture long-range interactions more effectively, making them suitable for complex spatio-temporal data [95].

Graph autoencoders. Graph autoencoders (GAEs) [161] extend the autoencoder framework to graph-structured data by using graph neural networks to encode node features and relationships. They are particularly useful for capturing relational dependencies and interaction structures.

3.2. Tasks of autoencoders in GeoAI

Autoencoders have been widely applied across a range of GeoAI tasks involving trajectory data, where their ability to learn compact and expressive latent representations enables efficient modeling of complex spatio-temporal patterns. The primary tasks addressed in the literature include compression, clustering, anomaly detection, similarity computation, trajectory prediction, classification, and trajectory generation. The relationship between these tasks and the learned latent representation is illustrated in Fig. 5. Rather than operating directly on raw data, most tasks rely on specific operations performed in the latent space, such as distance computation for similarity-based analysis, label mapping for supervised tasks, and decoding for generative applications. This perspective highlights the latent space as a unifying abstraction that supports diverse analytical objectives. While this taxonomy captures the most common mechanisms used to leverage latent representations, it is important to note that some approaches deviate from this structure by employing more specialized or hybrid strategies. Nevertheless, the presented framework provides a useful reference for understanding the dominant patterns in the literature. In the following, each task is discussed in detail, with emphasis on how the properties of the latent representation influence its effectiveness for different GeoAI applications.

Trajectory compression. Trajectory compression aims to reduce the storage and transmission costs of trajectory data while preserving essential structural information. Traditional approaches rely on geometric simplification techniques, whereas deep learning methods [4] leverage representation learning to achieve higher compression ratios with minimal information loss. In this context, autoencoders learn compact latent representations that encode trajectory structure [14], enabling efficient reconstruction and transmission.

Trajectory clustering. Trajectory clustering focuses on grouping trajectories with similar movement patterns to uncover underlying structures such as common routes or behavioral patterns. Classical methods depend on distance metrics and alignment techniques, while modern approaches increasingly rely on learned representations [137]. Autoencoders facilitate clustering by transforming trajectories into latent embeddings where similarity is more easily captured using conventional clustering algorithms [48], or by jointly integrating clustering objectives into the autoencoder training process [100].

Anomaly/outlier detection. Anomaly detection aims to identify trajectories or trajectory segments that deviate from normal patterns, which is critical for applications such as traffic monitoring and safety analysis. In GeoAI, this task is often formulated as learning the distribution of normal behavior [162]. Autoencoders are particularly effective in this setting [16,84], as abnormal trajectories typically exhibit higher reconstruction errors or lie in low-density regions of the latent space.

Trajectory similarity computation. Similarity computation involves measuring the resemblance between trajectories, forming the basis for tasks such as retrieval, clustering, and recommendation. Traditional approaches rely on handcrafted distance measures, which are often sensitive to noise and sampling inconsistencies [163]. Autoencoders address these limitations by learning latent representations [90] where similarity can be computed more robustly using standard distance metrics.

Trajectory prediction. Trajectory prediction focuses on forecasting future positions of moving objects based on historical observations, enabling applications such as traffic management and autonomous navigation [19]. This task inherently requires modeling temporal dependencies and uncertainty. Autoencoder-based approaches, particularly sequence-to-sequence and recurrent models [80,87], learn latent dynamics that capture motion patterns and support accurate and often multimodal predictions.

Trajectory recovery/reconstruction. Trajectory recovery aims to restore missing, fragmented, sparse, or low-sampling-rate trajectories by inferring unobserved movement states from partial observations. Unlike trajectory prediction, which focuses on future movement, recovery reconstructs missing parts of an already observed trajectory, such as intermediate GPS points, fragmented segments, or map-constrained road positions. Encoder–decoder and autoencoder-based methods address this task by learning spatio-temporal dependencies from incomplete trajectories and decoding more complete representations, often incorporating road-network constraints, attention mechanisms, or graph structures to improve spatial feasibility [17,18,92].

Trajectory classification. Trajectory classification aims to assign trajectories to predefined categories, such as transportation mode [56], activity type, or behavioral class. Autoencoders contribute by learning discriminative latent representations that capture the underlying structure of trajectory data, enabling effective differentiation between distinct behavioral patterns [21].

Trajectory generation. Trajectory generation involves synthesizing realistic movement data that follow learned spatio-temporal patterns. This is particularly important for simulation, data augmentation, and privacy-preserving data sharing [83]. Generative autoencoder variants, such as variational [111] and adversarial autoencoders, enable sampling from structured latent spaces to produce diverse and plausible trajectories.

Other emerging tasks. In addition to the primary tasks outlined above, autoencoders have been applied to more specialized problems such as gait recognition [9] and trajectory segmentation [93]. Although these tasks are well-explored in the broader literature, Refs. [164–166] respectively, they are less frequently addressed within the studies considered in this review, possibly reflecting differences in focus, problem formulation, data characteristics, or model design. Nevertheless, they further demonstrate the versatility of autoencoder-based representation learning in GeoAI applications.

4. Autoencoder architecture overview for trajectory tasks

This section reviews the main autoencoder architecture families identified in the trajectory GeoAI literature, emphasizing how each model type has been adapted to specific trajectory representations, tasks, and application domains.

4.1. Feed-forward autoencoders

(Autoencoders for fixed-length vectors (after flattening the trajectory))

Built from fully connected layers, feed-forward autoencoders provide a straightforward mechanism for nonlinear dimensionality reduction, enabling compact trajectory representations that facilitate compression and clustering. For example, Ref. [28] employs a deep dense autoencoder to compress high-dimensional trajectory-like time-series sensor measurements, demonstrating that latent representations combined with residual encoding can significantly reduce storage and transmission requirements while preserving reconstruction fidelity. Moving from pure compression to clustering-oriented applications, Ref. [27] integrates a stacked autoencoder within a preprocessing pipeline for aircraft trajectories, where resampled and normalized flight paths are encoded into low-dimensional embeddings that improve similarity estimation and ensemble clustering robustness.

Similarly, Ref. [29] adopts a two-stage framework in which a multilayer feed-forward autoencoder first learns compact trajectory embeddings, followed by Gaussian Mixture Model–based clustering in the latent space, enabling probabilistic assignment and more structured identification of dominant flight patterns. Using clustering as a method of anomaly detection, Ref. [30] leverages a deep autoencoder to fuse shallow geometric features, extracted from trajectory segments, into high-dimensional deep features, that are then decoupled from spatial attributes, and utilized in an unsupervised clustering framework to effectively detect trajectory segments that deviate from normal behavior. Trajectory recovery also appears in fully connected deep autoencoder designs when incomplete trajectories are represented as partially observed coordinate matrices. In this direction, Ref. [46] proposes a Hadamard deep autoencoder for reconstructing fragmented collective-motion trajectories. Instead of computing the reconstruction loss over all entries, the model incorporates a binary indicator matrix through a Hadamard product so that training is performed only on observed trajectory segments, while the decoder reconstructs the missing parts. This formulation extends standard deep autoencoder reconstruction toward missing-data recovery, especially for coordinated multi-agent motion where the trajectory matrix exhibits low-rank structure.

4.2. Recurrent autoencoders (RNN/LSTM/GRU)

(Autoencoders that handle sequential nature of trajectories)

Recurrent autoencoders extend feed-forward architectures by explicitly modeling the sequential dependencies of trajectory data, using LSTM-based encoders to preserve temporal correlations within compact latent representations. For instance, Ref. [14] employs a sequence-to-sequence LSTM autoencoder that compresses resampled 2D/3D trajectories into fixed-length latent vectors derived from the final hidden state, enabling adjustable compression ratios while preserving geometric structure through reconstruction-based training. Expanding this idea toward communication-efficient frameworks, Ref. [67] proposes two complementary strategies: a predictive LSTM model that transmits only initial trajectory segments and reconstructs the remainder under an error threshold, and a symmetric LSTM autoencoder that directly transmits low-dimensional latent embeddings instead of full trajectories. Recovery-oriented recurrent encoder–decoder models further show how sequence autoencoder principles can be adapted to missing trajectory data. Ref. [18] uses a GRU-based encoder–decoder with attention to reconstruct missing GPS trajectory points, incorporating trajectory curvature information to better preserve heading changes and applying a smoothing post-processor to reduce local reconstruction artifacts. This approach illustrates the suitability of recurrent encoder–decoder architectures for variable-length trajectory recovery, where the model must infer missing states from both preceding and following movement context rather than only compress a complete trajectory.

Shifting from compression to representation learning for clustering, Ref. [11] introduces a sequence-to-sequence recurrent autoencoder operating on motion-based behavioral descriptors extracted via sliding windows, producing latent embeddings that are subsequently clustered to group trajectories by movement dynamics rather than raw spatial similarity. This work [65] extends recurrent clustering frameworks by combining bidirectional LSTM encoders with hierarchical attention and contrastive learning to enhance latent space separability. Following data augmentation and positional encoding, trajectories are encoded with past–future context and optimized through a joint contrastive and clustering-oriented objective before applying K-means, yielding more discriminative and structurally organized embeddings than earlier recurrent autoencoder approaches.

4.3. Convolutional autoencoders

(Autoencoders for grid-like spatial trajectories.)

Convolutional autoencoders (CAEs) extend representation learning to structured trajectory data by exploiting local spatial or temporal correlations through convolutional filters. For example, Ref. [49] compares auxiliary and end-to-end deep clustering strategies for aircraft trajectories, contrasting post-hoc clustering on autoencoder embeddings, regularized using a t-SNE inspired term [167], with integrated clustering layers inspired by DCEC, thereby clarifying the distinction between embedding shaping and clustering-aware optimization. Complementing clustering-integrated approaches, Ref. [48] employs a convolutional autoencoder to learn grid-based embeddings of AIS vessel trajectories for similarity computation, treating clustering strictly as a downstream task performed in the learned latent space.

A closely related maritime similarity formulation is presented by Ref. [63], who transform AIS vessel trajectories into grid-based representations and train a convolutional autoencoder to extract compact feature vectors for trajectory similarity computation. After encoding the gridded trajectories, similarity is estimated through distances between the learned low-dimensional representations, reducing the dependence on exhaustive point-wise matching methods such as DTW and Fréchet distance. More recently, Ref. [64] extends this line of work through

VT-EncNet, a multimodal convolutional autoencoder that combines rasterized spatial trajectory masks with kinematic attributes such as speed and course. By using parallel 2D and 1D convolutional branches together with cross-attention-based feature fusion, the model learns latent representations that preserve both global route topology and local navigational dynamics, making it more suitable for vessel trajectory similarity computation in complex maritime traffic settings. Extending architectural design further, Ref. [47] proposes a Multi-feature Fusion Autoencoder with parallel convolutional branches encoding heterogeneous trajectory attributes (e.g., position and motion features), whose concatenated latent representations are subsequently clustered using conventional algorithms, highlighting the benefit of modular feature extraction without explicitly embedding clustering into the network.

4.4. Variational autoencoders (VAE)

(Autoencoders that provide probabilistic latent spaces, useful for generative tasks)

Building upon convolutional autoencoder architectures, Ref. [57] employs a convolutional variational autoencoder (CVAE) to encode highway vehicle trajectories into probabilistic latent representations. Centered around maneuver events and segmented into fixed temporal windows capturing both pre- and post-maneuver motion, trajectories are processed through a 1D convolutional encoder that outputs latent distribution parameters, with reconstruction and KL-regularization jointly shaping a smooth latent space where driving behaviors naturally organize for downstream clustering. Similarly, Ref. [50] utilizes a CVAE to achieve high-fidelity trajectory compression by encoding 6-second sequences of vehicle coordinates into a dense, low-dimensional context vector. By approximating the latent space with a standard normal distribution and incorporating a derivative-based smoothing term into the reconstruction loss, the model learns a continuous representation that inherently separates trajectories into distinct maneuver classes, such as lane-keeping and lane-changing, even without explicit supervision. Extending variational clustering, Ref. [100] similarly adopts a VAE with a latent mixture prior to model mobility behaviors, directly inferring soft cluster assignments from the learned probabilistic embeddings without external clustering algorithms.

4.5. Transformer-based autoencoders

While convolutional and recurrent architectures have proven effective for local feature extraction, they often struggle to capture the long-range global dependencies present in extended or complex trajectories. To address this, recent research has integrated self-attention mechanisms to learn more holistic spatio-temporal representations.

The authors of [93] introduce a Transformer-VAE framework for airspace trajectory clustering, combining segmentation, self-attention–based encoding of long-range dependencies, density-based initialization, and KL-guided joint optimization to align representation learning with cluster structure in an end-to-end manner. Adapting the masked autoencoder (MAE) framework for autonomous driving, Ref. [95] introduces Traj-MAE, a transformer-based architecture that applies multi-granular masking across trajectories and High-Definition maps to inherently learn complex social interactions and environmental topologies without human annotations. For trajectory recovery, Ref. [99] proposes RNTrajRec, a road-network-enhanced spatial-temporal Transformer framework. The model first learns road-segment representations through GridGNN, then represents each GPS point using a surrounding road-network subgraph and encodes the resulting spatial-temporal structure with GPSFormer. The Transformer component captures long-range temporal dependencies, while the graph-based road representation constrains recovery toward spatially feasible trajectories. Although the framework is not a conventional autoencoder, it follows an encoder-decoder recovery formulation and demonstrates how Transformer-based representation learning can be coupled with road-network structure for low-sampling-rate trajectory reconstruction.

Specifically designed to handle observational noise and information loss, Ref. [97] proposes a U-type robust sparse attention autoencoder (URSAA) that integrates structural skip connections with sparsity penalties and an attention mechanism to extract salient, noise-resilient movement features for robust trajectory clustering. Finally, discarding explicit spatial topologies in favor of global self-attention, Ref. [98] introduces the Trajectory Unified Transformer (TUTR), which utilizes an implicit mode-level encoder and cross-attention mechanisms to natively model social interactions and generate diverse, socially-aware trajectories in a single forward pass, thereby eliminating the computational bottleneck of expensive post-hoc clustering.

4.6. Graph autoencoders

To capture relational dependencies and multi-agent interactions, Ref. [113] presents a Spatio-Temporal Graph Auto-Encoder (STGAE) that combines spatial graphs and temporal convolutions to model dynamic traffic scenes. Transitioning from patch-based masking to relational structures, the framework bypasses standard reconstruction metrics by applying Kernel Density Estimation (KDE) directly to the latent space to flag abnormal driving behaviors as low-density outliers, effectively capturing the structural deviations of anomalous maneuvers. Addressing topological noise in multi-agent systems, Ref. [114] proposes MGAE-MDRI, a masked graph autoencoder model that integrates a Graph Attention Network to guide a preference random walk sampler. This architecture selectively masks and reconstructs redundant interaction edges, stabilizing dynamic relational inference to achieve highly accurate trajectory predictions. Graph-based recovery models extend this relational view from multi-agent interactions to road-network constrained trajectory completion. The authors of [115] propose MM-STGED, a micro–macro spatial-temporal graph-based encoder–decoder for map-constrained trajectory recovery. Instead of treating a trajectory only as a sequence, the method constructs a trajectory graph whose nodes represent observed GPS points and whose edges encode relative spatial-temporal relations between them. A graph-based decoder then incorporates macro-level information, including shared travel preferences and road conditions, to guide the recovery of missing map-constrained trajectory states.

4.7. Hybrid autoencoders

Hybrid autoencoders integrate multiple architectural paradigms, such as self-attention mechanisms, convolutional filters, recurrent layers, or probabilistic modeling, to simultaneously capture diverse spatio-temporal characteristics and long-range dependencies within complex trajectory data.

Advancing architectural design in this direction, Ref. [83] merges sequence-to-sequence LSTMs with variational inference to create a Sequential Variational Autoencoder (SVAE) for urban mobility. By compressing variable-length GPS trajectories into an intermediate state that parameterizes a Gaussian latent space, the framework enables both accurate path reconstruction and the synthesis of realistic, privacy-preserving mobility patterns via tunable KL-divergence regularization. Building directly upon the recurrent-variational structure of the SVAE, Ref. [84] replaces the standard unimodal prior with a Gaussian Mixture Model (GMM) to create a GM-VSAE. This architectural shift allows the recurrent encoder-decoder to map complex trajectories into a multimodal latent space, inherently isolating distinct route types into separate probabilistic clusters, enabling the generative network to evaluate reconstruction likelihoods on the fly for constant-time anomaly detection. For explicit spatio-temporal feature extraction, Ref. [20] uses a Convolutional LSTM (ConvLSTM) that directly processes pedestrian environments as 3D tensors. Instead of conventionally flattening spatial data for 1D recurrent processing, this architecture uses convolutional transitions within the LSTM cells to inherently preserve grid-like interactions across time steps and capture complex crowd dynamics.

In order to combat the severe observational noise inherent in skeletal joint tracking, Sheng and Li [9] create a Siamese Denoising Autoencoder (Siamese DAE) that utilizes a weight-sharing, LSTM-based twin-network to process paired clean and corrupted trajectories in parallel, explicitly forcing noisy inputs into the same robust latent representation as their clean counterparts to isolate pure biomechanical motion dynamics from sensor corruption. The work [41] introduces CMAE-Traj, a contrastive masked autoencoder framework that employs a temporal-social mask module to decouple short-term dynamics from long-term trends, utilizing a contrastive learning objective to refine the interactions between agents and their environment for more robust trajectory forecasting. Lastly, trajectory recovery also motivates explicitly hybrid autoencoder designs. Ref. [17] introduces a Map-informed Adaptive Spatio-Temporal Autoencoder for recovering coarsely sampled trajectories under road-network constraints. The model combines an encoder–decoder backbone with a pre-trained attributed network embedding module to inject road-segment topology into the trajectory representation, and an adaptive mask inference module to restrict and weight candidate road segments during decoding. This architecture illustrates how recurrent sequence modeling, road-network representation learning, attention-based inference, and autoencoder-style reconstruction can be integrated for map-informed trajectory recovery.

5. Applications and use cases

This section provides an overview of Geo-AI applications of autoencoders, organizing the reviewed studies according to the characteristics of the trajectory data and their respective domains of use. The following subsections highlight representative use cases spanning urban mobility and smart-city systems, maritime navigation, aviation flow analysis, and interaction-aware environments, illustrating the breadth of Geo-AI applications supported by autoencoder-based trajectory modeling.

5.1. Urban mobility & smart-city applications

Urban mobility and smart-city applications leverage trajectory data collected from transportation systems, mobile devices, and indoor positioning infrastructures to understand movement patterns, improve infrastructure management, and support data-driven urban services.

Addressing indoor mobility analysis in smart buildings, Ref. [52] converts indoor occupant trajectories into rasterized representations and employs a convolutional autoencoder to learn similarity-preserving embeddings that enable clustering of occupant movement behaviors for data-driven building management. Extending trajectory similarity modeling to large-scale urban mobility analytics, Ref. [13] proposes the seq2seq-based t2vec framework, which learns spatially aware trajectory embeddings robust to non-uniform sampling and noise, enabling scalable similarity search and trajectory clustering in large real-world transportation datasets. A related approach for similarity computation is presented in Ref. [90], where the At2vec model incorporates spatial, temporal, and activity semantics from location-based social network trajectories to learn representations robust to heterogeneous sampling for activity-aware mobility analysis.

Building on deep representation learning for mobility analysis, Ref. [35] addresses the problem of discovering latent mobility patterns and fine-grained behavioral groups in large-scale trajectory datasets. To this end, they introduce DETECT, a stacked autoencoder-based framework that jointly optimizes trajectory representation learning and clustering. In order to enable the discovery of frequently used urban travel routes from large-scale taxi trajectory data, Ref. [89] similarly, proposes a Deep Trajectory Clustering (DTC) framework that combines sequence-to-sequence autoencoder representations with joint clustering optimization to learn clustering-friendly trajectory embeddings. Urban mobility applications also require trajectory recovery when raw GPS traces are sparse, sampled at low rates, or affected by communication loss. This problem is especially important in intelligent transportation systems because many downstream services, such as vehicle navigation, travel-time estimation, and driver behavior analysis, rely on fine-grained trajectories aligned with the road network. Ref. [92] addresses this issue through MTrajRec, a map-constrained trajectory recovery framework that jointly recovers fine-grained missing points and maps them onto the road network. By predicting road segment identifiers and moving ratios through a Seq2Seq multi-task learning architecture, the method avoids the error accumulation of two-stage pipelines that first reconstruct trajectories and then apply map matching.

Addressing the challenge of identifying abnormal pedestrian movement in complex urban backstreets, Ref. [10] focuses on anomaly detection in CCTV-derived trajectories, where movement patterns are irregular and strongly shaped by spatial and environmental context. This is particularly important for urban safety monitoring in dense, unstructured environments. They propose an unsupervised variational autoencoder-based framework that learns typical movement patterns while incorporating spatial context, enabling both detection and contextual interpretation of anomalous behaviors. Exploring representation learning for large-scale urban mobility data, Ref. [21] focuses on the analysis of human mobility behaviors from large-scale GPS trajectory datasets. To this end, they propose a Dual Convolutional Supervised Autoencoder (Dual-CSA) framework that incorporates multiple spatio-temporal trajectory features, including movement characteristics, stop states, and turning behaviors, together with recurrence plot representations to capture temporal dynamics.

By learning structured latent representations through a supervised autoencoder with predefined class centroids, the approach enables data-driven mobility behavior analysis. In the same theme, Ref. [56] addresses the extraction of urban mobility patterns from large-scale GPS trajectory data, particularly in settings where labeled data are limited. They propose a Semi-Supervised Convolutional Autoencoder (SECA) framework that transforms raw trajectories into multi-channel representations capturing motion characteristics such as speed, acceleration, and jerk, allowing the model to learn spatio-temporal mobility patterns directly from the data. By combining convolutional autoencoding with semi-supervised learning, the approach enables effective pattern extraction under limited supervision.

Shifting focus to abnormal movement detection in urban traffic, Ref. [34] addresses anomaly detection in GPS-derived vehicle trajectories for large-scale traffic monitoring, where identifying unusual driving patterns is essential for detecting potential risks such as accidents, congestion irregularities, or unsafe maneuvers. This is particularly important in real-world traffic management systems, where labeled anomalies are scarce and behaviors vary across different urban contexts. Within this setting, they formulate the problem as an unsupervised representation learning task, where a stacked autoencoder captures normal spatio-temporal driving patterns and flags deviations through reconstruction errors. Focusing on driver-level risk profiling, Ref. [39] addresses the problem of evaluating anomalous driving behaviors from trajectory data, with applications in traffic safety monitoring and insurance risk assessment, where understanding long-term driving habits is crucial for identifying high-risk drivers. Unlike approaches that focus on individual trajectories, their work emphasizes modeling driver behavior across multiple trips under varying spatio-temporal conditions. In this context, they employ an unsupervised autoencoder to model driving habits relative to their spatio-temporal context, enabling the identification of risky behaviors without manually labeled data. To support urban traffic risk monitoring, Ref. [101] focuses on identifying potential hazards like accidents or road emergencies by detecting anomalous taxi movements, such as sharp turns or abrupt acceleration, within dense road networks. For that, they utilize a Variational Autoencoder (VAE) to learn the probability distribution of normal driving patterns, flagging anomalies as deviations from the learned latent structure to enable scalable detection without the need for direct trajectory comparisons.

Addressing urban traffic management within smart cities, Ref. [102] identifies anomalous vehicle trajectories, such as unusual detours or infrequent routes, in real-time as GPS data is generated. This online

detection capability allows for the immediate flagging of irregularities that deviate from typical movement patterns, helping to monitor traffic risks more effectively. The authors propose GeoGNFTOD, a model that fuses geographic and graph-based representations into an LSTM-based Variational Autoencoder to identify anomalies through reconstruction-based deviation. Lastly, aiming to enhance traffic monitoring and public safety, Ref. [86] focuses on the robust detection of unusual object behaviors within video surveillance data. By utilizing object-level trajectories rather than raw video frames, the approach provides long-term behavioral information while significantly reducing the computational resources required for motion analysis. Implementing an RNN-based autoencoder to learn trajectory representations, anomalies can be identified through a reconstruction-error-based distance metric within a nearest-neighbor framework.

5.2. Maritime navigation & infrastructure protection

Maritime navigation and infrastructure protection increasingly rely on trajectory analysis to detect abnormal vessel behaviors that may signal navigational hazards, illegal activities, or threats to critical port infrastructure.

In this context, Ref. [53] enables the precise detection of route deviations, abrupt speed changes, and irregular maneuvers in complex port environments. This is achieved by reformulating AIS-based vessel trajectory monitoring as an image reconstruction problem, processing multidimensional trajectory data as trajectory maps through a multi-scale convolutional autoencoder and identifying anomalies via structural dissimilarities. Extending reconstruction-based anomaly detection to complex maritime infrastructures, Ref. [94] addresses anomaly detection in AIS trajectories in cross-sea bridge areas, where navigational constraints and collision risks are especially critical. They propose a Transformer–VAE framework that leverages self-attention to capture long-range spatio-temporal dependencies and models a probabilistic latent distribution, enabling robust identification of subtle abnormal behaviors. Adding to the prior approaches, Ref. [104] prioritizes the monitoring of complex port and coastal environments where high traffic density and overlapping transit lanes create a sophisticated web of “normal” behaviors. This use case is specifically designed to distinguish subtle anomalies such as unauthorized waterway crossings, abrupt heading changes, or route deviations, within these congested settings where traditional, single-pattern monitoring often fails. To address this complexity, the authors employ a Gaussian Mixture Variational Autoencoder (GMVAE) that identifies anomalies by measuring reconstruction deviations against a learned multimodal distribution of ship trajectories, allowing it to simultaneously model diverse vessel patterns that standard VAEs would struggle to capture.

To address the critical need for proactive anomaly detection in complex maritime environments like congested port waters, Ref. [69] developed a framework designed to identify potential navigation conflicts before they escalate. The core of this system is a Gated Recurrent Unit (GRU) neural network that models historical vessel trajectories to predict future positions with high temporal efficiency. By integrating DBSCAN for route extraction and specific filtering techniques, the authors effectively reduce data redundancy, allowing the model to focus on the most relevant patterns for maritime safety. Focusing on the reliability of predicted trajectories, Ref. [87] proposes a trajectory forecasting framework specifically for collision avoidance and risk assessment in complex navigation areas like crossroads. The primary use case supports intelligent surveillance systems and autonomous ships by introducing an attention-based VarLSTM encoder-decoder framework. Using Bayesian modeling to quantify predictive uncertainty and integrating high-level ship intentions with real-world AIS data, the approach enables reliable forecasts and anomaly detection in high-traffic zones where multiple future paths are possible. For the purpose of enhancing maritime situational awareness, Ref. [40] focuses on the early identification of high-risk encounters to provide navigators and autonomous systems with actionable guidance for collision avoidance. This use case is supported by a dual linear autoencoder framework that jointly predicts entire vessel trajectories from historical AIS data over a 30-minute horizon. By quantifying positional uncertainty alongside these predictions, the method enables more informed decision-making for safe navigation in complex maritime environments.

5.3. Aviation flow analysis, safety & simulation

Aviation flow analysis and airspace risk assessment aim to enhance operational safety by detecting abnormal flight behavior and emerging traffic risks from large-scale trajectory data.

The paper Ref. [31] presents a solution for unsupervised aviation flow monitoring and airspace risk assessment by developing a method to simultaneously identify typical traffic patterns and detect flight anomalies like unexpected holding patterns. Their solution utilizes an end-to-end deep autoencoder (DAE) framework that incorporates a structured sparsity-inducing norm to isolate outliers and cluster assignment hardening to refine low-dimensional, spatial-temporal representations leading to superior performance in capturing fine-grained clusters and spatial anomalies. The work presented in Ref. [33] provides a framework for aviation flow monitoring and airspace safety analysis by detecting tactical controller interventions such as vectoring and speed changes. These insights into real-world traffic dynamics allow for a better understanding of controller workload and the statistical probability of rare safety events. By quantifying these human-induced deviations, the research supports the development of more realistic training simulations and optimized airspace designs. This is done with an autoencoder-based framework that flags trajectories producing a high reconstruction error. Building upon this, Ref. [32] further investigates aviation flow monitoring and airspace safety analysis by focusing on the interpretation of anomalous trajectories to infer air traffic control (ATC) actions and assess contextual traffic complexity. In particular, they aim to characterize controller interventions and airspace workload dynamics based on deviations from nominal flight behavior. To achieve this, they extend the previous framework by modeling nominal route behavior through feed-forward autoencoders and analyzing reconstruction-error-based anomalies within specific city-pair flows, linking anomaly detection to operational risk indicators.

Shifting focus to abnormal movement detection in aviation systems, Ref. [74] addresses on the challenge of identifying atypical flight behaviors from airport-level ADS-B trajectory data, where many safety-critical anomalies cannot be adequately predefined in advance. This is particularly important for flight data monitoring and aviation safety analysis, as undetected irregular maneuvers may indicate potential risks or precursors to accidents. In order to enable early identification of such behaviors, an LSTM-based autoencoder framework is developed, that models normal trajectory profiles and detects abnormal maneuvers through reconstruction error thresholds. In order to identify operationally significant deviations in aircraft altitude and speed during critical flight phases without the need for labeled events, Ref. [54] addresses anomaly detection in high-dimensional Flight Operational Quality Assurance (FOQA) time-series data. This problem is central to modern aviation safety, where increasing system complexity and the lack of predefined anomaly definitions make it difficult to proactively detect emerging risks using traditional rule-based or supervised approaches. Within this context, they employ a Convolutional Variational Auto-Encoder (CVAE) to model multivariate flight data and detect safety-critical anomalies. Addressing the need for high-fidelity safety simulations, Ref. [58] focuses on generating realistic 4-dimensional flight trajectories for mid-air collision risk assessment in high-traffic terminal areas. They introduce a Temporal Convolutional Variational Autoencoder (TCVAE) that incorporates Temporal Convolutional Networks (TCNs) to capture long-range temporal dependencies and utilizes a Variational Mixture of Posteriors (VampPrior) to model complex multimodal distributions, producing

diverse, physically plausible synthetic trajectories that account for variability induced by air traffic control interventions.

5.4. Interaction-aware trajectory modeling

Interaction-aware trajectory modeling focuses on capturing the complex dependencies among multiple agents, such as pedestrians, robots, and vehicles, to enable socially compliant navigation and safe motion planning in dynamic, shared environments.

5.4.1. Human and crowded-space interactions

Modeling social interactions in crowded environments is essential for understanding collective movement patterns and enabling socially-aware navigation. To this end, Ref. [80] addresses the problem of predicting pedestrian trajectories in complex scenes, where diverse motion patterns and destination intentions lead to highly multimodal behaviors. They introduce an LSTM-based framework that employs automatic route-class clustering to group pedestrians by latent motion patterns, training cluster-specific sequence-to-sequence models to improve trajectory prediction. Addressing the challenge of predicting pedestrian motion in dynamic and highly interactive environments, Ref. [70] aims to improve the stability and reliability of multimodal trajectory predictions across diverse crowd scenarios, where varying interaction patterns can lead to inconsistent temporal modeling. They introduce an Adaptive Forgetting-Controlled RNN (AFC-RNN) that regulates temporal forgetting through self-attention-derived memory factors, yielding more stable pedestrian motion predictions.

To improve pedestrian trajectory forecasting from a first-person perspective, Ref. [75] tackles the challenge of predicting pedestrian motion in autonomous driving scenarios, where accurate anticipation of crossing behavior is critical for avoiding collisions and ensuring road safety. This task is particularly challenging due to the complex interactions between pedestrians and vehicles, as well as the need to integrate multiple cues such as vehicle motion, pedestrian intention, and scene dynamics. They introduce a Holistic LSTM that adaptively incorporates vehicle speed, global scene dynamics, and crossing intentions into specialized memory cells to model these interactions. For socially compliant robot navigation in crowded environments, Ref. [79] proposes an RNN encoder–decoder framework that models both human–human and human–space interactions. The approach incorporates complementary social and physical attention mechanisms to capture the influence of nearby pedestrians and surrounding scene context, enabling the prediction of collision-free and socially aware trajectories.

Capturing the inherent unpredictability of human motion is essential for reliable trajectory prediction in real-world environments. For this purpose, Ref. [112] introduces SocialVAE, a stochastic generative framework designed for both pedestrian navigation and highly dynamic scenarios such as sports. The model leverages a timewise variational autoencoder with social attention to account for human–human interactions and generate diverse, multimodal future trajectories, further enhanced by backward posterior approximation and final-position clustering. Enabling agents to anticipate diverse human responses in scenarios such as traffic weaving and crowded sidewalks, Ref. [111] adopts a Conditional Variational Autoencoder (CVAE) framework for interaction-aware trajectory prediction. By conditioning predictions not only on past interaction history but also on candidate future robot actions, the approach captures the coupled dynamics between human and robot behaviors. This allows the model to generate multimodal distributions over future human trajectories, reflecting the multiple plausible ways humans may react in interactive settings. Such probabilistic reasoning supports proactive and safe robot planning by explicitly modeling uncertainty and variability in human decision-making. Scaling these generative models to heterogeneous crowds, Ref. [59] introduces DCENet, a CVAE-based framework that utilizes self-attention to model dynamic spatial-temporal interactions, producing multimodal trajectory sets for pedestrians, cyclists, and vehicles in mixed traffic. To produce a diverse "fan" of socially-acceptable paths for heterogeneous agents, AMENet [106] models multimodal future trajectories in mixed traffic environments. It achieves this by employing a conditional variational autoencoder (CVAE) framework, where self-attention is used to selectively focus on salient interaction features across agents. In particular, "Attentive Dynamic Maps" are introduced to prioritize key aspects of agent behavior for interaction-aware trajectory prediction.

5.4.2. Vehicle and traffic-stream interactions

Moving from social spaces to structured roads, vehicle interactions are defined by driving rules and maneuvers rather than fluid social norms. In highway scenarios, this translates into anticipating discrete driving decisions to ensure safe motion planning for autonomous vehicles in high-speed traffic. Ref. [91] focuses on this setting by predicting whether a driver intends to change lanes or stay in the current lane. The model then uses a conditional variational autoencoder (CVAE) to generate multiple plausible future trajectories conditioned on the inferred intention, enabling the system to represent uncertainty in driver behavior. Complementing intention-based methods, uncertainty-aware trajectory prediction is used to support safe decision-making in dynamic highway traffic. In these cases, accurate modeling of surrounding vehicle motion over time is essential for robust autonomous driving, particularly under noisy or incomplete observations. Ref. [82] addresses this by generating naturalistic future trajectories while also estimating prediction uncertainty for surrounding vehicles. The approach is based on a recurrent variational autoencoder (VAE), which captures temporal traffic behavior and leverages its generative structure to produce realistic multimodal motion predictions. In fast, highway driving scenarios, anticipating the maneuvers of leading vehicles is essential for enabling autonomous systems to make safe tactical decisions, such as braking or overtaking. In this context, Ref. [88] evaluates the capacity of intelligent vehicles to predict leader–follower dynamics by forecasting longitudinal velocity and position. Through a comparative analysis of LSTM, GRU, and Stacked Autoencoder (SAE) architectures, the study identifies LSTM as the most effective model for this task.

In complex urban traffic environments, real-time and context-aware trajectory prediction is critical for enabling autonomous systems to safely navigate among heterogeneous agents such as vehicles, pedestrians, and cyclists. In this setting, Ref. [108] focuses on generating accurate and map-compliant multimodal trajectory predictions under strict latency constraints. To achieve this, they implement ContextVAE, a lightweight timewise VAE that integrates semantic map information and neighboring agent states through a dual-attention mechanism, enabling high-fidelity predictions in under 30 ms. In high-speed highway scenarios, accounting for interaction dynamics and environmental constraints is essential for generating safe and risk-aware trajectory predictions, particularly when surrounding vehicles and road boundaries impose varying levels of driving risk. In this context, Ref. [61] proposes a Driving Risk Map-based framework that models these constraints explicitly, utilizing a CVAE to generate socially plausible trajectories conditioned on quantified risk distributions derived from the environment and neighboring vehicles.

Complementing interaction-driven prediction with long-term behavioral planning, Ref. [110] addresses the challenge of forecasting trajectories in complex environments where agents must navigate toward uncertain goals while avoiding obstacles such as buildings and road boundaries. To address this, they introduce MUSE-VAE, a multi-scale framework that follows a coarse-to-fine strategy by first predicting high-level destination goals and subsequently generating detailed, multimodal trajectories that remain physically plausible and environmentally compliant. In highway driving scenarios where safety is paramount, ensuring the trustworthiness and physical validity of predicted trajectories is essential for reliable decision-making. For this purpose, Ref. [103] introduces a Descriptive Variational Autoencoder (DVAE) for interpretable and safety-aware trajectory prediction in safety-critical driving scenarios, which embeds vehicle-dynamics equations directly into the decoder,

forcing the latent space to encode physically meaningful parameters such as longitudinal acceleration and lane-change behavior. This design enables unsupervised trajectory learning while supporting rule-based validation of model outputs for trustworthy decision-making.

5.5. Synthetic validation and latent pattern intelligence

Expanding into the domain of trajectory intelligence, VAEs facilitate a comprehensive lifecycle of movement data, spanning from the creation of privacy-preserving synthetic datasets to the discovery of complex behavioral patterns and structural outliers.

Reducing the reliance on costly real-world testing is a central challenge in the safety validation of autonomous driving systems, particularly when evaluating rare and safety-critical scenarios. Within this context, Ref. [85] develops a recurrent autoencoder-based framework that learns latent representations of driving trajectories to generate realistic, variable-length synthetic data. Beyond generation, the learned latent space is further exploited for clustering and outlier detection, enabling the identification and categorization of rare maneuvers for scenario-based verification. Similarly, limited access to large-scale trajectory data, often due to privacy constraints or regional restrictions, poses a significant barrier to the development and evaluation of data-driven mobility systems. To address this, Ref. [22] introduces TrajVAE, an LSTM-based generative framework designed to produce high-fidelity synthetic trajectories that preserve the statistical characteristics of real-world data. This capability supports both the training of autonomous systems and large-scale urban traffic analysis in environments where direct data collection is impractical.

Shifting focus from synthetic data generation to latent pattern intelligence, detecting structural irregularities in real-world trajectory data becomes essential for identifying abnormal driving behaviors and sensor-related anomalies without relying on rigid similarity measures. In this context, Ref. [16] introduces a hybrid framework that combines a denoising sequential autoencoder with a density-based Local Outlier Factor (LOF) algorithm to enable data-driven anomaly detection. By leveraging reconstruction errors computed through a Haversine-weighted loss function, the approach effectively captures irregular movement patterns and GPS anomalies while avoiding dependence on predefined spatial similarity rules. Pivoting from anomaly detection to structural comparisons, Ref. [13] shifts the focus, to enabling scalable similarity search and large-scale trajectory clustering in real-world transportation systems. This is achieved through t2vec, a proposed seq2seq-based deep representation learning framework that embeds trajectories into spatially-aware vectors with linear-time retrieval complexity. Ultimately, these learned embeddings are highly robust to non-uniform sampling and low-frequency GPS noise. To enable the discovery of universal mobility patterns across highly distributed datasets, Ref. [12] introduces a space and time invariant clustering framework that accounts for significant spatio-temporal shifts. This approach allows for the identification of identical traffic maneuvers even when they occur in completely different cities or at different times. By coupling a sliding-window feature extraction module with a sequence-to-sequence autoencoder, the authors translate raw trajectories into fixed-length behavioral representations, effectively isolating intrinsic movement dynamics from absolute geographical coordinates.

6. Comparative synthesis of autoencoder architectures & tasks for trajectory GeoAI

This section provides a comparative synthesis of the reviewed corpus from descriptive, methodological, and practical perspectives. Section 6.1 first examines publication and citation trends to contextualize the growth and visibility of autoencoder-based trajectory GeoAI. Section 6.2 then analyzes the distribution of autoencoder architecture families across trajectory-analysis tasks and application domains, while Section 6.3 discusses the task-specific representation requirements that shape model suitability. Section 6.4 provides a formal cross-architecture analysis of the main inductive biases, strengths, and limitations of each architecture family, and Section 6.5 summarizes architecture–task suitability through a qualitative comparison. Section 6.6 presents dataset-based metric comparisons where meaningful benchmark overlap exists, and Section 6.7 concludes with practitioner-oriented guidelines for task-aware architecture selection.

6.1. Publication and citation trends in the reviewed corpus

To supplement the taxonomy-based analysis with descriptive statistical evidence, we examined the publication and citation patterns of the papers included in the reviewed corpus. Fig. 6 presents the number of reviewed papers by publication year. The corpus contains 77 papers published between 2017 and 2026, with a clear increase in activity after 2018 and the highest number of included studies appearing in 2020. Overall, 63 of the 77 reviewed papers were published from 2020 onward, indicating that autoencoder-based trajectory GeoAI has been particularly active in recent years. The lower values observed for 2024–2026 should be interpreted cautiously, as they may reflect the strict inclusion criteria of this survey, indexing delays, and the fact that 2026 is an incomplete year.

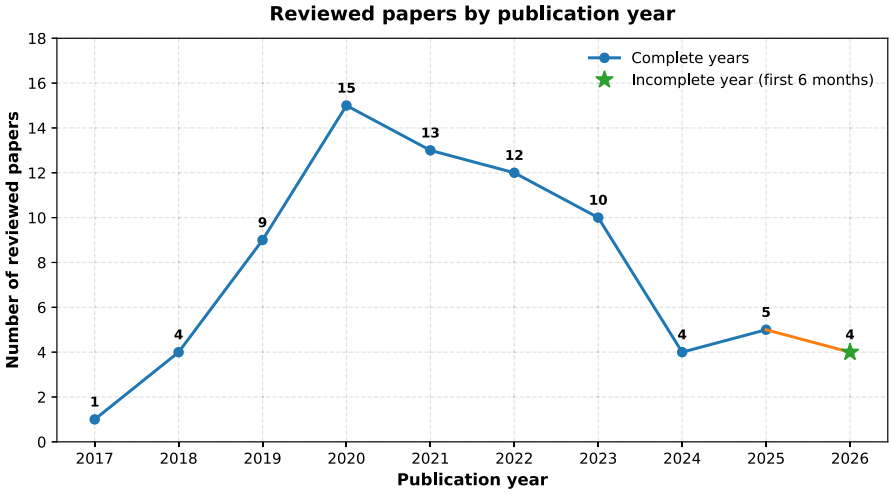

Fig. 6. Number of reviewed papers by publication year. The value for 2026 corresponds to an incomplete year.

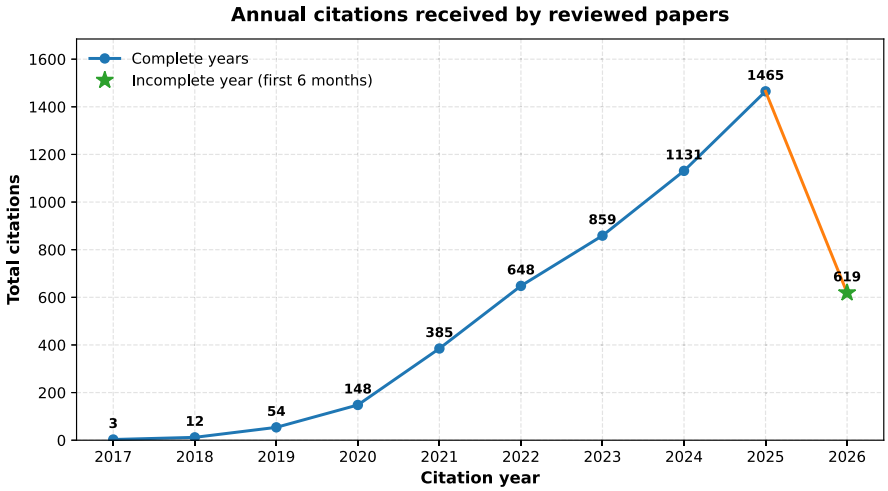

Fig. 7. Annual citations received by the reviewed papers. The value for 2026 corresponds to the first six months of the year and should be interpreted as a partial-year count.

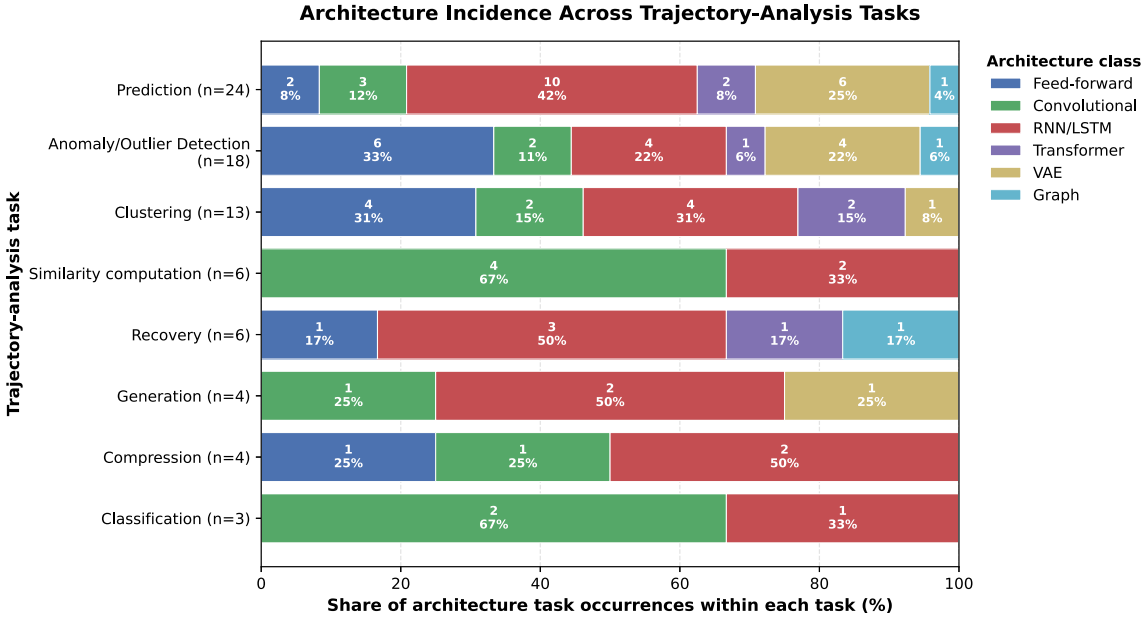

Fig. 8. Relative distribution of autoencoder architecture classes within each trajectory-analysis task. Each horizontal bar sums to 100% for a given task, while the labels inside the segments report the corresponding number of architecture–task occurrences and percentage share. Counts represent methodological incidence in the reviewed literature rather than direct performance comparisons, as the studies differ in datasets, preprocessing choices, baselines, and evaluation protocols.

Fig. 7 shows the annual citations received by the reviewed papers. The citation counts increase substantially over time, from only a few citations in the early years of the corpus to more than one thousand annual citations in both 2024 and 2025. This trend suggests that the reviewed literature has gained increasing visibility and scholarly attention. The value reported for 2026 should also be interpreted as a partial-year count. Together, these publication and citation patterns provide additional evidence for the relevance and timeliness of autoencoder-based methods in trajectory GeoAI, while the following distribution analyses further characterize the corpus in terms of architectures, tasks, use cases, datasets, and application domains.

6.2. Distribution of autoencoder architectures across tasks and application domains

To provide a quantitative overview of the reviewed literature before moving to the detailed cross-architecture and task-specific analysis, we first examine how autoencoder architecture classes are distributed across trajectory-analysis tasks and application domains. The purpose of this distributional analysis is not to establish direct performance superiority, since the reviewed studies differ substantially in datasets, preprocessing strategies, evaluation metrics, and experimental protocols. Instead, the distributions provide a descriptive view of methodological tendencies in the literature, showing which architecture families are most frequently associated with particular trajectory problems and data contexts.

Fig. 8 summarizes the distribution of autoencoder architecture classes across major trajectory-analysis tasks. Prediction is the most represented task, reflecting the strong role of sequence modeling, uncertainty estimation, and interaction-aware forecasting in recent trajectory research. Within this task, RNN/LSTM-based architectures account for the largest share, followed by VAE-based models, indicating the importance of temporal dependency modeling and probabilistic latent

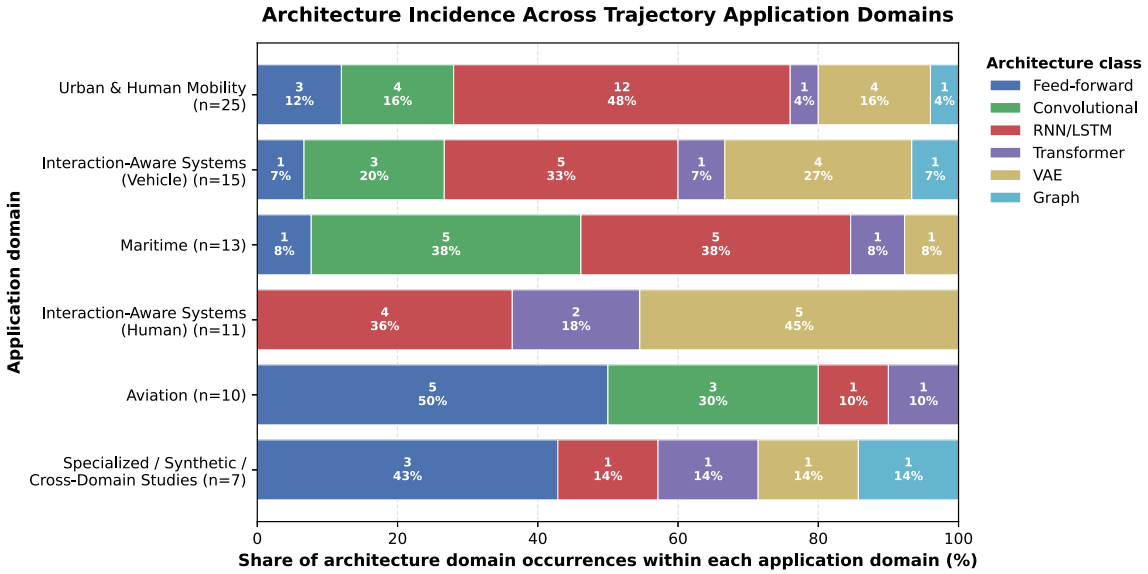

Fig. 9. Relative distribution of autoencoder architecture classes within each trajectory application domain. Each horizontal bar sums to 100% for a given domain, while the labels inside the segments report the corresponding number of architecture–domain occurrences and percentage share. Interaction-aware studies covering multiple agent contexts were counted once in each relevant context; therefore, counts represent architecture–domain occurrences rather than unique paper counts. Low-frequency standalone domains were grouped under "Specialized/Synthetic/Cross-Domain Studies" to avoid overinterpreting categories represented by very few studies.

representations for forecasting future movement. Anomaly/outlier detection shows a more balanced distribution, with feed-forward, RNN/LSTM, VAE, convolutional, transformer-based, and graph-based variants all appearing in the reviewed literature. This diversity reflects the fact that anomalies may be formulated in different ways, including reconstruction deviations, temporal irregularities, spatial-pattern deviations, low-density latent representations, or relational inconsistencies in multi-agent settings.

Clustering is mainly associated with feed-forward and RNN/LSTM-based autoencoders, while convolutional and transformer-based variants also appear. This pattern suggests that clustering-oriented works often rely on latent embeddings that preserve either global trajectory structure, temporal dynamics, or spatially organized movement patterns. Similarity computation, classification, compression, and generation are represented by fewer studies, so their percentages should be interpreted with caution. Nevertheless, the observed distributions are still informative: similarity computation is mainly associated with convolutional and recurrent representations with recent maritime studies further strengthening the role of grid-based and multimodal convolutional autoencoders, classification is dominated by convolutional variants, and generation is mostly associated with recurrent and variational formulations.

Trajectory recovery and reconstruction form another lower-frequency but methodologically diverse task group. Recovery is most often associated with recurrent encoder–decoder formulations, reflecting the need to infer missing or low-sampling-rate trajectory states from sequential context. At the same time, feed-forward, transformer-based, and graph-based variants also appear, indicating that recovery may require different architectural mechanisms depending on whether the missing information is modeled as fragmented coordinate matrices, temporally sparse GPS sequences, road-network-constrained trajectories, or relational spatial-temporal graphs. This distribution suggests that recovery is not simply an extension of compression-oriented reconstruction, but a task-specific reconstruction problem shaped by missingness, sampling sparsity, and spatial feasibility constraints. Overall, the task-level distribution indicates that architecture selection is strongly shaped by the operation expected from the latent space: reconstruction for compression and anomaly detection, distance preservation for clustering and similarity, temporal encoding for prediction, discriminative feature learning for classification, and smooth latent sampling for generation.

Fig. 9 complements the task-level view by showing how architecture classes are distributed across trajectory application domains. This domain-level perspective is important because application domains are associated with different trajectory data properties, such as sampling frequency, spatial constraints, interaction structure, uncertainty, and contextual heterogeneity. Urban and human mobility studies are dominated by RNN/LSTM-based models, followed by convolutional and VAE-based architectures. This reflects the sequential and behavior-oriented nature of urban mobility data, where temporal dependencies, recurrent movement patterns, and latent behavioral structures are central modeling targets. Trajectory recovery studies further reinforce the concentration of trajectory GeoAI research in urban transportation settings, where sparse GPS sampling, road-network constraints, and map-matched trajectory reconstruction naturally motivate recurrent, transformer-based, and graph-enhanced encoder–decoder formulations. Maritime studies also show strong use of RNN/LSTM-based and convolutional variants, consistent with the need to model vessel routes both as ordered AIS sequences and as spatial patterns or trajectory maps.

Aviation studies display a different profile, with feed-forward architectures accounting for the largest share, followed by convolutional models. This suggests that several aviation works formulate trajectory analysis through fixed-length or preprocessed trajectory representations where compact global embeddings and reconstruction-based deviation analysis are sufficient. In contrast, interaction-aware human and vehicle systems show a stronger presence of RNN/LSTM, VAE, transformer-based, and graph-based architectures. This is consistent with the methodological demands of interaction-aware modeling, where future motion depends not only on an individual's trajectory history but also on uncertainty, social or traffic interactions, and the behavior of surrounding agents. The presence of graph-based methods is limited but concentrated in interaction-aware or cross-domain settings, indicating their emerging role in modeling relational structures rather than individual trajectories alone.

The specialized, synthetic, and cross-domain category aggregates low-frequency application contexts to avoid overinterpreting single-study domains. Although this category is smaller than the main

application domains, it highlights the architectural diversity of studies that do not fit cleanly into urban, maritime, aviation, or interaction-aware settings. Taken together, the two distributional views show that architecture usage is not random across the literature. Instead, architecture choices are closely linked to task formulation and data context: recurrent models dominate sequence-heavy problems, convolutional models are favored when trajectories are represented as grids or spatial maps, variational models are common when uncertainty or generation is important, transformer-based models appear in settings requiring long-range dependency modeling, and graph autoencoders are emerging for relational and multi-agent trajectory structures.

6.3. Task-specific characteristics and representation requirements

The reviewed literature shows that autoencoder-based trajectory analysis cannot be understood as a single generic representation-learning problem. Although all tasks rely on some form of latent representation, each task imposes different requirements on what information must be preserved, emphasized, compressed, separated, reconstructed, or generated. Therefore, before comparing architecture families, it is important to clarify the analytical demands of each trajectory task. This subsection discusses the main task-specific characteristics observed across the reviewed studies, focusing on the role of the latent representation, the information required by each task, and the main evaluation and interpretation challenges.

Compression. Trajectory compression aims to reduce storage, transmission, or computational cost while preserving enough information to reconstruct the original movement pattern with acceptable fidelity. The central requirement is therefore a compact latent representation that retains the spatial and temporal structure needed for reconstruction. In this task, performance is not determined by reconstruction quality alone, but by the trade-off between compression ratio and reconstruction error. A highly compressed representation is useful only if it does not remove information that is operationally important, such as trajectory shape, temporal ordering, motion continuity, or key behavioral transitions. This trade-off is especially important in real-time or bandwidth-limited settings, where methods must avoid both excessive data volume and excessive reconstruction delay. Consequently, compression studies typically emphasize metrics such as compression ratio, reconstruction loss, MSE, SNR, or geographic reconstruction error. A recurring challenge is that stronger compression may reduce latent capacity, causing the reconstructed trajectory to lose fine-grained motion details, local deviations, or temporal dependencies. Therefore, trajectory compression should be interpreted as a balance between compactness, fidelity, latency, and downstream usability.

Recovery, reconstruction, and imputation. Trajectory recovery aims to restore missing, fragmented, sparse, or low-sampling-rate trajectories into more complete movement representations. Although closely related to autoencoder reconstruction, the task differs from standard compression because the target is not merely to reproduce a complete input after dimensionality reduction, but to infer unobserved or degraded trajectory states from partial evidence. The central requirement is therefore a representation that preserves temporal continuity, local motion dynamics, and spatial feasibility under incomplete observations. For free-space GPS recovery, this may involve reconstructing missing coordinates while maintaining heading, curvature, and smoothness. For fragmented collective-motion data, the representation must exploit dependencies among agents or trajectory segments to infer missing portions without corrupting observed ones.

In road network settings, recovery becomes more constrained because the model must infer not only where missing points are located, but also whether they are consistent with the road topology, candidate road segments, moving ratios, and plausible route choices. Recent graph and transformer-based encoder–decoder formulations further show that recovery benefits from modeling road network structure, local spatial-temporal context, and shared travel preferences, rather than relying only on sequential interpolation. The main challenge is that visually smooth or numerically accurate reconstructions may still be geographically invalid if they violate network constraints, omit realistic route choices, or ignore sampling uncertainty. Therefore, recovery should be evaluated through both point-level reconstruction errors and trajectory-level validity, including map consistency, route plausibility, temporal continuity, and downstream utility for tasks such as similarity computation, prediction, navigation, or anomaly detection.

Clustering. Trajectory clustering aims to group trajectories into meaningful movement patterns, route structures, behavioral types, or operational flows without relying on predefined labels. The main requirement is not merely compactness, but latent separability: trajectories that are similar according to the intended analytical criterion should be close in the representation space, while different patterns should become distinguishable. This makes clustering highly dependent on how similarity is defined. In some studies, similarity corresponds to geometric route shape; in others, it reflects movement behavior, temporal dynamics, contextual transitions, traffic flows, or operational procedures. As a result, clustering is particularly sensitive to preprocessing, trajectory normalization, sequence-length handling, feature construction, and the inclusion or exclusion of geographic context. A key insight from the reviewed works is that raw spatial proximity is often insufficient, because trajectories may be behaviorally similar even when they occur in different places or at different times, while geographically similar trajectories may represent different behaviors. Clustering evaluation should therefore combine quantitative metrics, such as silhouette score, Davies–Bouldin index, NMI, ARI, or clustering accuracy when labels exist, with visual and domain-based interpretability. The main challenge is that clustering quality depends not only on numerical separation, but also on whether the resulting groups correspond to meaningful mobility patterns, traffic flows, or behavioral categories.

Anomaly and outlier detection. Anomaly detection seeks to identify trajectories, trajectory segments, or movement behaviors that deviate from learned normal patterns. The task is important in safety monitoring, traffic management, maritime surveillance, aviation analysis, driver-risk assessment, and pedestrian safety. Its central representation requirement is sensitivity to abnormal deviations while remaining robust to noise, irregular sampling, and legitimate but unusual behavior. Unlike clustering or classification, anomaly detection often operates under weak supervision or without labeled anomalies, because abnormal events are rare, expensive to annotate, or difficult to define in advance. Therefore, the latent representation must capture the structure of normal behavior well enough that deviations become detectable through reconstruction error, likelihood, latent-space density, or feature-level dissimilarity. However, anomaly detection is especially vulnerable to ambiguity. A high anomaly score does not automatically explain the cause of the abnormality; it may indicate a true risk, a GPS artifact, weather-related deviation, controller intervention, route change, an abrupt but valid maneuver, or data-quality problem. For this reason, anomaly detection should not be treated as a purely binary decision task. It often requires severity ranking, threshold calibration, case-based inspection, and domain-contextual interpretation. Evaluation is also difficult because ground truth may be synthetic, expert-defined, weakly labeled, or indirectly inferred. Thus, effective anomaly analysis requires both quantitative metrics and qualitative validation of detected cases.

Trajectory similarity. Trajectory similarity computation aims to measure how closely two trajectories resemble each other and often serves as a foundation for retrieval, clustering, recommendation, anomaly detection, and route analysis. The key requirement is a distance-preserving or similarity-preserving latent space. Unlike compression, where the goal is to reconstruct the input, similarity computation requires the

representation to preserve relationships between trajectories. This is challenging because trajectory similarity may depend on spatial path shape, temporal alignment, speed profile, activity semantics, road-network structure, or behavioral intent. Traditional trajectory-distance measures can be sensitive to noise, unequal lengths, non-uniform sampling, and temporal misalignment, while also becoming computationally expensive for large datasets. Autoencoder-based similarity methods address this by mapping trajectories into compact representations where vector distances can be computed more efficiently.

However, the quality of the resulting similarity depends strongly on which aspects of the original trajectory are encoded. For example, grid-based convolutional autoencoders emphasize route geometry by transforming trajectories into spatial representations and comparing the resulting latent vectors, which improves efficiency relative to exhaustive point-wise matching. At the same time, more recent multimodal convolutional variants show that geometric similarity alone may be insufficient when trajectories share similar corridors but differ in speed, heading, or navigational intent. A representation that preserves route geometry may therefore fail to preserve timing, semantic context, or fine-grained motion dynamics, while a representation that incorporates motion or contextual attributes may support a more behaviorally meaningful notion of similarity. Another important limitation is evaluation: there is rarely a universal ground-truth benchmark for trajectory similarity, so studies often evaluate similarity indirectly through retrieval quality, clustering performance, or qualitative inspection. As a result, similarity computation should be interpreted according to the intended notion of resemblance rather than as a universal trajectory-matching problem.

Prediction and forecasting. Trajectory prediction aims to estimate future movement from observed trajectory history and, in many cases, from surrounding spatial, social, or environmental context. Its central requirement is a representation that captures temporal continuity, motion dynamics, uncertainty, and possible future variability. Prediction differs from reconstruction because the target is not the observed input but an unknown future trajectory. This makes the task inherently uncertain, especially in pedestrian, vehicle, maritime, and interaction-aware settings where multiple futures may be plausible from the same past observation. For short horizons, local dynamics such as speed, acceleration, heading, and recent motion history may be sufficient. For longer horizons, however, the representation must often encode goals, intentions, route classes, environmental constraints, map structure, neighboring agents, or traffic rules. A recurring insight across the reviewed prediction studies is that motion history alone is often insufficient when future movement depends on interactions or high-level decisions. Evaluation commonly uses displacement-based metrics such as ADE and FDE, along with miss rate, likelihood-based metrics, runtime, and sometimes collision or consistency measures. The main challenge is to avoid producing averaged futures that are numerically plausible but behaviorally unrealistic. Therefore, trajectory prediction should be evaluated not only by positional error, but also by plausibility, diversity, uncertainty calibration, map or context compliance, and operational usefulness.

Classification. Trajectory classification assigns trajectories or trajectory segments to predefined categories such as transportation mode, activity type, maneuver class, or behavioral class. The central requirement is discriminative representation learning. In contrast to clustering, where groups are discovered without labels, classification requires the latent space to separate known categories in a way that supports reliable label assignment. This means that the representation must preserve features that distinguish classes, which may include speed, acceleration, stop duration, turning behavior, recurrence patterns, motion rhythm, semantic context, or temporal dynamics. A major challenge is that different classes can have overlapping movement signatures. For example, transportation modes such as bus and car, or walking and biking, may exhibit similar speed ranges or route characteristics under certain conditions. Classification also depends heavily on segmentation quality: if a trajectory segment contains multiple modes or behaviors, the class label becomes ambiguous. Labeled data availability is another recurring limitation, since trajectory labels are often sparse, expensive, noisy, or unevenly distributed across classes. For this reason, classification studies often require careful feature construction, segment definition, class balancing, and validation under limited-label conditions. Evaluation typically relies on accuracy, precision, recall, F1-score, and confusion analysis, but interpretation should also consider which classes are systematically confused and why.

Generation and synthetic trajectory modeling. Trajectory generation aims to synthesize new trajectories that resemble real movement patterns while supporting applications such as simulation, data augmentation, privacy-preserving data sharing, and rare-event analysis. The central requirement is a latent space that is not only compact, but also smooth, sampleable, diverse, and semantically meaningful. Unlike compression, where the decoder reconstructs known inputs, generation requires sampling new latent codes and decoding them into plausible trajectories. Therefore, generation quality depends on whether the learned latent distribution captures the variability of real movement without producing physically impossible, contextually invalid, or overly repetitive trajectories. The reviewed studies show that generation must balance realism and diversity. Highly realistic samples may lack coverage of rare behaviors, while highly diverse samples may become implausible or violate spatial, temporal, or operational constraints. Evaluation is therefore more complex than standard reconstruction assessment. It may include trajectory-distance measures, distributional similarity, coverage, matching scores, visual inspection, simulation-based validation, or downstream utility tests. For safety-critical settings, such as aviation or autonomous driving, generated trajectories should also be checked for physical feasibility and operational plausibility. Thus, trajectory generation should be understood as both a representation-learning problem and a validation problem: a generated trajectory is useful only if it is realistic, diverse, contextually valid, and suitable for the intended downstream use.

Overall, these task-specific differences show that the value of an autoencoder representation depends on the analytical role assigned to the latent space. Compression requires compactness and reconstruction fidelity; clustering requires separability and interpretable grouping; anomaly detection requires sensitivity to deviations and contextual interpretation; similarity computation requires preservation of pairwise relationships; prediction requires temporal dynamics, uncertainty, and future plausibility; classification requires discriminative structure; and generation requires realistic and diverse sampling. These distinctions provide the basis for the following cross-architecture analysis and task–architecture suitability discussion, where the focus shifts from what each task requires to how different autoencoder families satisfy these requirements.

6.4. Cross-architecture analysis: Formal models, inductive biases, and limitations

This subsection provides a formal and comparative analysis of the main autoencoder families identified in the reviewed trajectory literature. First, a general autoencoder formulation is introduced as a common reference point. Then, each architecture family is examined in terms of its core mathematical mechanism, input assumptions, inductive bias, trajectory-task relevance, and main limitations. This structure enables a direct comparison of how different autoencoder designs encode fixed-vector, spatial, temporal, probabilistic, attention-based, and relational trajectory representations.

6.4.1. General autoencoder formulation for trajectory representation learning

Before comparing the main autoencoder families used in trajectory representation learning, we first define a common reconstruction-based

formulation. Let the reviewed trajectory dataset be denoted as:

$$D = \{x_i\}_{i=1}^{N}$$

where D is a collection of N trajectories or trajectory-derived representations. Depending on the method, each input x_i may correspond to a fixed-length feature vector, an ordered sequence of trajectory states, a rasterized trajectory image, a set of map or agent tokens, or a graph-structured multi-agent representation. In the sequential case, a trajectory can be written as:

$$x_i = (p_{i,1}, p_{i,2}, \dots, p_{i,T_i})$$

where $p_{i,t} \in \mathbb{R}^d$ denotes the trajectory state of trajectory i at time step t. The dimensionality d depends on the available attributes and may include spatial coordinates together with additional motion or contextual variables, such as speed, heading, acceleration, timestamp, semantic context, or agent-level features.

In its general form, an autoencoder consists of an encoder and a decoder. The encoder maps the input trajectory representation x_i into a latent code:

$$z_i = f_\theta(x_i)$$

Here, f_θ denotes the encoder function, θ represents its learnable parameters, and z_i is the latent representation of the input trajectory. The decoder then maps the latent code back to a reconstructed trajectory representation:

$$\hat{x}_i = g_\phi(z_i)$$

Here, g_ϕ denotes the decoder function, ϕ represents its learnable parameters and $\hat{x}_i$ is the reconstructed version of the original input x_i. Therefore, the general autoencoder pipeline can be understood as a mapping from the original trajectory representation to a latent representation and then back to a reconstructed representation: $x_i \to z_i \to \hat{x}_i$.

The latent representation z_i is typically lower-dimensional, more structured, or more task-relevant than the original input. Its role is to retain the information required for the target trajectory-analysis task while discarding redundant or noisy variations. Training is usually based on minimizing a reconstruction objective, optionally combined with a regularization and/or a structural constraint:

$$\min_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \left[L_{rec}(x_i, \hat{x}_i) + \lambda \mathcal{R}(\phi, \theta, z_i) \right] \tag{1}$$

In expression (1) the term $L_{rec}(x_i, \hat{x}_i)$ measures the discrepancy between the original and reconstructed trajectory representation, while $\mathcal{R}(\phi, \theta, z_i)$ denotes a regularization term or structural constraint imposed on the model parameters or latent space. The coefficient λ controls the relative contribution of this regularization term. For trajectory data, the reconstruction loss may take different forms depending on the input representation and task, including but not limited to, mean squared error, mean absolute error, geographic distance, sequence-wise reconstruction loss, image- or map-based reconstruction loss, structural similarity loss, or token-level reconstruction loss.

This common formulation provides the basis for comparing different autoencoder families. The main distinction between architectures is not only the choice of reconstruction loss, but also the way the encoder and decoder are parameterized and the type of structure they assume in the input. Feed-forward autoencoders operate on fixed-length vectors, recurrent autoencoders model temporal dependencies in ordered sequences, convolutional autoencoders exploit local spatial or grid-like structure, variational autoencoders impose probabilistic latent distributions, transformer-based autoencoders rely on attention over trajectory or map tokens, and graph autoencoders model relational structure among agents, locations, or road-network elements. The following analysis therefore uses this common formulation as a reference point for comparing autoencoder families in terms of their mathematical structure, input assumptions, latent-space behavior, inductive biases, and limitations for trajectory representation learning.

6.4.2. Feed-forward autoencoders

Feed-forward autoencoders constitute the most basic autoencoder family and provide the natural baseline for comparing more specialized trajectory models. In this architecture, the encoder and decoder are composed of fully connected layers. Therefore, the input trajectory must be represented as a fixed-length vector before it can be processed. This representation may be obtained by resampling trajectories to a common length, flattening coordinate sequences, extracting handcrafted motion features, constructing fixed-size trajectory windows, or using other preprocessing steps that transform the original movement data into a vector form.

Formally, let the initial input to the encoder be written as:

$$h_i^{(0)} = x_i$$

A feed-forward encoder then applies a sequence of dense nonlinear transformations:

$$h_i^{(\ell)} = \sigma\left(W_e^{(\ell)} h_i^{(\ell-1)} + b_e^{(\ell)}\right), \text{ for } \ell = 1, \dots, L_e$$

The latent representation is obtained at the final encoder layer:

$$z_i = h_i^{(L_e)}$$

The decoder applies a corresponding sequence of dense transformations to map the latent code back to the input space:

$$\hat{x}_i = g_\phi(z_i)$$

The feed-forward autoencoder is commonly trained by minimizing a reconstruction objective of the form:

$$\mathcal{L}_{\text{FF-AE}} = \frac{1}{N} \sum_{i=1}^{N} \left[d(x_i, \hat{x}_i) + \lambda \left(||\theta||_2^2 + ||\phi||_2^2 \right) \right] \tag{2}$$

Here in (2) the $d(x_i, \hat{x}_i)$ term, measures the discrepancy between the original fixed-length trajectory representation and its reconstruction, while the regularization term controls the complexity of the encoder and decoder parameters. In many applications, this distance is implemented as mean squared error or mean absolute error, although task-specific distances may also be used when the input representation has a particular spatial or temporal meaning.

The main inductive bias of feed-forward autoencoders is global vector-based compression. Each layer learns transformations over the entire input vector, without explicitly modeling temporal recurrence, local spatial neighborhoods, probabilistic latent distributions, attention-based token interactions, or graph relations. This makes feed-forward autoencoders simple, flexible, and computationally straightforward, but also strongly dependent on the quality of the fixed-length representation supplied to them. If temporal order, local spatial structure, or inter-agent relations are not already encoded in the input vector, the model has no architectural mechanism for recovering them.

In trajectory analysis, this property makes feed-forward autoencoders useful when the main objective is dimensionality reduction, compact reconstruction, or latent feature extraction from preprocessed trajectory representations. In compression-oriented studies [28], the bottleneck layer provides a compact code that can reduce storage or transmission costs, although reconstruction fidelity depends strongly on the bottleneck size and may require residual coding, filtering, or other post-processing steps. For recovery and reconstruction, feed-forward autoencoders are mainly relevant when incomplete trajectories can be represented as fixed-length coordinate matrices or flattened vectors, allowing missing entries or fragmented segments to be reconstructed through masked reconstruction objectives [46]. Their main caveat is that they do not explicitly model temporal ordering, road-network constraints, or relational context unless these are already encoded in the input representation. In clustering-oriented studies [27], feed-forward

and stacked autoencoders are often used as feature-learning modules before applying GMM, K-means, hierarchical clustering, or clustering-refinement methods. In anomaly detection [34], they support unsupervised scoring by assuming that common trajectories or behavior vectors are reconstructed more accurately than atypical ones.

Compared with more specialized architectures, feed-forward autoencoders offer simplicity and broad applicability, but their representational assumptions are weaker. Recurrent autoencoders are better suited when temporal order and variable-length sequence dynamics must be modeled directly. Convolutional autoencoders are more appropriate when trajectories are represented as rasterized maps, recurrence plots, or other grid-like structures with local spatial patterns. Variational autoencoders provide a probabilistic latent space for uncertainty-aware modeling and generation, while transformer-based and graph-based autoencoders are better equipped for long-range token dependencies and relational multi-agent structures. Thus, feed-forward autoencoders are best understood as a compact fixed-vector baseline: effective when preprocessing already exposes the relevant trajectory information, but less expressive when the task requires explicit modeling of time, space, uncertainty, or interaction.

6.4.3. Convolutional autoencoders

Convolutional autoencoders extend the basic encoder–decoder formulation by replacing dense transformations with convolutional operations. Their central assumption is that the input contains local structure that should be preserved during representation learning. In trajectory analysis, this usually requires transforming trajectories into structured tensor representations, such as rasterized trajectory images, trajectory maps, recurrence plots, gridded spatial representations, or fixed-length multichannel temporal windows. Thus, while feed-forward autoencoders operate on flattened fixed-length vectors, convolutional autoencoders are designed to exploit local spatial, temporal, or image-like regularities in the input representation.

Let the structured input representation of trajectory i be denoted by X_i. This may correspond to a trajectory image, a multichannel grid, a recurrence plot, or a fixed-size temporal tensor. The first hidden representation is initialized as:

$$H_i^{(0)} = X_i$$

A convolutional encoder then applies a sequence of convolutional transformations:

$$H_i^{(\ell)} = \sigma\left(K_e^{(\ell)} * H_i^{(\ell-1)} + b_e^{(\ell)}\right), \text{ for } \ell = 1, \ldots, L_e$$

Here $K_e^{(\ell)}$ denotes the convolutional filters at encoder layer ℓ, the symbol $*$ denotes convolution, $b_e^{(\ell)}$ is a bias term, and σ is a nonlinear activation function. The encoder progressively extracts higher-level local patterns and compresses them into a latent representation:

$$z_i = f_\theta(X_i)$$

The decoder then reconstructs the structured trajectory representation using transposed convolutions, upsampling operations, or convolutional decoding layers:

$$\hat{X}_i = g_\phi(z_i)$$

The corresponding convolutional autoencoder objective can be written as:

$$\mathcal{L}_{\text{CAE}} = \frac{1}{N}\sum_{i=1}^{N}\left[d(X_i, \hat{X}_i) + \lambda \mathcal{R}(\theta, \phi, z_i)\right] \tag{3}$$

In expression (3) the term $d(X_i, \hat{X}_i)$ measures the discrepancy between the original and reconstructed structured trajectory representation. Depending on the representation, this discrepancy may be implemented using mean squared error, mean absolute error, structural similarity, or a weighted combination of pixel-level and structure-preserving losses. For trajectory-map or image-like inputs, a common reconstruction form can be expressed as:

$$d(X_i, \hat{X}_i) = \alpha\|X_i - \hat{X}_i\|_1 + \beta\|X_i - \hat{X}_i\|_2^2 + \gamma\left(1 - \text{SSIM}(X_i, \hat{X}_i)\right)$$

Here, the first two terms measure point-wise reconstruction error, while the structural similarity term encourages the reconstructed map or trajectory image to preserve local and global visual structure. This type of loss is especially relevant when trajectories are encoded as images or spatial maps, because small pixel-wise errors may not fully reflect whether the reconstructed trajectory preserves meaningful route shape, local turns, density patterns, or spatial continuity.

The main inductive bias of convolutional autoencoders is locality. Convolutional filters learn patterns from neighboring positions in the input, making them suitable for trajectory representations where nearby pixels, cells, time steps, or feature channels carry related information. This differs from feed-forward autoencoders, where all input dimensions are connected globally and the model does not explicitly exploit local structure. As a result, convolutional autoencoders can be more parameter-efficient and better aligned with trajectory images, gridded spatial fields, recurrence plots, and multichannel motion tensors. However, this advantage depends on whether the trajectory-to-tensor transformation preserves meaningful movement information. If important temporal order, speed variation, or semantic context is lost during rasterization or gridding, the convolutional model can only learn from the structure that remains in the constructed representation.

Across the reviewed studies, convolutional autoencoders are particularly useful in tasks where trajectory geometry or local spatio-temporal structure can be encoded visually or tensorially. In similarity computation [48], convolutional encoders learn compact features from rasterized trajectory images, allowing trajectory comparison through distances in latent space rather than expensive point-wise alignment. In anomaly detection [53], convolutional variants reconstruct multivariate time-series windows or trajectory maps, so abnormal behavior can be detected through reconstruction error or structural dissimilarity. In clustering [49], convolutional embedded clustering methods use reconstruction together with clustering-oriented objectives to create more separable latent spaces. In classification [21], supervised or semi-supervised convolutional autoencoders learn discriminative representations from recurrence plots, motion-feature tensors, or structured trajectory segments. In generation and compression [50], convolutional or temporal-convolutional variants can preserve local trajectory patterns while producing compact or sampleable latent representations, although the strongest generative evidence often involves additional variational mechanisms.

Compared with other autoencoder families, convolutional autoencoders occupy an intermediate position. They are more structured than feed-forward autoencoders because they encode local spatial or temporal neighborhoods, but they are less naturally suited than recurrent models for variable-length sequential dependencies unless the sequence is first converted into a fixed-size tensor. They can preserve spatial patterns more effectively than dense models, but they do not inherently model probabilistic uncertainty like variational autoencoders, global token interactions like transformers, or explicit inter-agent relations like graph autoencoders. Their main strength is therefore representation-dependent: they are powerful when the trajectory problem can be reformulated as reconstruction or feature learning over a meaningful grid, image, map, or multichannel tensor. Their main limitation is the same dependency: poor rasterization, inappropriate resolution, excessive sparsity, or loss of temporal semantics can directly limit the quality of the learned latent representation.

6.4.4. Recurrent autoencoders: RNN, LSTM, and GRU variants

Recurrent autoencoders extend the autoencoder framework to ordered trajectory sequences. Unlike feed-forward autoencoders, which

require trajectories to be represented as fixed-length vectors, and convolutional autoencoders, which rely on structured grid-like or image-like representations, recurrent autoencoders are designed to process sequential data directly. Their main inductive bias is temporal dependency modeling: the representation of a trajectory state depends not only on the current input but also on the accumulated hidden state from previous time steps. This makes recurrent autoencoders particularly relevant for trajectory data, where movement evolves over time and where speed, direction, acceleration, route progression, and behavioral transitions are often meaningful only when interpreted sequentially.

For a trajectory written as a sequence of states:

$$x_i = (p_{i,1}, p_{i,2}, \dots, p_{i,T_i})$$

a basic recurrent encoder updates a hidden state at each time step. In its simplest form, this can be written as:

$$h_{i,t} = f_\theta(p_{i,t}, h_{i,t-1})$$

or, more explicitly, as:

$$h_{i,t} = \tanh(W_p p_{i,t} + U_h h_{i,t-1} + b_h)$$

Here, $p_{i,t}$ is the trajectory state at time step t, $h_{i,t}$ is the hidden representation after observing the sequence up to time t, and f_θ is a recurrent transition function with learnable parameters θ. In the explicit form, W_p transforms the current trajectory state, U_h transforms the previous hidden state $h_{i,t-1}$ (which carries the accumulated historical context), b_h is a consolidated bias term that absorbs the offsets for both linear transformations, and tanh introduces nonlinearity. The key idea is that the current hidden state recurrently summarizes both the current observation and the previous temporal context. The final hidden state, or a transformation of all hidden states, is then used as the latent representation of the trajectory:

$$z_i = h_{i,T_i}$$

In a sequence-to-sequence recurrent autoencoder, the decoder initializes its own hidden state from the latent code and reconstructs the trajectory step by step. This process is defined by the following recurrent system:

$$\hat{h}_{i,t} = f_\phi(\hat{p}_{i,t-1}, \hat{h}_{i,t-1})$$

$$\hat{p}_{i,t} = o_\phi(\hat{h}_{i,t})$$

where f_ϕ is the decoder recurrence, o_ϕ maps the decoder hidden state to a reconstructed trajectory state, and $\hat{p}_{i,t}$ is the reconstructed state at time step t. To anchor this recurrence, the decoder's memory is initialized at time step $t = 0$ directly using the latent representation:

$$\hat{h}_{i,0} = z_i$$

The completely reconstructed trajectory sequence is therefore:

$$\hat{x}_i = (\hat{p}_{i,1}, \hat{p}_{i,2}, \dots, \hat{p}_{i,T_i})$$

This vanilla recurrent formulation is conceptually simple: each state is processed in order, and the hidden state is repeatedly updated. However, a simple RNN stores temporal information only through $h_{i,t}$, which can make longer dependencies difficult to preserve during training. LSTM autoencoders address this limitation by introducing a separate cell state, denoted c_t, which acts as a longer-term memory, and by using gates to regulate how information enters, leaves, or remains in that memory. At each time step t, the LSTM receives the current trajectory state $p_{i,t}$, the previous hidden state $h_{i,t-1}$, and the previous cell state $c_{i,t-1}$ for a specific trajectory i. To avoid notation conflicts with the trajectory index i, the input gate is denoted as a_t. The forget gate decides how much of the previous cell state should be retained:

$$f_{i,t} = \sigma(W_f p_{i,t} + U_f h_{i,t-1} + b_f)$$

The input gate decides how much new information should be written into memory:

$$a_{i,t} = \sigma(W_a p_{i,t} + U_a h_{i,t-1} + b_a)$$

The candidate cell state creates the new candidate memory content:

$$\tilde{c}_{i,t} = \tanh(W_c p_{i,t} + U_c h_{i,t-1} + b_c)$$

The cell state is then updated by combining retained past memory with selected new information:

$$c_{i,t} = f_{i,t} \odot c_{i,t-1} + a_{i,t} \odot \tilde{c}_{i,t} \tag{4}$$

Finally, the output gate decides which part of the updated cell state should be exposed as the hidden state:

$$o_{i,t} = \sigma(W_o p_{i,t} + U_o h_{i,t-1} + b_o)$$

$$h_{i,t} = o_{i,t} \odot \tanh(c_{i,t})$$

In these equations, σ produces values between 0 and 1, so the gates act as soft filters. Values close to 0 suppress information, while values close to 1 preserve or pass information. The operator $\odot$ denotes element-wise multiplication. The forget gate $f_{i,t}$ controls how much previous memory is retained, the input/activation (a) gate $a_{i,t}$ controls how much new candidate information is stored, and the output gate $o_{i,t}$ controls how much of the updated memory becomes visible as the hidden state. The most important distinction from a vanilla RNN is therefore the additive cell-state update defined in Eq. (4). This update provides a more stable memory path across time, helping the model preserve longer-range trajectory information such as route progression, maneuver evolution, stop–move patterns, or delayed effects of earlier movement states.

GRU autoencoders provide a related but simpler gated recurrent alternative. A GRU does not maintain a separate cell state $c_{i,t}$. Instead, it uses gates to directly update the hidden state for trajectory i. A compact GRU formulation is:

$$r_{i,t} = \sigma(W_r p_{i,t} + U_r h_{i,t-1} + b_r) \tag{5}$$

$$u_{i,t} = \sigma(W_u p_{i,t} + U_u h_{i,t-1} + b_u) \tag{6}$$

$$\tilde{h}_{i,t} = \tanh\left(W_h p_{i,t} + U_h(r_{i,t} \odot h_{i,t-1}) + b_h\right) \tag{7}$$

$$h_{i,t} = (1 - u_{i,t}) \odot h_{i,t-1} + u_{i,t} \odot \tilde{h}_{i,t} \tag{8}$$

The operations governing the GRU cell are defined in Eqs. (5)–(8). Here, the reset gate in Eq. (5) controls how much of the previous hidden state is used when computing the candidate hidden state in Eq. (7). Meanwhile, the update gate in Eq. (6) controls the interpolation (the blending) between the previous hidden state and this new candidate state to produce the final, updated hidden representation $h_{i,t}$ in Eq. (8). Compared with LSTMs, GRUs are usually more compact because they use fewer gates and do not maintain a separate cell state. This can make them attractive in trajectory applications where computational efficiency matters, while still retaining a stronger temporal modeling mechanism than a vanilla RNN. The reconstruction objective for a recurrent autoencoder is usually sequence-wise:

$$\mathcal{L}_{\text{RNN-AE}} = \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{t=1}^{T_i} d(p_{i,t}, \hat{p}_{i,t}) + \lambda \mathcal{R}(\theta, \phi, z_i) \right] \tag{9}$$

In Eq. (9) the reconstruction error is accumulated across time steps and combined with a trajectory-specific latent regularization term. The distance metric $d(p_{i,t}, \hat{p}_{i,t})$ may correspond to mean squared error, mean

absolute error, geographic distance, Haversine distance, or another task-specific trajectory-state discrepancy. This formulation makes recurrent autoencoders suitable for tasks where preserving the temporal evolution of the trajectory is more important than reconstructing a static vector or image-like representation.

It is important to distinguish standard recurrent autoencoders from recurrent probabilistic autoencoders. A standard RNN, LSTM, or GRU autoencoder learns deterministic hidden states and latent representations. It does not, by itself, learn an explicit probability distribution over latent variables. Distributional modeling appears when recurrent encoders and decoders are combined with variational mechanisms, such as recurrent VAE or CVAE formulations. In those cases, the encoder estimates the parameters of a latent distribution, such as a mean and variance, and samples latent variables from that distribution. Such recurrent-variational models are especially relevant for uncertainty-aware prediction, multimodal forecasting, and trajectory generation, but their probabilistic formulation belongs to the VAE family rather than to plain recurrent autoencoders.

Across the reviewed literature, recurrent autoencoders are especially prominent because many trajectory tasks are inherently sequential. In compression [67], LSTM and RNN variants encode complete trajectories into compact latent representations while preserving temporal dependencies, or they reduce communication by predicting future samples and transmitting updates only when prediction error becomes too large. For trajectory recovery, recurrent encoder–decoder variants are suitable when missing or low-sampling-rate points must be inferred from surrounding sequential context [18,92]. Their strength lies in modeling temporal continuity, but their performance depends on how missingness, sampling intervals, candidate road segments, and map-constrained outputs are handled during decoding. In clustering [89], recurrent sequence-to-sequence embeddings transform variable-length movement histories into fixed-length latent vectors that can represent movement behavior rather than only raw spatial proximity. In anomaly detection [74], recurrent autoencoders model normal sequential dynamics and identify deviations through reconstruction error, learned trajectory-distance behavior, or generative likelihood when combined with probabilistic extensions. In similarity computation [13], recurrent representation learning replaces expensive point-matching distances with latent-vector comparisons while improving robustness to low sampling rates, noise, and irregular temporal observations. In prediction [69], LSTM and GRU architectures form a dominant backbone because forecasting requires modeling temporal dependencies from observed motion histories. In generation [85], recurrent encoder-decoder and recurrent generative variants support the synthesis of sequential trajectories that preserve temporal movement structure.

Compared with feed-forward and convolutional autoencoders, recurrent autoencoders provide a stronger mechanism for temporal modeling. Feed-forward autoencoders can only model temporal information if it is already encoded into a fixed vector, while convolutional autoencoders capture local patterns in structured tensors but often require fixed-size rasterized or windowed representations. Recurrent autoencoders, by contrast, process trajectories as ordered sequences and update the latent representation as new states are observed. This makes them more suitable for variable-length trajectories, time-dependent movement behavior, and sequential forecasting. However, recurrent models also have important limitations. Their performance depends heavily on preprocessing, feature construction, sequence alignment, and the definition of the trajectory state. Plain recurrent reconstruction may not produce clustering-friendly, anomaly-sensitive, or prediction-ready latent spaces without additional objectives such as clustering loss, contrastive loss, attention, route-class modeling, or uncertainty-aware latent variables. Moreover, deterministic recurrent predictors may average over multiple plausible futures, and standard LSTM-based models may struggle with very long-range dependencies, complex multi-agent interactions, or high-dimensional contextual structures.

Therefore, recurrent autoencoders are best understood as sequence-oriented autoencoder models. Their main advantage over feed-forward and convolutional variants is their ability to encode temporal order and motion dynamics directly. Their main weakness is that temporal recurrence alone is often insufficient for complex trajectory tasks: interaction modeling, long-range dependency handling, uncertainty estimation, and task-specific latent-space shaping frequently require additional mechanisms. This explains why many of the strongest recurrent trajectory models in the reviewed literature are hybrid systems that combine LSTM or GRU backbones with attention, clustering objectives, contrastive learning, variational latent variables, route classes, or domain-specific preprocessing.

6.4.5. Variational and conditional variational autoencoders

Variational autoencoders differ from deterministic autoencoders by replacing a single latent code with a latent probability distribution. In a standard feed-forward, convolutional, or recurrent autoencoder, the encoder maps an input trajectory representation directly to a deterministic latent vector. In a VAE, the encoder instead estimates the parameters of a probability distribution over latent variables. This makes the latent space smoother, more regularized, and sampleable, which is particularly useful for trajectory generation, uncertainty-aware prediction, multimodal forecasting, and anomaly detection.

For an input trajectory representation x_i, the VAE encoder defines an approximate posterior distribution over the latent variable:

$$q_\phi(z_i \mid x_i) = \mathcal{N}\left(\mu_\phi(x_i), \mathrm{diag}(\sigma^2_\phi(x_i))\right)$$

Here, the encoder does not output only one latent vector. Instead, it outputs a mean vector $\mu_\phi(x_i)$ and a variance vector $\sigma^2_\phi(x_i)$. These two vectors define a Gaussian distribution from which the latent code can be sampled. The latent variable is obtained through the reparameterization trick:

$$z_i = \mu_\phi(x_i) + \sigma_\phi(x_i) \odot \varepsilon, \quad \text{where } \varepsilon \sim \mathcal{N}(0, I)$$

This formulation separates the random sampling operation from the learnable encoder parameters, allowing the model to be trained with gradient-based optimization. The decoder then reconstructs the input by defining a likelihood distribution over the reconstructed trajectory representation: $p_\theta(x_i \mid z_i)$. In practice, the decoder maps the sampled latent variable z_i to a reconstructed output $\hat{x}_i$, but the probabilistic interpretation is that the decoder models how likely the observed trajectory representation is under the sampled latent code. The VAE is trained by maximizing the evidence lower bound, or equivalently by minimizing the negative evidence lower bound. A common loss form is:

$$\mathcal{L}_{\mathrm{VAE}} = \frac{1}{N}\sum_{i=1}^{N}\left[-\mathbb{E}_{q_\phi(z_i|x_i)}\left[\log p_\theta(x_i \mid z_i)\right] + \beta D_{\mathrm{KL}}\left(q_\phi(z_i \mid x_i) \parallel p(z)\right)\right] \tag{10}$$

The first term in Eq. (10) is the reconstruction term. It encourages the decoder to reconstruct the original trajectory representation from the sampled latent variable. The second term is the Kullback–Leibler divergence between the approximate posterior $q_\phi(z_i \mid x_i)$ and a prior distribution $p(z)$, usually a standard normal distribution. This KL term regularizes the latent space by encouraging the learned posterior distributions to remain close to the prior. The coefficient β controls the strength of this regularization. When $\beta = 1$, the formulation corresponds to the standard VAE objective; when β is adjusted, the model can trade reconstruction fidelity against latent-space regularity.

If the decoder likelihood is assumed to be Gaussian with an identity covariance matrix, the reconstruction term is implemented as a squared reconstruction error:

$$-\log p_\theta(x_i \mid z_i) \propto \|x_i - \hat{x}_i\|^2$$

Thus, the VAE objective can be interpreted as a reconstruction–regularization trade-off. The reconstruction term forces the model to preserve information needed to rebuild the trajectory representation, while the KL term forces the latent space to follow a smooth and sampleable distribution. This is the main mathematical distinction between VAEs and deterministic autoencoders. A deterministic autoencoder learns a point representation $z_i = f_\phi(x_i)$, whereas a VAE learns a distribution $q_\phi(z_i \mid x_i)$ and samples from it.

Conditional variational autoencoders extend this formulation by conditioning both the encoder and decoder on additional information c_i. In trajectory modeling, c_i may represent observed trajectory history, agent type, map context, road-network information, maneuver class, destination, neighboring agents, or environmental context. The conditional encoder and decoder can be written as:

$$q_\phi(z_i \mid x_i, c_i)$$

and

$$p_\theta(x_i \mid z_i, c_i)$$

The corresponding conditional VAE objective is:

$$\mathcal{L}_{\text{CVAE}} = \frac{1}{N}\sum_{i=1}^{N}\Big[-\mathbb{E}_{q_\phi(z_i|x_i,c_i)}\left[\log p_\theta(x_i \mid z_i, c_i)\right] + \beta D_{\text{KL}}\left(q_\phi(z_i \mid x_i, c_i) \,\|\, p(z \mid c_i)\right)\Big] \tag{11}$$

This conditional form shown in (11) is especially important for trajectory prediction, where the future trajectory is not generated from an unconditional latent space, but from a latent distribution conditioned on the observed past and relevant contextual information. In this setting, different samples from the latent distribution can correspond to different plausible futures under the same observed history.

Some trajectory studies further extend the VAE prior beyond a single standard Gaussian. For example, mixture-based formulations can define the prior as:

$$p(z_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$$

Here, the latent prior is a mixture of K Gaussian components, where π_k is the weight of component k, and μ_k and Σ_k are its mean and covariance. This type of prior is useful when the trajectory distribution is naturally multimodal, such as when vessel routes, pedestrian behaviors, or vehicle maneuvers form several distinct latent groups. Compared to a single Gaussian prior, a mixture prior can represent multiple modes of normal behavior more explicitly.

The main inductive bias of VAE-based architectures is probabilistic latent regularization. Unlike feed-forward autoencoders, which learn deterministic fixed-vector embeddings, or recurrent autoencoders, which learn deterministic sequence representations, VAEs organize the latent space as a distribution. This allows the model to sample new latent codes, quantify uncertainty, and represent multiple plausible outputs. This property is especially valuable in trajectory domains where future motion is uncertain, observed data are incomplete, or realistic synthetic trajectories are needed.

Across the reviewed literature, VAE-based models appear in several roles. In clustering [100], variational formulations can jointly learn latent representations and cluster assignments, while separating cluster-level structure from individual trajectory variation. This is useful when the goal is not only to reconstruct trajectories but also to organize mobility behaviors into robust latent groups. In anomaly detection [101], VAEs learn distributions of normal movement and detect deviations through reconstruction discrepancy, likelihood-related scores, or distance between original and generated trajectories. Mixture-based, context-aware, and graph-enhanced VAE variants further address multi-class trajectory distributions, contextual interpretability, and online anomaly detection. In prediction [106], CVAE and timewise VAE variants are particularly important because future trajectories are inherently multimodal. Latent sampling allows the model to generate multiple plausible futures rather than collapsing them into a single averaged prediction. In generation [22], VAEs provide a natural mechanism for sampling synthetic trajectories from a learned latent distribution, making them suitable for data augmentation, simulation, and privacy-sensitive mobility modeling.

Compared with deterministic autoencoder families, the main strength of VAEs is that they provide a structured and sampleable latent space. Feed-forward, convolutional, and recurrent autoencoders can produce useful embeddings, but their latent spaces are not necessarily continuous, smooth, or suitable for sampling. A VAE explicitly encourages this structure through KL regularization. This makes VAE-based architectures more appropriate when the task requires uncertainty modeling, multimodal output, controlled generation, or distribution-aware anomaly scoring. However, this advantage comes with several limitations. The KL regularization can reduce reconstruction sharpness or fidelity if it is too strong, while weak regularization can produce a latent space that is less useful for sampling. A simple Gaussian prior may also be insufficient for complex or multimodal trajectory distributions, motivating Gaussian-mixture, conditional, timewise, or context-aware extensions. In addition, VAE performance depends heavily on how the trajectory input and conditioning variables are represented. Without appropriate social, spatial, temporal, map, or semantic context, a VAE may learn a smooth latent space but still fail to capture the true sources of trajectory variation.

For these reasons, VAE-based autoencoders are best understood as probabilistic representation-learning models. Their key contribution is not only reconstruction, but the construction of a latent distribution that can support sampling, uncertainty estimation, and multimodal reasoning. This makes them more expressive than deterministic autoencoders for generation and prediction, and more distribution-aware for anomaly detection and clustering. At the same time, their effectiveness depends on the suitability of the latent prior, the balance between reconstruction and regularization, and the availability of meaningful conditioning information for the trajectory task.

6.4.6. Transformer-based autoencoders

Transformer-based autoencoders extend the autoencoder framework by replacing recurrence and local convolution with self-attention. Their main inductive bias is global dependency modeling: each token in the input can directly attend to every other token. This is particularly relevant for trajectory representation learning when the meaning of a movement state depends not only on its local temporal neighborhood, but also on distant past states, neighboring agents, map elements, route context, maneuver modes, or long-range spatio-temporal structure. Compared with recurrent autoencoders, Transformer-based models do not process trajectory states strictly one step at a time. Compared with convolutional autoencoders, they are not restricted to local receptive fields. Instead, they learn pairwise relevance among the tokens that define the trajectory or trajectory scene.

For a trajectory or trajectory-scene representation, let the input be written as a sequence of tokens:

$$S_i = (s_{i,1}, s_{i,2}, \ldots, s_{i,M_i})$$

Here, $s_{i,m}$ denotes token m of sample i, and M_i is the number of tokens. Depending on the trajectory task, a token may represent a trajectory state, a short trajectory segment, an agent, a map element, a lane polyline, a spatial grid cell, or a learned patch from a trajectory-map representation. Since self-attention alone does not inherently encode order or spatial position, each token is first embedded and combined with positional, temporal, semantic, or map-related information:

$$e_{i,m} = \text{Emb}(s_{i,m}) + r_{i,m}$$

Here, $\mathrm{Emb}(s_{i,m})$ is the learned embedding of the token, while $r_{i,m}$ represents additional information such as position in the sequence, timestamp, agent identity, map location, or semantic type. The embedded input sequence is therefore:

$$E_i = (e_{i,1}, e_{i,2}, \dots, e_{i,M_i})$$

The core operation of the Transformer encoder is self-attention. For the embedded token matrix E_i, the model computes query, key, and value representations:

$$Q_i = E_i W_Q$$

$$K_i = E_i W_K$$

$$V_i = E_i W_V$$

The query matrix Q_i represents what each token is looking for, the key matrix K_i represents what each token offers for comparison, and the value matrix V_i contains the information that will be passed forward after attention weights are computed. The attention weights are obtained through scaled dot-product attention:

$$A_i = \mathrm{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right)$$

Here, A_i is the attention matrix and d_k is the dimensionality of the key/query vectors. The scaling term $\sqrt{d_k}$ prevents the dot products from becoming too large, which stabilizes the softmax operation. Each entry of A_i expresses how strongly one token attends to another token. The self-attention output is then:

$$\mathrm{SA}(E_i) = A_i V_i$$

This means that each token representation is updated as a weighted combination of the value vectors of all tokens. In trajectory terms, a token corresponding to a current movement state can attend to earlier trajectory states, neighboring agents, map elements, or other contextual tokens if they are relevant for reconstructing or representing the input.

In practice, Transformers use multi-head attention rather than a single attention operation. Multiple attention heads are learned in parallel:

$$\mathrm{head}_j = \mathrm{Attention}(E_i W_{Q,j}, E_i W_{K,j}, E_i W_{V,j}), \quad \text{for } j = 1, \dots, H$$

$$\mathrm{MultiHead}(E_i) = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \dots, \mathrm{head}_H) W_O$$

Each head can learn a different type of relationship. In trajectory analysis, one head may emphasize temporal continuity, another may capture agent interaction, another may focus on map constraints, and another may capture long-range route dependencies. The outputs of all heads are concatenated and projected through W_O. A standard Transformer encoder block combines multi-head attention with residual connections, normalization, and a position-wise feed-forward network:

$$E_i' = \mathrm{LayerNorm}(E_i + \mathrm{MultiHead}(E_i))$$

$$H_i = \mathrm{LayerNorm}(E_i' + \mathrm{FFN}(E_i'))$$

The residual connections help preserve the original token information, while layer normalization stabilizes training. The feed-forward network applies additional nonlinear transformation to each token representation after the attention step. After several such blocks, the Transformer encoder produces contextualized latent token representations:

$$Z_i = T_\theta(E_i)$$

Here, T_θ denotes the Transformer encoder with learnable parameters θ, and Z_i contains the latent representation of all input tokens after global attention has been applied. Depending on the specific autoencoder design, the latent representation may remain token-wise, or it may be pooled into a single global representation:

$$z_i = \mathrm{Pool}(Z_i)$$

The token-wise representation Z_i is useful when the decoder reconstructs the full input sequence token by token, while the pooled representation z_i is useful when a compact global trajectory embedding is needed for clustering, retrieval, anomaly scoring, or downstream prediction.

The Transformer decoder reconstructs the original token sequence from the latent representation. In a general autoencoder form, this can be written as:

$$\hat{Z}_i = T_{\phi,\mathrm{dec}}(Z_i)$$

$$\hat{s}_{i,m} = O_\phi(\hat{Z}_{i,m}), \text{ for } m = 1, \dots, M_i$$

Here, $T_{\phi,\mathrm{dec}}$ denotes the Transformer decoder, O_ϕ maps each decoded token representation back to the reconstruction space, and $\hat{s}_{i,m}$ is the reconstructed version of the original token $s_{i,m}$. The reconstructed sample is therefore:

$$\hat{S}_i = (\hat{s}_{i,1}, \hat{s}_{i,2}, \dots, \hat{s}_{i,M_i})$$

The reconstruction objective for a Transformer autoencoder can be written as:

$$\mathcal{L}_{\mathrm{TAE}} = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{m=1}^{M_i} d(s_{i,m}, \hat{s}_{i,m}) + \lambda \mathcal{R}(\theta, \phi, Z_i)\right] \tag{12}$$

In Eq. (12) the reconstruction loss is accumulated over tokens. The distance $d(s_{i,m}, \hat{s}_{i,m})$ depends on the type of token being reconstructed. For coordinate or trajectory-state tokens, it may be a displacement error, mean squared error, mean absolute error, or geographic distance. For map or grid tokens, it may be a reconstruction loss over spatial attributes. For categorical or semantic tokens, it may be a cross-entropy loss. The regularization term $\mathcal{R}(\theta, \phi, Z_i)$ may include weight decay, sparsity constraints, clustering-oriented constraints, or other task-specific penalties.

This formulation shows the main mathematical distinction of Transformer autoencoders. Feed-forward autoencoders treat the input as a fixed vector, convolutional autoencoders exploit local neighborhoods, and recurrent autoencoders propagate information through sequential hidden states. Transformer autoencoders instead reconstruct the input through globally contextualized token representations. Their latent representation is not produced by scanning the trajectory step by step, but by allowing all trajectory, agent, and context tokens to interact through attention.

Across the reviewed literature, Transformer and attention-based autoencoder variants are most relevant when trajectory representations depend on long-range or heterogeneous context. In clustering [93], attention-based representations can emphasize salient trajectory features and reduce the influence of noisy or redundant components before clustering. Transformer-based latent representations are particularly useful when clusters depend on nonlinear spatio-temporal dependencies rather than only local geometry or raw coordinate similarity. In anomaly detection [94], Transformer-based autoencoder variants can reconstruct normal multivariate trajectory patterns and identify deviations when abnormality depends on long-range temporal context, speed, heading, route progression, or interactions among several trajectory attributes. In prediction [95], masked Transformer autoencoder variants use the same self-attention mechanism but train the encoder through reconstruction of hidden trajectory or map tokens, making them useful for self-supervised pretraining before downstream forecasting. In recovery [99], the same global attention mechanism can relate sparse observed points, road-network context, and missing trajectory states, making

Transformer-based encoder–decoder variants useful when reconstruction depends on long-range temporal dependencies or map-conditioned reasoning. However, this benefit depends strongly on tokenization, positional encoding, candidate masking, and the computational cost of attention over long trajectories or dense map representations.

Compared with recurrent autoencoders, Transformer-based autoencoders are stronger at direct global dependency modeling because any token can attend to any other token without passing information through a chain of hidden states. This can be advantageous for long trajectories, multi-agent scenes, and map-conditioned trajectory tasks. Compared with convolutional autoencoders, Transformers are less constrained by local spatial neighborhoods and can connect distant trajectory states or context elements directly. Compared with variational autoencoders, however, standard Transformer autoencoders are not inherently probabilistic; they require a variational extension if latent sampling, uncertainty estimation, or explicit distributional modeling is needed. Compared with graph autoencoders, Transformers can model interactions through learned attention weights, but they do not impose an explicit relational topology unless agent, road-network, or map structure is encoded into the token design.

The main limitation of Transformer-based trajectory autoencoders is their dependence on representation design. A Transformer does not automatically know which trajectory points, agents, map elements, or temporal positions are meaningful. This information must be exposed through tokenization, positional encoding, context encoding, attention design, and task-specific objectives. In addition, full self-attention has quadratic complexity with respect to the number of tokens, which can become expensive for long trajectories, dense multi-agent scenes, or high-resolution map inputs. The reviewed evidence also suggests that many strong Transformer-based trajectory models are not generic Transformer autoencoders, but hybrid or task-specific systems. Masked autoencoder variants add a self-supervised reconstruction strategy, Transformer–VAE variants add probabilistic latent modeling, and some Transformer prediction models use encoder–decoder attention without being strictly autoencoder-based.

Therefore, Transformer-based autoencoders are best understood as global-context representation learners. Their main advantage over feed-forward, convolutional, and recurrent autoencoders is their ability to model long-range dependencies and heterogeneous token interactions within a unified architecture. Their main weakness is that this flexibility comes with substantial design dependence: token construction, positional encoding, context representation, reconstruction target, and computational cost strongly influence the learned latent representation. In the context of trajectory GeoAI, they are most appropriate when the task requires reasoning over long temporal horizons, multiple agents, map structure, or complex contextual relationships, while simpler architectures may be sufficient when the task is dominated by fixed-vector compression, local spatial reconstruction, or short-range sequential dynamics.

6.4.7. Graph autoencoders

Graph autoencoders extend the autoencoder framework to relational trajectory representations. Instead of treating a trajectory as a fixed vector, image, sequence, or token set, graph-based models represent entities and their interactions as nodes and edges. In trajectory analysis, nodes may correspond to agents, trajectory points, road-network elements, or spatial locations, while edges may represent physical proximity, interaction strength, road connectivity, influence between agents, or inferred dynamic relations. This makes graph autoencoders particularly relevant when the target behavior cannot be understood from isolated trajectories alone, but depends on interaction structure.

For a trajectory scene or relational trajectory sample, let the graph be written as:

$$G_i = (V_i, \mathcal{E}_i)$$

where V_i is the set of nodes and $\mathcal{E}_i$ is the set of edges. The graph can be represented by a node-feature matrix and an adjacency matrix:

$$X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n_i}]$$
$$A_i \in \{0,1\}^{n_i \times n_i}$$

Here, $x_{i,v}$ is the feature vector of node v, n_i is the number of nodes in graph G_i, and A_i describes which nodes are connected. In trajectory applications, node features may include position, velocity, acceleration, heading, or other agent-level attributes, while the adjacency matrix may encode observed, predefined, or learned relationships between agents or spatial elements. A graph encoder updates node representations by aggregating information from neighboring nodes. A common message-passing form is:

$$h_{i,v}^{(l+1)} = \sigma\left(W^{(l)} h_{i,v}^{(l)} + \sum_{u \in \mathcal{N}(v)} \alpha_{i,vu}^{(l)} W_N^{(l)} h_{i,u}^{(l)} \right)$$

Here, $h_{i,v}^{(l)}$ is the representation of node v at layer l, $\mathcal{N}(v)$ is the set of neighboring nodes connected to v, and $\alpha_{i,vu}^{(l)}$ controls the influence of neighboring node u on node v. The term $W^{(l)} h_{i,v}^{(l)}$ updates the node using its own features, while the summation term aggregates information from its neighbors. In simple terms, each node representation is updated using both its own movement state and the states of related agents or spatial elements.

After several graph-encoding layers, the model obtains latent node representations:

$$Z_i = f_\theta(X_i, A_i)$$

The latent representation Z_i may be used directly as a set of node embeddings, or it may be pooled into a graph-level representation:

$$z_i = \mathrm{Pool}(Z_i)$$

A graph decoder then reconstructs node features, edges, or both. Node-feature reconstruction can be written as:

$$\hat{X}_i = g_\phi(Z_i)$$

while edge or adjacency reconstruction can be written as:

$$\hat{A}_i = \mathrm{sigmoid}(Z_i Z_i^T)$$

The adjacency reconstruction equation means that two nodes are predicted to be connected when their latent representations are similar or compatible. A general graph autoencoder objective can therefore be written as:

$$\mathcal{L}_{\mathrm{GAE}} = \frac{1}{N} \sum_{i=1}^{N} \left[d_X(X_i, \hat{X}_i) + \gamma d_A(A_i, \hat{A}_i) + \lambda \mathcal{R}(\theta, \phi, Z_i) \right] \tag{13}$$

In objective (13), $d_X(X_i, \hat{X}_i)$ measures node-feature reconstruction error, $d_A(A_i, \hat{A}_i)$ measures edge or adjacency reconstruction error, γ controls the relative importance of structural reconstruction, and $\mathcal{R}(\theta, \phi, Z_i)$ denotes regularization or a task-specific constraint.

The main inductive bias of graph autoencoders is relational message passing. Feed-forward autoencoders assume fixed vector inputs, convolutional autoencoders assume local grid-like structure, recurrent autoencoders assume temporal sequence structure, and Transformer autoencoders learn global token interactions through attention. Graph autoencoders instead assume that the relevant structure is explicitly relational: what matters is not only the state of each trajectory or agent, but also how nodes influence one another through edges. This makes them especially suitable for multi-agent trajectory problems, where abnormality or future motion may depend on interaction patterns rather than isolated movement histories.

In the reviewed literature, graph autoencoder evidence appears in two main contexts. For anomaly detection [113], spatio-temporal graph autoencoders model interacting agents as graph nodes and use relational structure to detect abnormal multi-agent behavior. The evidence suggests that graph-based latent representations are useful when anomalies arise from interactions, such as aggressive or unusual maneuvers that may not appear anomalous when an agent is considered alone. In this setting, latent-space density estimation can be more effective than reconstruction error alone, because normal interaction patterns occupy high-density regions of the learned latent space, while abnormal behaviors appear as low-density outliers. For prediction [98], masked graph autoencoder models use graph reconstruction to improve dynamic relational inference, especially when future trajectories depend on evolving and noisy interactions among multiple agents. For recovery [115], graph-based encoder–decoder models are especially relevant when missing trajectory states must remain consistent with relational structure, such as road-network topology, spatial-temporal relations between observed points, or shared travel-flow patterns. Their main caveat is that performance depends heavily on graph construction, edge semantics, and the quality of the road-network or contextual information used during decoding.

Compared with recurrent and Transformer-based models, graph autoencoders provide a more explicit representation of relational structure. Recurrent models are strong for temporal dependency modeling, but they do not directly encode interaction topology unless additional relational modules are added. Transformers can learn interactions through attention, but the relational structure remains implicit unless it is encoded through tokens or attention masks. Graph autoencoders make the relational structure part of the model input and learning objective, which is useful when agent-agent, location-location, or road-network relations are central to the task.

However, the conclusions for graph autoencoders should be interpreted cautiously because the reviewed evidence is limited. The available studies support graph autoencoders mainly for interaction-aware anomaly detection and dynamic multi-agent prediction, not for the full range of trajectory-analysis tasks. Their effectiveness also depends heavily on graph construction, node-feature design, edge definition, temporal windowing, and the choice of scoring or reconstruction objective. In particular, the evidence suggests that plain reconstruction error may be insufficient in some anomaly-detection settings, while latent-space density estimation or masked edge reconstruction can be more useful. Moreover, graph-based methods may be unnecessarily complex for single-agent trajectories or tasks where relational structure is weak, static, or poorly defined.

Therefore, graph autoencoders are best understood as specialized relational autoencoder models. Their main advantage is the ability to encode interactions among agents or spatial elements explicitly. Their main limitation is that this advantage only materializes when the graph structure is meaningful and task-relevant. Given the small number of reviewed studies, graph autoencoders should be presented as a promising but still underrepresented architecture family for trajectory GeoAI, particularly suitable for multi-agent, interaction-aware, and network-structured trajectory problems.

6.4.8. Summary of cross-architecture differences

Overall, the reviewed architectures differ primarily in the structural assumptions they impose on trajectory data. Feed-forward autoencoders provide fixed-vector compression, convolutional autoencoders exploit local grid-like or image-like structure, recurrent autoencoders model temporal order, variational autoencoders impose probabilistic latent distributions, Transformer-based autoencoders capture global token interactions, and graph autoencoders encode explicit relational structure. These differences indicate that architecture choice should be guided less by generic reconstruction accuracy alone and more by the form of trajectory structure required by the task.

Table 5
Property codes used in the architecture–task comparison matrix.

Code	Property captured in the comparison
Representational and methodological strengths	
LR	Learns compact or task-useful latent representations
LS	Supports separable, discriminative, or cluster-friendly latent structure
TM	Captures temporal or sequential motion dynamics
SS	Captures spatial, local, or map-like structure
CI	Models context, interactions, environment, or relational structure
UM	Supports uncertainty, multimodality, sampling, or generation
EF	Improves efficiency, scalability, or operational practicality
UL	Supports unlabeled, self-supervised, or weakly labeled learning settings
Caveats and limitations	
RD	Sensitive to representation, preprocessing, tokenization, or graph construction
RE	Affected by reconstruction-error, fidelity, or latent-capacity limitations
HT	Sensitive to hyperparameters, thresholds, masking ratios, or cluster number
CA	Often requires context, auxiliary modules, or task-specific mechanisms
EV	Evaluation, interpretation, ground truth, or benchmarking remains difficult
CD	Sensitive to contaminated, noisy, or weakly curated training data

6.5. Architecture-task comparison

Table 6 summarizes the qualitative relationship between autoencoder architecture families and trajectory-analysis tasks using the property codes defined in Table 5. The columns LR-UL describe representational and methodological properties, whereas the columns RD-CD describe recurring caveats and limitations. A checkmark (✓) indicates that a property or limitation is clearly evidenced or central to the corresponding architecture-task relation, a partial mark (±) indicates that it is conditional, hybrid, or secondary, and a dash (–) indicates that it is not central or not clearly supported by the reviewed evidence. The table should therefore be interpreted as a synthesis of methodological suitability and recurring architectural behavior, not as a direct performance ranking across models, since the reviewed studies differ in datasets, preprocessing pipelines, baselines, evaluation protocols, and task definitions.

The comparison shows that different architecture families contribute to trajectory analysis through different structural assumptions. Feed-forward, linear, and related autoencoder variants are most useful when trajectories or trajectory-derived quantities can be represented as fixed-length vectors. Their main contribution is compact latent representation learning and computational practicality, especially for compression, clustering, anomaly scoring, and some latent-space prediction pipelines. However, their usefulness depends strongly on representation design, preprocessing, reconstruction fidelity, and, for recovery/reconstruction tasks, whether missing trajectory states can be expressed in a fixed-length or matrix-like form rather than requiring explicit sequential or relational reasoning. Convolutional autoencoders become more suitable when trajectory information is encoded as local, spatial, map-like, image-like, or tensor-based structure. This explains their relevance for similarity computation, classification, anomaly detection, and some prediction or generation settings, but also explains their sensitivity to rasterization, resolution, window construction, and the choice of reconstruction target.

Recurrent autoencoders, especially LSTM and GRU variants, are the clearest fit when the task depends on temporal ordering and motion dynamics. Their strong association with TM across compression, clustering, anomaly detection, similarity computation, recovery, prediction, classification, and generation reflects their ability to model trajectories as ordered sequences rather than static vectors or images. In the recovery rows, this temporal modeling capacity is especially important because missing or low-sampling-rate trajectory states must be inferred from surrounding sequential context rather than reconstructed from a complete input alone. Nevertheless, recurrent models often require additional mechanisms when the task depends on context, interactions, long-range dependencies, uncertainty, or latent-space separability. VAE

Table 6
Qualitative architecture-task comparison of autoencoder families for trajectory representation learning. Columns LR-UL denote representational and methodological properties, while columns RD–CD denote recurring caveats and limitations, as defined in Table 5. A checkmark indicates a central or clearly evidenced property, a partial mark indicates a conditional, hybrid, or secondary property, and a dash indicates that the property is not central or not clearly supported by the reviewed evidence.

Architecture family	Task	LR	LS	TM	SS	CI	UM	EF	UL	RD	RE	HT	CA	EV	CD
Feed-forward/linear/AE	Compression	✓	–	–	–	–	–	✓	–	✓	✓	–	–	–	–
	Clustering	✓	✓	±	–	±	–	±	✓	✓	–	✓	–	✓	–
	Anomaly detection	✓	±	±	–	±	–	±	✓	✓	–	–	–	✓	✓
	Prediction	✓	±	±	–	±	±	✓	✓	✓	–	✓	✓	–	–
	Recovery/Reconstruction	✓	–	±	–	±	–	±	✓	✓	✓	±	–	✓	✓
Convolutional AE	Compression	✓	±	±	✓	–	±	±	–	✓	✓	–	–	–	–
	Clustering	✓	✓	–	✓	–	–	±	✓	✓	–	✓	–	–	–
	Anomaly detection	✓	±	±	✓	±	±	±	✓	✓	–	✓	–	✓	✓
	Similarity	✓	–	–	✓	–	–	✓	✓	✓	–	–	–	✓	–
	Prediction	✓	±	±	✓	✓	✓	±	–	✓	–	–	✓	–	–
	Classification	✓	✓	±	✓	±	–	±	✓	✓	–	–	–	✓	–
	Generation	✓	–	✓	±	±	✓	±	–	✓	–	–	–	✓	–
LSTM/GRU RNN AE	Compression	✓	–	✓	–	–	–	±	–	–	✓	–	–	–	–
	Clustering	✓	✓	✓	–	±	–	±	✓	✓	–	✓	✓	–	–
	Anomaly detection	✓	±	✓	–	±	–	±	✓	–	–	✓	–	✓	–
	Similarity	✓	–	✓	–	±	–	✓	✓	✓	–	–	–	–	–
	Prediction	✓	–	✓	–	±	±	±	–	–	–	–	✓	–	–
	Classification	✓	✓	✓	–	–	–	±	–	✓	–	–	–	✓	–
	Generation	✓	–	✓	–	±	±	±	–	–	–	–	✓	✓	–
	Recovery/Reconstruction	✓	–	✓	±	±	–	±	±	✓	✓	✓	✓	✓	✓
Transformer/Attention AE	Clustering	✓	✓	✓	±	±	±	±	✓	✓	–	–	✓	✓	–
	Anomaly detection	✓	±	✓	–	±	±	±	✓	–	–	✓	–	✓	✓
	Prediction	✓	–	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	–	–
	Recovery/Reconstruction	✓	–	✓	✓	✓	–	±	±	✓	✓	✓	✓	✓	✓
VAE/CVAE	Clustering	✓	✓	±	±	✓	✓	±	✓	✓	–	✓	–	–	–
	Anomaly detection	✓	±	±	±	±	✓	±	✓	✓	–	✓	–	–	✓
	Prediction	✓	–	✓	±	✓	✓	±	–	✓	–	–	✓	–	–
	Generation	✓	–	✓	±	±	✓	±	–	✓	–	–	–	✓	–
Graph AE	Anomaly detection	✓	±	±	✓	✓	±	±	✓	✓	–	–	–	✓	–
	Prediction	✓	–	±	✓	✓	–	±	✓	✓	–	–	✓	✓	–
	Recovery/Reconstruction	✓	–	✓	✓	✓	–	±	±	✓	✓	✓	✓	✓	✓

and CVAE models are distinguished by their probabilistic latent-space formulation. This explains their strong association with UM, especially in prediction and generation, where multiple plausible futures or synthetic trajectories may be required. Their limitations are mainly related to the choice of latent prior, conditioning variables, reconstruction-regularization balance, and evaluation of generated or probabilistic outputs.

Transformer-based autoencoders are most relevant when trajectory representation requires global token interaction, long-range dependencies, social context, or map-conditioned reasoning. This explains their strong profile in prediction and their partial relevance for clustering, anomaly detection, and recovery, especially when trajectory completion requires long-range temporal context, map-conditioned reasoning, or interactions between observed and missing trajectory states. However, their effectiveness depends heavily on tokenization, positional/context encoding, masking strategy, and task-specific design. Graph autoencoders are the most explicitly relational architecture family. Their checkmarks under SS and CI reflect their ability to model node features and edge-based interactions among agents, locations, or road-network elements. However, because only a small number of reviewed studies use graph autoencoders, their conclusions should be interpreted cautiously and primarily as evidence for interaction-aware anomaly detection, multi-agent prediction, and selected network-constrained recovery settings rather than as broad evidence across all trajectory tasks.

The task dimension of the table also reveals important patterns. Compression is mainly associated with compact representation learning and efficiency, but it is constrained by reconstruction fidelity and bottleneck capacity. Clustering depends not only on compact latent representations but also on latent separability, which explains the inclusion of LS and the recurring limitations related to hyperparameters, cluster number, and interpretability. Anomaly detection benefits from unlabeled or weakly labeled learning because abnormal examples are often scarce, but it is sensitive to contaminated training data, thresholding, and the interpretation of reconstruction or latent-space deviations. Similarity computation benefits from learned latent spaces that make trajectory comparison more efficient, but the usefulness of the learned distance still depends on how trajectories are represented. Prediction is the most demanding task because it often requires temporal dynamics, contextual information, uncertainty, and task-specific auxiliary mechanisms. Classification requires discriminative latent structure, while

generation requires sampleable or distribution-aware latent representations and remains difficult to evaluate because realism, diversity, and operational validity are not captured by a single metric.

Overall, the table supports the central conclusion that architecture choice should be guided by the structure that must be preserved for the target trajectory task. Feed-forward models are appropriate for fixed-vector compression and feature learning; convolutional models are appropriate for local spatial or tensorized trajectory structure; recurrent models are appropriate for temporal dynamics; VAE/CVAE models are appropriate for uncertainty, multimodality, and generation; Transformer-based models are appropriate for global token interactions and long-range context; and graph autoencoders are appropriate when explicit relational structure is central. At the same time, the limitation columns show that no architecture family is universally preferable. Most reviewed methods depend on representation design, task-specific objectives, auxiliary context, and domain-specific evaluation, which reinforces the need for task-aware architecture selection rather than generic autoencoder comparison.

6.6. Dataset-based metric comparisons and comparability limits

While the preceding analysis compared the reviewed studies in terms of task, architecture, and methodological design, an additional question is whether their reported results can be compared quantitatively across shared datasets. Such a comparison can provide a more concrete view of how different autoencoder-based approaches perform under partially similar experimental settings. However, this type of metric-level comparison is subject to important limitations in trajectory research. The use of the same dataset does not necessarily imply that studies follow the same task formulation, preprocessing pipeline, prediction horizon, supervision setting, sampling strategy, output format, or evaluation metric. Consequently, dataset overlap alone is not sufficient to support a fair numerical comparison.

For this reason, the metric-level comparison was limited to dataset groups where shared dataset usage was accompanied by sufficient alignment in task type and reported metrics. Among the reviewed papers, the clearest cases were the ETH/UCY and Stanford Drone Dataset benchmark family for pedestrian trajectory prediction, and GeoLife for transportation mode identification and mobility-behavior clustering. These datasets recur across multiple reviewed studies and include subsets of papers that address broadly comparable tasks with partially overlapping evaluation metrics. Even in these cases, the comparison should be interpreted cautiously, since differences in preprocessing, supervision, sampling strategy, and metric definitions may still affect the reported values.

Other datasets and data sources appear in more than two reviewed papers, but were not expanded into detailed performance tables because frequency of use alone does not guarantee comparability. NGSIM is a representative example. Although it is widely used, it is better treated here as a traffic trajectory data source rather than a single standardized benchmark protocol. The reviewed NGSIM-based papers use different subsets, preprocessing choices, prediction horizons, and task formulations, including trajectory compression, vehicle trajectory prediction, maneuver-aware forecasting, and communication-oriented spatio-temporal compression. A direct metric-level table would therefore risk comparing methods that solve substantially different problems.

A similar limitation applies to recurring benchmarks such as TrajNet++, highD, and Porto Taxi. Although these datasets provide useful evidence of dataset adoption in the reviewed literature, the corresponding studies differ in task setting, metric definition, evaluation protocol, or reported output. They are therefore discussed as examples of heterogeneous benchmark usage rather than as direct performance comparisons. This distinction is important because meaningful metric comparison requires alignment not only in dataset name, but also in task, evaluation horizon, output format, and reported metrics. Based on these selection criteria, the following subsubsections present the detailed metric-level comparisons for the ETH/UCY-SDD pedestrian trajectory benchmark family and for GeoLife.

6.6.1. ETH/UCY and SDD dataset-based prediction comparison

The ETH/UCY benchmark group and the Stanford Drone Dataset (SDD) provide the clearest opportunity for metric-level comparison among the reviewed trajectory prediction studies, since several papers evaluate pedestrian trajectory forecasting under broadly similar observation-prediction settings. Table 7 summarizes the reported ADE/FDE values for the reviewed papers that use at least one of these benchmarks. However, the table should not be interpreted as a strict leaderboard. The comparison is affected by three main factors: (i) not all methods evaluate on both ETH/UCY and SDD; (ii) some papers introduce additional sampling or post-processing strategies, such as final-position clustering, which improve best-of-(K) displacement errors but change the inference procedure; and (iii) some studies report additional probabilistic or environment-aware metrics, such as NLL, Brier-ADE/FDE, or collision-related measures, which are not consistently available across all papers. For this reason, the final column marks the metric comparability of each row or reported variant: A denotes direct comparison under broadly aligned benchmark and metric settings, B denotes partial comparison due to dataset coverage, sampling strategy, post-processing, or additional evaluation differences, and C denotes contextual evidence only. When a row reports multiple variants, the main label refers to the non-marked result, while daggered values (†) are assigned the tier indicated in the final column. Asterisked values (∗) indicate an additional unit or protocol caveat and should be interpreted according to the corresponding table note.

Table 7
Metric-level comparison on shared pedestrian trajectory prediction benchmarks. Values are reported as ADE/FDE where available; ETH/UCY values are in meters unless otherwise marked, while SDD values are in pixels. The final column indicates metric comparability: A = directly comparable, B = partially comparable, and C = contextual only. (†) denotes the SocialVAE final-position clustering variant, treated as B due to post-processing; (∗) denotes normalized-pixel values rather than the meter-based ETH/UCY convention.

Ref.	Method	Architecture Family	ETH/UCY (avg.)	SDD	Additional metrics/variants	Metric comparability
[70]	AFC-RNN	RNN + adaptive forgetting controller	0.19/0.32	7.17/11.44	—	A
[112]	SocialVAE	Timewise VAE + GRU	0.24/0.42 0.21/0.33†	8.88/14.81 8.10/11.72†	FPC: 0.21/0.32–0.33 ETH/UCY; 8.10/11.72 SDD; NLL	A †: B
[98]	TUTR	Transformer encoder-decoder	0.21/0.36	7.76/12.69	brier-ADE/FDE	A
[110]	MUSE-VAE	Multi-scale CVAE	—	6.36/11.10	KDE NLL, ECFL	B
[111]	Multimodal CVAE/Trajectron++	CVAE + spatio-temporal graph RNN	0.21/0.41	—	BoN ADE/FDE, KDE NLL	B
[79]	RNN encoder-decoder + ST Att.	RNN encoder-decoder + social/physical attention	0.144/0.286*	—	Normalized-pixel ADE/FDE	C

Table 8
Metric-level comparison of GeoLife-based classification studies. The comparison is limited because the methods differ in supervision regime, preprocessing, segmentation, and metric reporting. B denotes partial comparability within the same broad task and data source.

Ref.	Method	Architecture family	Accuracy	Weighted F-measure	Metric comparability
[56]	SECA	Semi-supervised convolutional autoencoder + CNN classifier	0.768	0.764	B
[21]	Dual-CSA	Dual CNN-based supervised autoencoder with predefined class centroids	0.89475	—	B

Table 9
Metric-level comparison of GeoLife-based trajectory clustering studies. Daggered DETECT values are taken from the VAMBC comparison table and are therefore directly comparable with VAMBC under NMI, ARI, and ACC. Asterisked uRSAA values correspond to a filtered three-class GeoLife subset containing bike, bus, and car trajectories. A = directly comparable, B = partially comparable, and C = contextual only.

Ref.	Method	Architecture family	RI	MI	Purity	FMI	NMI	ARI	ACC/Accuracy	Class P/R/F1	Metric comparability
[35]	DETECT	RNN autoencoder + embedded clustering refinement	0.76	1.26	0.89	0.81	0.644†	0.646†	0.800†	—	A†/B
[100]	VAMBC	Variational sequence clustering	—	—	—	—	0.697	0.700	0.825	—	A
[97]	uRSAA	U-type robust sparse autoencoder + attention	—	—	—	—	—	—	68.3*	Bike: 0.57/0.52/0.54 Bus: 0.64/0.95/0.76 Car: 0.98/0.58/0.73	C

The strongest direct comparison is obtained for AFC-RNN, SocialVAE, and TUTR, which all report ADE/FDE on both ETH/UCY and SDD. Their results show that recent RNN-based, VAE-based, and Transformer-based designs all achieve competitive performance on standard pedestrian trajectory prediction benchmarks, with differences increasingly shaped by how multimodality is handled. SocialVAE improves its displacement errors when final-position clustering is applied, while TUTR explicitly avoids inference-time post-processing and instead reports probability-aware Brier metrics alongside ADE/FDE. AFC-RNN, by contrast, focuses on improving recurrent temporal modeling through adaptive forgetting control. These differences suggest that similar ADE/FDE values may correspond to different practical trade-offs in inference cost, probabilistic interpretability, and multimodal sample selection.

The remaining rows are useful but less directly comparable. MUSE-VAE overlaps with this group mainly through SDD and emphasizes environment-aware forecasting using metrics such as KDE NLL and ECFL. The Multimodal CVAE/Trajectron++ study provides relevant generative-model evidence on ETH/UCY, but its results are reported in a broader best-of-(N) and likelihood-oriented evaluation setting. Finally, the RNN encoder-decoder with spatio-temporal attention is included as contextual evidence because its reported values use a normalized-pixel scale rather than the meter-based ETH/UCY convention used in later studies. Overall, this benchmark group supports a cautious but meaningful comparison: ADE/FDE remains the most common basis for cross-paper evaluation, but architectural conclusions should be drawn together with the associated sampling strategy, uncertainty modeling, and metric protocol.

6.6.2. *GeoLife dataset-based classification and clustering comparisons*

GeoLife appears in the reviewed literature under different task formulations, and therefore it is separated here into two metric-comparison groups. The first group concerns transportation mode identification or classification, where the objective is to assign trajectory segments to transport modes such as walking, biking, bus, driving, or train. The second group, discussed separately, concerns trajectory or mobility-behavior clustering, where the objective is to discover behavioral groups from trajectory representations rather than predict predefined transport labels. This separation is necessary because the same data source does not imply a shared evaluation protocol: the studies differ in supervision regime, segmentation strategy, feature construction, and reported metrics. A further GeoLife-based study addresses missing trajectory recovery rather than classification or clustering, using segmented GeoLife trajectories with artificially constructed missing points and evaluating reconstruction accuracy under different missing-rate configurations. Because its objective, output format, and metrics are centered on coordinate recovery rather than transport-mode prediction or mobility-pattern clustering, it is not included in the metric-level GeoLife comparisons in Tables 8 and 9.

Table 8 summarizes the two GeoLife-based studies that address transportation mode identification or classification. Both methods rely on autoencoder-based representation learning, but they use different learning strategies. SECA combines a convolutional autoencoder with a CNN classifier in a semi-supervised framework, using both labeled and unlabeled GPS trajectory segments. Dual-CSA instead uses a dual supervised autoencoder with predefined class centroids, combining recurrence-plot features and feature-segment representations to improve class separation in the latent space.

The results show that Dual-CSA reports a higher GeoLife accuracy than SECA. However, this comparison should be interpreted cautiously rather than as a strict leaderboard. SECA is evaluated under a semi-supervised setting with varying proportions of labeled data and reports both accuracy and weighted F-measure, while Dual-CSA reports its best GeoLife accuracy and provides class-level classification information but does not report a directly equivalent weighted F-measure summary. For this reason, both rows are assigned metric comparability level B: they share the same broad task and GeoLife data source, but differ in training protocol, preprocessing, and metric reporting.

The second GeoLife comparison concerns trajectory clustering and mobility-behavior clustering. Unlike the transportation mode identification studies, these methods do not primarily classify trajectory segments into predefined transport modes. Instead, they learn trajectory representations and group trajectories according to latent mobility patterns, contextual transitions, or movement behavior. Table 9 summarizes the reported GeoLife results for the reviewed clustering-oriented studies.

The most comparable pair is DETECT and VAMBC. Both address mobility-behavior clustering from context-enriched trajectory sequences, where raw trajectories are transformed through stay-point extraction and nearby geographic context before clustering. DETECT reports its original GeoLife results using RI, MI, Purity, and FMI, while

VAMBC re-evaluates DETECT alongside other baselines using NMI, ARI, and ACC. For this reason, the DETECT row includes both the originally reported metrics and the VAMBC-table metrics, with daggered values indicating the latter. This makes the DETECT-VAMBC comparison more informative than comparing DETECT's original metrics directly against VAMBC's metrics.

The uRSAA row is included as contextual evidence rather than as a direct comparison. Although it also uses GeoLife, it evaluates trajectory clustering on a filtered three-class subset containing bike, bus, and car trajectories, with 100 trajectories sampled for each class. Its reported precision, recall, F1, and accuracy therefore reflect a narrower transportation-category clustering setup rather than the six-class mobility-behavior setting used by DETECT and VAMBC. Overall, the GeoLife clustering group shows that shared use of GeoLife can support meaningful comparison when task and metric protocols are aligned, but it also reinforces the need to distinguish mobility-behavior clustering from more restricted trajectory-category clustering.

6.7. Practitioner guidelines for architecture selection

The preceding comparisons indicate that autoencoder architecture selection should begin with the structure of the trajectory problem rather than with architecture popularity alone. In practice, the most important questions are: what form the trajectory representation takes, what information must be preserved in the latent space, whether uncertainty or multimodality is central to the task, and whether the application requires explicit modeling of context or interactions. Therefore, practitioners should treat the architecture families discussed above as complementary design choices rather than as universally ranked alternatives.

A first guideline is to match the model backbone to the input representation. If trajectories are represented as fixed-length vectors, handcrafted features, or flattened windows, feed-forward or linear autoencoders provide a simple and computationally practical baseline. They are appropriate when compact representation learning is the main objective, but they should not be expected to recover temporal, spatial, or relational structure unless that information is already encoded in the input features. If trajectories are transformed into raster images, recurrence plots, gridded maps, occupancy fields, or multichannel tensors, convolutional autoencoders are more appropriate because they exploit local spatial or tensor-based structure. However, their effectiveness depends strongly on the quality of the trajectory-to-grid transformation, resolution, and reconstruction target. If the trajectory is naturally represented as an ordered sequence, recurrent autoencoders, especially LSTM or GRU variants, are usually the most direct choice because they model temporal dependencies and variable-length movement histories more naturally than fixed-vector or image-based models.

A second guideline is to match the latent-space behavior to the task objective. Compression requires compactness and reconstruction fidelity, so simple feed-forward, linear, convolutional, or recurrent autoencoders may be sufficient depending on whether the input is vector-based, image-based, or sequential. Clustering requires not only compact latent representations but also separable and interpretable latent structure; therefore, practitioners should consider clustering-aware objectives, contrastive losses, mixture priors, or post-hoc clustering validation rather than relying on reconstruction loss alone. Anomaly detection is suitable for reconstruction-based or density-based autoencoders when normal behavior can be learned from mostly clean data, but it requires careful threshold calibration and inspection of false positives, especially when the training data may contain undetected anomalies. Similarity computation benefits from latent spaces that preserve pairwise trajectory relationships and reduce computational cost, but the chosen representation must match the intended notion of similarity, such as route geometry, temporal behavior, activity semantics, or network proximity.

A third guideline is to use probabilistic and context-aware models when deterministic reconstruction is insufficient. For trajectory prediction and generation, future movement is often uncertain and multimodal. In such cases, VAE or CVAE-based models are preferable because they provide sampleable latent distributions and can represent multiple plausible outcomes. Conditional variants are especially useful when predictions depend on observed history, map context, destination, agent type, or neighboring agents. Transformer-based autoencoders are appropriate when the task requires long-range dependency modeling or heterogeneous token interactions, such as joint reasoning over trajectories, maps, and social context. However, their performance depends heavily on tokenization, positional encoding, masking strategy, and computational budget. Graph autoencoders should be considered when the core structure of the problem is relational, such as interaction-aware anomaly detection, multi-agent prediction, or road-network-based reasoning. Because graph autoencoders remain underrepresented in the reviewed literature, they should currently be considered carefully and with high task specificity in mind.

A fourth guideline is to begin with the simplest architecture that matches the dominant structure of the task and add complexity only when the task requires it. For example, a fixed-vector compression problem does not necessarily require a Transformer or graph autoencoder. A short-horizon sequence prediction problem may be adequately handled by an LSTM or GRU, while a long-horizon, interaction-aware, map-conditioned prediction problem may require a CVAE, Transformer, graph module, or hybrid architecture. Similarly, if the main challenge is uncertainty or multimodality, adding a variational mechanism may be more important than changing the entire backbone. If the main challenge is local spatial structure, convolutional encoding may be more useful than attention. If the main challenge is explicit agent-agent or road-network relation modeling, graph-based representation becomes more relevant.

Finally, practitioners should evaluate architecture choice using task-aligned metrics rather than reconstruction loss alone. Compression should be assessed through both compression ratio and reconstruction fidelity; clustering should combine quantitative validity indices with domain interpretability; anomaly detection should examine threshold sensitivity, false alarms, and contamination of normal training data; prediction should report displacement error together with uncertainty or multimodality when applicable; and generation should evaluate realism, diversity, and operational plausibility. When shared benchmarks exist, reported metrics can provide useful supporting evidence, but they should not override representation and task suitability, especially when preprocessing pipelines, prediction horizons, and evaluation protocols differ across studies.

7. Available resources

This section summarizes practical resources that support research on autoencoder-based trajectory GeoAI. Section 7.1 first reviews publicly available datasets used for trajectory analysis and related GeoAI tasks, highlighting their application domains, data types, and availability. Section 7.2 then presents relevant software resources, including model implementations, trajectory-analysis repositories, simulation tools, and general-purpose frameworks that can support experimentation, evaluation, and future method development.

7.1. Public datasets

Publicly available datasets play a critical role in the development and evaluation of trajectory analysis methods, as they enable reproducibility and facilitate fair comparison between approaches. In the context of GeoAI and autoencoder-based models, these datasets span a wide range of domains, including transportation, human mobility, autonomous driving, aviation, and maritime navigation. Many of the works reviewed in this paper rely on such established benchmarks, which have become standard for evaluating tasks such as trajectory prediction, clustering, and anomaly detection. Table 10 provides an overview of the most relevant datasets used in this domain, categorized by application

Table 10
Publicly available datasets.

Dataset	Domain	Type	Availability	Source
T-Drive	Taxi/Urban Mobility	Real	Public	[66]
Porto Taxi (ECML PKDD 2015)	Taxi/Urban Mobility	Real	Public	[68]
GeoLife	Human Mobility	Real	Public	[36–38]
SHL Dataset	Human Mobility	Real	Public	[55]
NGSIM	Vehicle	Real	Public	[51]
highD	Vehicle	Real	Public (request)	[62]
inD	Vehicle	Real	Public (request)	[107]
INTERACTION	Autonomous Driving	Real	Public	[42]
Argoverse Motion Forecasting	Autonomous Driving	Real	Public	[96]
Argoverse 2 Motion Forecasting	Autonomous Driving	Real	Public	[43,44]
Waymo Open Motion	Autonomous Driving	Real	Public (request)	[45]
nuScenes	Autonomous Driving	Real	Public	[109]
KITTI	Autonomous Driving	Real	Public	[169]
ApolloScape	Autonomous Driving	Real	Public	[170]
ETH Pedestrians	Pedestrian	Real	Public	[71]
UCY Pedestrians	Pedestrian	Real	Public	[72]
Stanford Drone Dataset (SDD)	Pedestrian/Multi-agent	Real	Public	[73]
JAAD	Pedestrian/Vehicle	Real	Public	[76,77]
PIE	Pedestrian/Vehicle	Real	Public	[78]
TrajNet + +	Pedestrian	Real	Public	[60]
OpenSky Network	Aviation	Real	Public	[168]
MarineCadastre AIS	Maritime	Real	Public	[105]
SUMO Traffic (LuST) Scenario	Synthetic	Synthetic	Public	[171]
Edinburgh Informatics Forum	Pedestrian/Outdoor	Real	Public	[81]

area, data type, and availability. Official sources are provided through citations; in addition, alternative sources are included where necessary to facilitate dataset retrieval when official sources are temporarily unavailable or difficult to access.

At the same time, a significant portion of the literature makes use of custom datasets, often constructed by collecting raw data from publicly available platforms, such as the OpenSky Network [168], or from domain-specific sensing systems, followed by task-specific preprocessing. Consequently, even when common data sources are used, the resulting datasets may differ substantially across studies, limiting direct comparability. The datasets included in Table 10 are not restricted solely to those explicitly used in the reviewed papers. Widely adopted and well-established benchmarks have also been incorporated to provide a more comprehensive overview of the field.

7.2. Software and frameworks

A range of publicly available software resources supports trajectory analysis tasks, including model implementations, simulation tools, and general-purpose frameworks. Table 11 summarizes the most relevant resources identified in this work, covering both repositories associated with the methods reviewed in this paper and additional tools for trajectory synthesis and model development. The listed implementations span tasks such as prediction, clustering, classification, anomaly detection, similarity computation, and generation, while simulation platforms enable controlled data generation and experimentation. Together, these resources provide practical support for developing, evaluating, and extending trajectory-based approaches.

8. Emerging paradigms beyond autoencoders for trajectory GeoAI

Recent GeoAI research has increasingly explored paradigms that fall outside the core taxonomy of autoencoder-based trajectory representation learning considered in this survey. This section briefly discusses two adjacent directions: emerging trajectory modeling approaches beyond autoencoders, such as diffusion models, flow matching, and LLM-based methods, and broader remote sensing GeoAI developments based on masked autoencoding, self-supervised learning, hyperspectral foundation models, and multimodal Earth-observation models. These works are not all part of the analyzed trajectory-autoencoder corpus, but they help contextualize the role of autoencoders within the wider evolution of GeoAI representation learning.

8.1. Emerging trajectory modeling paradigms beyond autoencoders

This subsection discusses trajectory-centered paradigms that have recently emerged beyond conventional autoencoder-based representation learning. It focuses on diffusion models, flow matching, and LLM-based approaches, emphasizing how these methods address trajectory generation, prediction, reasoning, simulation, and modeling, and how they relate to the autoencoder-based methods reviewed in this survey.

8.1.1. Diffusion models for trajectory generation and multimodal prediction

Diffusion models represent one of the most important recent generative paradigms for trajectory analysis, particularly in tasks where the objective is to model uncertainty, generate realistic movement samples, or produce multiple plausible future trajectories. Unlike autoencoder-based models, which typically learn a compact latent representation and reconstruct or decode trajectories from that representation or use a probabilistic latent-space formulation, diffusion models formulate generation as a gradual denoising process. In this setting, trajectory samples are progressively corrupted with noise during training, while generation is performed by learning the reverse process that transforms random noise into realistic trajectories. This formulation is especially relevant for trajectory data because movement behavior is inherently stochastic, and many trajectory tasks require modeling not only one expected output, but a distribution of plausible mobility patterns.

A first group of diffusion-based studies focuses on synthetic trajectory generation and human mobility simulation. DiffTraj [174] applies a diffusion probabilistic model to GPS trajectory generation, aiming to produce privacy-preserving synthetic trajectories that retain the spatial-temporal properties and utility of real data. The model reconstructs geographic trajectories from white noise through a reverse denoising process and introduces a Traj-UNet architecture to incorporate conditional information during generation. In this sense, DiffTraj extends earlier VAE- and GAN-based trajectory generation approaches by directly modeling the trajectory distribution through iterative denoising rather than relying on a single latent bottleneck or adversarial training. Building on this direction, ControlTraj [175] introduces a topology-constrained diffusion framework for controllable trajectory generation. Its main contribution is the integration of road-network structure into the generative process through a masked road-segment autoencoder and a geographic denoising UNet, allowing generated trajectories to follow topological constraints and adapt to new urban environments. This is particularly relevant from a GeoAI perspective, because it shows that diffusion-based trajectory generation can be conditioned not only on trip-level attributes, but also on explicit geographic structure.

A related but broader mobility-oriented direction is represented by Ref. [176], which models human mobility simulation as a trajectory generation process based on uncertainty reduction. Rather than focusing only on coordinate-based trajectory synthesis, the method addresses discrete location-index trajectories and aims to learn a universal mobility pattern from a trajectory dataset. The model is evaluated not only for generation, but also through zero-shot trajectory prediction and reconstruction, suggesting that diffusion-based generation can serve as a more general mobility representation mechanism. This point is important for the present survey because it connects diffusion models to the broader question of generalizable trajectory representation learning. Whereas autoencoders usually learn representations through reconstruction objectives, TrajGDM suggests that iterative generative

Table 11
Official and Community Software implementations for trajectory Analysis.

Project	Task	Ref.	Source
Trajectory clustering via deep representation learning	Clustering	[11]	[GitHub]
Outlier Vehicle Trajectory Detection Using Deep Autoencoders in Santiago, Chile	Anomaly Detection	[34]	[GitHub]
A Denoising Hybrid Model for Anomaly Detection in Trajectory Sequences	Anomaly Detection	[16]	[GitHub]
Deep Representation Learning for Trajectory Similarity Computation	Similarity Computation	[13]	[GitHub]
PoPPL: Pedestrian Trajectory Prediction by LSTM With Automatic Route Class Clustering	Prediction	[80]	[GitHub]
Vehicles Trajectory Prediction Using Recurrent VAE Network	Prediction	[82]	[GitHub]
Dual Supervised Autoencoder Based Trajectory Classification Using Enhanced Spatio-Temporal Information	Classification	[21]	[GitHub]
Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using GPS Trajectory Data	Classification	[56]	[GitHub]
Pedestrian Trajectory Prediction Based on Deep Convolutional LSTM Network	Prediction	[20]	[GitHub]
Context-Aware Timewise VAEs for Real-Time Vehicle Trajectory Prediction	Prediction	[108]	[GitHub]
Deep generative modelling of aircraft trajectories in terminal maneuvering areas	Generation	[58]	[GitHub]
Exploring Dynamic Context for Multi-path Trajectory Prediction	Prediction	[59]	[GitHub]
SocialVAE: Human Trajectory Prediction using Timewise Latents	Prediction	[112]	[GitHub]
Simulation of Urban MObility (SUMO)	Trajectory Synthesis	[172]	[Link]
Multi-Agent Transport Simulation (MATSim)	Transport Simulation	–	[Link]
Open-Source Simulator for Autonomous Driving (CARLA)	Driving Simulation	[173]	[Link]
Autoencoder Implementation Library (pytorch)	Autoencoder models	–	[GitHub]
Autoencoder Implementation Library (keras)	Autoencoder models	–	[GitHub]
Autoencoder Implementation Framework (pytorch)	Autoencoder models	–	[GitHub]

modeling may also capture transferable mobility structures that can support multiple trajectory tasks.

A second group of diffusion-based studies focuses on stochastic trajectory prediction, especially in pedestrian and interaction-aware environments. The method presented in Ref. [177] formulates pedestrian trajectory prediction as a reverse process of motion indeterminacy diffusion, where future motion is generated by progressively reducing uncertainty from ambiguous walkable areas toward plausible future trajectories. This differs from many earlier stochastic prediction methods based on GANs or CVAEs, where multimodality is often represented through a latent variable. Instead, MID explicitly models the transition from indeterminate to determinate motion using a diffusion process conditioned on observed trajectories and social interactions. However, the iterative nature of diffusion introduces an important computational limitation, since many denoising steps may be required during inference. Leapfrog [178] addresses this limitation by introducing a leapfrog initializer that directly estimates an expressive intermediate denoised distribution, thereby skipping many denoising steps while maintaining diverse and accurate predictions. Taken together, MID and Leapfrog illustrate both the strength and the main practical challenge of diffusion-based trajectory prediction: diffusion models can represent complex multimodal future distributions, but their deployment depends on reducing the cost of iterative sampling.

In autonomous-driving and multi-agent settings, diffusion models have also been used to model joint future distributions and controllable prediction scenarios. MotionDiffuser [179] represents the future trajectories of multiple agents as a joint distribution learned through conditional diffusion. Instead of predicting each agent independently, the model uses a permutation-invariant denoiser to generate consistent multi-agent futures, while also supporting constrained sampling through differentiable cost functions. This makes diffusion particularly attractive for interaction-aware trajectory modeling, where plausible futures must respect not only individual motion histories but also agent-agent interactions, road context, and possible physical constraints. SingularTrajectory [180] extends the use of diffusion toward a more general trajectory prediction setting by unifying different prediction protocols within a shared Singular space and refining adaptive prototype paths through a diffusion-based predictor. Its emphasis on generality across stochastic, deterministic, domain-adaptation, momentary-observation, and few-shot settings further highlights the role of diffusion models as flexible generators of future motion rather than task-specific predictors.

Overall, diffusion-based trajectory models complement autoencoder-based methods rather than replacing them. Autoencoders remain particularly effective for compact representation learning, reconstruction, anomaly scoring, clustering, similarity computation, and efficient latent-space operations. Diffusion models, in contrast, are especially suitable when the goal is high-fidelity generation, multimodal prediction, controllable sampling, or synthetic trajectory simulation. Their strengths are most visible in tasks where many plausible outputs exist, such as GPS trajectory generation, human mobility simulation, pedestrian forecasting, and multi-agent autonomous-driving prediction. At the same time, their dependence on iterative denoising introduces computational overhead, and their effectiveness often depends on the quality of conditioning information, such as road topology, social context, environmental maps, or agent interactions. This creates a natural transition toward flow matching, which has recently emerged as a related generative paradigm aiming to retain the distribution-modeling benefits of diffusion while improving sampling efficiency.

8.1.2. *Flow matching for efficient generative trajectory modeling*

Flow matching has recently emerged as a closely related alternative to diffusion-based generative modeling, with particular relevance for trajectory tasks that require both multimodal generation and efficient inference. While diffusion models generate trajectories through iterative denoising, flow matching learns a time-dependent vector field that transports samples from a simple source distribution, such as Gaussian noise, toward the target data distribution. Foundational work on flow matching introduced this paradigm as a simulation-free training framework for continuous normalizing flows, showing that conditional flow matching can regress tractable per-sample vector fields without explicitly estimating an intractable marginal vector field [181]. This formulation also connects flow matching to diffusion, since diffusion paths can be treated as a special case, while optimal transport-based paths can define straighter transformations that are often more efficient to integrate [181,182]. From the perspective of trajectory analysis, this is important because it directly targets one of the main limitations identified in diffusion-based trajectory models: the cost of repeated denoising steps during inference.

The first trajectory-oriented applications of flow matching emphasize this efficiency advantage. T-CFM [183] applies conditional flow matching to trajectory forecasting and generation, unifying prediction and planning within a common generative formulation. Instead of reconstructing trajectories through a long denoising chain, the model learns a time-varying vector field that maps noisy trajectory samples toward feasible future trajectories or plans. This allows the same framework to be evaluated across adversarial tracking, aircraft trajectory forecasting, and long-horizon planning. In comparison with diffusion-based trajectory generation, the main contribution of T-CFM is therefore not a

different trajectory representation, but a more efficient generative mechanism that preserves multimodal sample quality while substantially reducing sampling cost. This makes flow matching particularly relevant for robotic and mobility applications where trajectory generation must support real-time decision-making.

More recent flow-matching models adapt this principle to multimodal human and traffic trajectory prediction. MoFlow [184] focuses on human trajectory forecasting and explicitly addresses two limitations of diffusion-style sampling: independently sampled futures may fail to cover diverse modes coherently, and iterative ODE or denoising-based sampling remains expensive. To address this, MoFlow predicts multiple future trajectories jointly and introduces a flow-matching loss that encourages at least one prediction to match the ground truth while maintaining diversity across the full set of predicted futures. Its IMLE-based distillation further converts the teacher flow model into a one-step student model, showing how flow-based trajectory predictors can preserve multimodality while becoming suitable for faster deployment. TrajFlow [185] extends similar ideas to large-scale autonomous-driving motion prediction. It formulates multimodal motion forecasting as a flow-matching problem conditioned on agent history and road-map context, but differs from independent and identically distributed generative sampling by producing multiple plausible trajectories in a single pass. Its use of confidence ranking and self-conditioning further highlights a key practical concern in trajectory forecasting: efficient generation alone is insufficient unless predicted trajectories are also coherent, diverse, and reliably ranked for downstream decision-making.

Overall, flow matching occupies an intermediate position between autoencoder-based trajectory representation learning and diffusion-based generative modeling. Like diffusion, it is well suited to multimodal prediction and generative trajectory modeling; however, it replaces iterative stochastic denoising with learned probability flows that can often be sampled with fewer function evaluations. Compared with autoencoders, flow-matching methods are less focused on compact latent reconstruction and more focused on transporting distributions between noise and realistic trajectory futures. Therefore, their most immediate relevance lies in efficient trajectory forecasting, long-horizon planning, and multimodal motion generation rather than in compression, anomaly scoring, or similarity-based trajectory retrieval. At the same time, flow matching remains an emerging direction in trajectory GeoAI, with most current applications concentrated in robotics, pedestrian forecasting, and autonomous-driving motion prediction. Future work could explore hybrid designs in which autoencoder-derived latent trajectory representations or geographic context embeddings condition flow-matching models, combining the representational compactness of autoencoders with the efficient multimodal generation of flow-based models.

8.1.3. LLM-based trajectory modeling and semantic reasoning

Large Language Model (LLM)-based methods represent a different emerging direction from diffusion and flow-matching models. Rather than focusing primarily on reconstructing trajectories, denoising samples, or transporting probability distributions, LLM-based approaches aim to exploit semantic knowledge, contextual reasoning, natural-language interfaces, and generalization capabilities. This makes them particularly relevant for trajectory tasks where movement data must be interpreted in relation to human intentions, activity semantics, traffic rules, scene context, or high-level behavioral descriptions. In this sense, LLM-based trajectory methods should be viewed as complementary to autoencoder-based representation learning. Autoencoders provide compact latent representations of movement patterns, while LLMs provide a mechanism for incorporating semantic and contextual knowledge that is often absent from purely geometric trajectory encoders.

One important group of LLM-based studies focuses on semantic interpretation of human mobility. TSI-LLM [186] formulates trajectory semantic inference as the extraction of occupational category, activity sequence, and trajectory description from individual mobility traces. Instead of treating trajectories only as coordinate or region sequences, it reformats raw movement data into spatio-temporal trajectory chains enriched with temporal information, distances, and POI-related attributes, and then uses context-inclusive prompts to guide semantic inference. Similarly, Mobility-LLM [187] addresses check-in sequence analysis by reprogramming POI visit records into representations that LLMs can process, using a visiting intention memory network and human travel preference prompts to support tasks such as next-location prediction, trajectory user linking, and time prediction. These studies show that LLMs are especially useful when trajectory analysis requires interpreting why movements occur, not only where or when they occur. This distinguishes them from most autoencoder-based trajectory models, which typically learn latent representations from reconstruction or prediction objectives without explicitly modeling activity purpose, occupation, travel preference, or narrative trajectory descriptions.

A second group of methods uses LLMs to enhance trajectory prediction by injecting motion cues, scene semantics, or explainable reasoning. LG-Traj [188] applies LLMs to pedestrian trajectory prediction by generating motion cues from observed pedestrian trajectories, which are then combined with past coordinates, future motion clusters, and social-interaction modeling. In this case, the LLM does not replace the trajectory prediction network; instead, it provides additional semantic cues that improve the representation of motion patterns. LC-LLM [189] follows a more direct language-modeling formulation for autonomous driving lane change prediction. It converts heterogeneous driving information, including target-vehicle states, surrounding-vehicle states, and map context, into natural language prompts, and fine-tunes the LLM to predict lane change intentions, future trajectories, and explanations. This highlights a central advantage of LLM-based methods: they can connect trajectory prediction with interpretable reasoning about vehicle interactions, traffic rules, and possible intentions. Traj-LLM [190] further explores the use of pre-trained LLMs for autonomous driving trajectory prediction without relying on explicit prompt engineering. By transforming sparse agent and lane context into representations compatible with pre-trained language models and combining them with lane-aware probability learning, it positions LLMs as high-level interaction modeling modules rather than purely textual reasoning engines.

A third emerging direction uses LLMs for trajectory generation, simulation, and automated modeling workflows. Trajectory-LLM [191] proposes a language-based data generator for autonomous driving trajectory prediction. Instead of directly translating textual interaction descriptions into trajectories, it introduces an intermediate interaction-behavior-trajectory process, where driving behaviors and behavioral logic guide the generation of more realistic vehicle trajectories. This is relevant to trajectory GeoAI because it suggests that LLMs can serve as controllable interfaces for creating synthetic trajectory data, potentially reducing dependence on costly real-world data collection. In human mobility simulation, TrajLLM [192] uses a modular agent-based framework with persona generation, activity selection, destination prediction, and memory to generate realistic daily mobility trajectories. This direction differs from diffusion or autoencoder-based generation because the simulated trajectories are driven by interpretable activity and persona-level reasoning rather than only learned geometric distributions. TrajAgent [193] extends the role of LLMs even further by proposing an agentic framework for automated trajectory modeling across multiple tasks, datasets, and specialized models. In this setting, the LLM acts as a controller that supports task understanding, model selection, data processing, optimization, and collaboration with smaller trajectory models.

Overall, LLM-based trajectory methods are not direct replacements for autoencoder-based approaches. Their main contribution lies in semantic enrichment, explainability, task automation, and high-level contextual reasoning, while their limitations include dependence on prompt or input formatting, grounding quality, computational cost, and the risk of producing semantically plausible but geometrically inconsistent outputs. Autoencoders remain better suited for compact representation learning, reconstruction, anomaly detection, similarity

computation, and efficient latent-space operations. LLMs, by contrast, are strongest when trajectory data must be connected to intentions, activities, rules, narratives, or user-facing explanations. A promising future direction is therefore the development of hybrid models in which autoencoder-based latent trajectory representations provide compact and geometrically grounded movement encodings, while LLMs provide semantic interpretation, explanation, task guidance, or controllable generation constraints.

8.1.4. Relationship to autoencoder-based trajectory representation learning

The recent emergence of diffusion models, flow matching, and LLM-based methods does not diminish the relevance of autoencoder-based trajectory representation learning, but rather clarifies its position within a broader methodological landscape. Autoencoders remain primarily suited to learning compact latent representations that support reconstruction, compression, anomaly detection, clustering, similarity estimation, and downstream trajectory analysis. Their main strength lies in transforming complex movement sequences into structured latent spaces that can be reused across multiple trajectory tasks. In contrast, diffusion and flow-matching methods are more directly oriented toward probabilistic generation and multimodal forecasting, where the goal is to produce diverse and realistic future trajectories or synthetic mobility patterns. LLM-based methods, meanwhile, address a different limitation of many trajectory models by introducing semantic reasoning, activity interpretation, explainability, and controllable natural-language interfaces.

At the same time, these paradigms should not be interpreted as directly comparable under a single performance criterion. The reviewed autoencoder-based methods and the emerging diffusion, flow-matching, and LLM-based approaches often target different tasks, datasets, output formats, and evaluation protocols. Therefore, their relationship is better understood in terms of complementarity rather than replacement. Diffusion models and flow matching may be useful when trajectory uncertainty and multimodality are central, while autoencoder-based representations remain valuable when compactness, reconstruction, and latent-space analysis are important. Similarly, LLMs can provide semantic context or explanatory layers, but they do not inherently solve the need for geometrically grounded and efficient trajectory encodings.

A promising future direction is the development of hybrid trajectory GeoAI frameworks that combine these strengths. Autoencoder-derived latent representations could provide compact and structured trajectory encodings for conditioning diffusion or flow-matching models, while LLMs could support semantic labeling, task guidance, explanation, or controllable generation. Such combinations may help bridge the gap between efficient representation learning, high-fidelity generative modeling, and semantically meaningful trajectory analysis. However, this remains an emerging direction, and further work is needed to evaluate these integrations under consistent datasets, tasks, and metrics.

8.2. Remote sensing, hyperspectral imaging, and foundation model-based GeoAI

This subsection discusses recent remote sensing and Earth-observation developments that are adjacent to the trajectory-focused scope of this survey. It focuses on masked autoencoding, self-supervised learning, hyperspectral and spectral foundation models, and multimodal Earth-observation foundation models, highlighting their relevance to reconstruction-based and transferable geospatial representation learning.

8.2.1. Masked autoencoding and self-supervised pretraining in remote sensing

A closely related direction in recent GeoAI research concerns masked autoencoding and self-supervised pretraining for remote sensing imagery. Although these methods are not trajectory-analysis approaches, they are relevant to this survey because they use reconstruction-based objectives to learn transferable geospatial representations from large unlabeled datasets. This is especially important in remote sensing, where labeled data are costly but optical, multispectral, SAR, and temporal Earth-observation imagery are continuously collected. Masked image modeling therefore extends the autoencoder principle from direct input reconstruction to large-scale self-supervised representation learning, where an encoder learns from partially observed imagery and a decoder reconstructs masked content during pretraining.

Several works adapt masked autoencoding to the specific structure of satellite imagery. SatMAE [194] extends the MAE framework [195] to temporal and multispectral satellite data through positional encodings and masking strategies that operate across temporal or spectral dimensions. This reflects the fact that remote sensing imagery often includes repeated observations and physically meaningful spectral bands rather than only RGB channels. RingMo [196] similarly develops a masked image modeling framework for large-scale optical remote sensing data, with a patch incomplete masking strategy designed for dense small-object scenes where fully masked patches may remove entire objects. These examples show that masked reconstruction objectives require domain-specific adaptation for geospatial imagery.

Other works modify the MAE framework by incorporating scale and feature-level reconstruction. Scale-MAE [197] addresses the multiscale nature of geospatial imagery by using ground sample distance information in the positional encoding and reconstructing low- and high-frequency image components through a multiscale decoder. This is relevant because identical pixel dimensions may correspond to very different ground areas depending on sensor resolution and coverage. FG-MAE [198] focuses instead on the reconstruction target, arguing that raw pixel reconstruction may overemphasize low-level details and noise, especially in SAR imagery. By reconstructing features such as histograms of oriented gradients and normalized difference indices, it incorporates spatial and spectral prior knowledge into the self-supervised objective.

Overall, these methods show that masked autoencoding has become an important self-supervised learning paradigm in remote sensing GeoAI. Their contribution lies not only in using encoder-decoder architectures, but in adapting reconstruction-based learning to geospatial image properties such as spectral structure, temporal repetition, sensor noise, object density, and spatial scale. Since their inputs and downstream tasks differ from the trajectory-focused methods reviewed in the main taxonomy, they are best understood as adjacent evidence that autoencoder-style reconstruction objectives remain relevant in modern GeoAI.

8.2.2. Hyperspectral and spectral foundation models

Another relevant direction concerns hyperspectral and spectral foundation models for remote sensing. These methods use reconstruction-based pretraining to learn transferable representations from high-dimensional spectral imagery. Unlike RGB images, multispectral and hyperspectral data contain ordered bands that encode material, vegetation, water, soil, and built-environment properties across the electromagnetic spectrum. This makes spectral remote sensing suitable for representation learning, but also introduces challenges related to dimensionality, redundancy, sensor-specific band configurations, noise, and limited labels.

Early masked autoencoding approaches for hyperspectral image classification illustrate the value of reconstruction-based objectives in this setting. MAEST [199] combines a masked reconstruction branch with a supervised classification branch, allowing the classifier to benefit from features learned by reconstructing corrupted hyperspectral inputs affected by atmospheric, thermal, quantization, and redundancy-related noise. Similarly, masked spatial-spectral transformer models [200] use spatial-spectral patch embeddings, spectral positional information, and masked reconstruction on unlabeled hyperspectral observations before supervised fine-tuning. These works show that reconstruction can act both as a pretext task and as a mechanism for more robust spectral-spatial feature extraction when labeled samples are scarce.

More recent spectral foundation models extend this idea to larger-scale pretraining. S2MAE [201] treats spectral remote sensing images as three-dimensional spatial-spectral tensors and applies masked autoencoding with compact cube tokens, high masking ratios, and progressive pretraining across large Sentinel-2 datasets. This exploits local spectral continuity and spatial invariance while avoiding the limitations of RGB-oriented pretraining. SpectralGPT [202] develops this direction further through 3D token generation, tensor masking, progressive pretraining, and multi-target reconstruction to preserve spatial-spectral coupling and sequential spectral information.

Hyperspectral foundation models further specialize this paradigm for HSI interpretation. HyperSIGMA [203], for example, uses large-scale hyperspectral pretraining and combines spatial and spectral subnetworks through a spectral enhancement module. It also introduces sparse sampling attention to reduce spatial and spectral redundancy and improve contextual feature extraction. Unlike models focused only on high-level tasks, HyperSIGMA is evaluated across classification, target detection, anomaly detection, change detection, unmixing, denoising, and super-resolution. This broader task coverage is important because hyperspectral analysis often requires both semantic understanding and signal-level reconstruction or restoration.

Overall, hyperspectral and spectral foundation models show that reconstruction-based learning remains relevant in modern GeoAI. Their importance lies in adapting autoencoder-style objectives to spectral data, where neighboring bands are ordered, correlated, and semantically meaningful. Although outside the trajectory-centered focus of this survey, they demonstrate that autoencoder-based representation learning continues to evolve in geospatial domains characterized by high dimensionality, limited labels, and sensor-specific structure.

8.2.3. Multimodal earth-observation foundation models

Beyond masked autoencoding and spectral foundation models, recent Earth-observation research has moved toward multimodal foundation models that integrate different sensors, temporal observations, and contextual information. SkySense [204], for example, is a large-scale multimodal remote sensing foundation model trained on optical and SAR temporal sequences. Its factorized spatio-temporal encoder separates spatial feature extraction from multimodal temporal fusion, while multi-granularity contrastive learning and geo-context prototype learning improve representation learning across modalities, spatial granularities, and regions. This reflects a broader GeoAI trend in which foundation models are expected to support heterogeneous downstream tasks such as classification, segmentation, object detection, and change detection.

DOFA [205] addresses multimodal Earth observation from the perspective of sensor flexibility. Instead of relying on fixed input channels or modality-specific encoders, it uses a wavelength-conditioned dynamic hypernetwork to generate input-dependent weights for different spectral configurations. This allows a single Transformer-based model to process optical, SAR, multispectral, and hyperspectral data.

Although these models are not primarily autoencoder architectures, they are relevant to this survey because they show how self-supervised and foundation model-based GeoAI is increasingly designed around transferable representations, multimodal fusion, and adaptation across heterogeneous geospatial data sources.

8.2.4. Relation to the trajectory-focused taxonomy of this survey

The remote sensing and spectral foundation models discussed above are related to autoencoder-based GeoAI at the level of representation learning, but they differ from the trajectory-focused methods reviewed in the main taxonomy. Masked autoencoding methods for satellite, multispectral, and hyperspectral imagery directly extend the autoencoder principle by learning latent representations through partial observation and reconstruction. In these cases, reconstruction is not only used for denoising or dimensionality reduction, but also as a self-supervised pretraining objective that enables the model to exploit large unlabeled geospatial datasets. Spectral and hyperspectral models further adapt this idea to data with ordered bands, spatial-spectral coupling, and sensor-specific characteristics, while multimodal Earth-observation foundation models generalize the representation-learning objective toward heterogeneous inputs, such as optical, SAR, multispectral, hyperspectral, and temporal data.

Nevertheless, these works are not included in the main taxonomy of analyzed papers because their inputs, tasks, datasets, and evaluation protocols differ from those of trajectory-based GeoAI. The core taxonomy of this survey focuses on autoencoder-based methods for trajectory representation learning and related tasks such as compression, clustering, anomaly detection, similarity computation, recovery, prediction, classification, and generation. In contrast, remote sensing foundation models typically address image or spectral-cube-based tasks such as scene classification, semantic segmentation, change detection, target detection, unmixing, denoising, and super-resolution. Therefore, they are best understood as an adjacent GeoAI direction that reinforces the broader relevance of reconstruction-based and self-supervised representation learning, while remaining outside the trajectory-centered scope of the reviewed corpus.

9. Open challenges and future directions

This section synthesizes the main open challenges and future research directions identified from the reviewed trajectory-focused corpus and from adjacent GeoAI developments discussed in Section 8. Section 9.1 first summarizes the remaining limitations affecting robustness, transferability, interpretability, evaluation, and deployment, while Section 9.2 outlines promising directions for developing more robust, uncertainty-aware, multimodal, explainable, and operational autoencoder-based GeoAI.

9.1. Remaining challenges

Despite the significant progress of autoencoder-based methods for trajectory representation learning, several challenges remain that limit their reliability, transferability, and practical deployment in real-world GeoAI applications. The reviewed literature already contains partial responses to some of these issues, including semi-supervised learning, denoising reconstruction, masked autoencoding, probabilistic latent modeling, context-aware fusion, physically constrained decoding, and efficient latent-space retrieval. However, these contributions remain fragmented across specific tasks, datasets, and application domains. Since the strict inclusion criteria of this survey focus primarily on autoencoder-based trajectory GeoAI, the following discussion distinguishes between challenges directly observed in the reviewed trajectory corpus and broader GeoAI robustness issues that become relevant when trajectory analysis is combined with adjacent geospatial modalities, such as remote sensing, Earth observation, environmental sensing, and multimodal spatial context. This distinction is important because these domains are not treated as separate application categories in the main taxonomy, but they remain closely connected to future autoencoder-based GeoAI through shared representation-learning, reconstruction, fusion, and robustness requirements.

9.1.1. Limited labeled data and weak supervision

A persistent challenge in GeoAI is the limited availability of high-quality labeled data. This issue is especially important for trajectory analysis, where labels such as transportation mode, maneuver type, abnormal behavior, route intent, or activity category are often expensive, noisy, incomplete, or unavailable. In anomaly detection, the problem is even more pronounced because abnormal events are rare, domain-specific, and difficult to define exhaustively in advance. Similar limitations also appear in adjacent GeoAI settings, such as remote sensing and environmental monitoring, where pixel-level, object-level, or event-level annotations often require expert interpretation and may not be consistently available across regions, sensors, or time periods.

Autoencoders are naturally relevant in this setting because they can exploit unlabeled data through reconstruction-based training. Several reviewed studies demonstrate this advantage in unsupervised or weakly supervised trajectory-analysis settings, including anomaly detection, trajectory clustering, recovery, and similarity computation. Semi-supervised convolutional autoencoder models have also been used for transportation mode identification under limited labeled data, while masked and contrastive reconstruction strategies further illustrate the growing role of self-supervised objectives. Nevertheless, limited supervision remains an open challenge because reconstruction alone does not guarantee that the learned latent space preserves the semantic factors required by downstream GeoAI tasks. A model may reconstruct trajectories accurately while failing to separate meaningful behaviors, detect rare events, or transfer to new regions. This highlights the gap between learning compact representations from unlabeled data and learning representations that are semantically reliable for downstream geospatial analysis.

9.1.2. Robustness under noise, degradation, and remote sensing variability

Robustness is one of the most urgent challenges for autoencoder-based GeoAI. Within the trajectory-focused corpus, robustness mainly concerns noisy coordinates, missing observations, irregular sampling, low sampling rates, map-matching errors, communication loss, and sensor-specific inconsistencies. AIS and ADS-B trajectories may contain positional inaccuracies, temporal gaps, duplicate records, abnormal reporting intervals, or inconsistent movement attributes. In urban mobility, pedestrian, maritime, and aviation settings, such imperfections can significantly alter the learned representation, especially when models assume that the training data reflect normal and reliable movement patterns.

A related set of robustness challenges appears in adjacent GeoAI domains that are not systematically covered by the main taxonomy of this review but are increasingly connected to trajectory analysis through multimodal modeling. In remote sensing and Earth-observation GeoAI, data are often affected by imaging degradation, atmospheric interference, cloud cover, shadows, occlusions, illumination differences, sensor noise, sensor-specific spectral response, spatial-resolution differences, and spectral or spatial variability during acquisition. These factors can cause the same geographic area to appear differently across sensors, seasons, viewing angles, resolutions, and acquisition conditions. As discussed in Section 8, masked autoencoding, self-supervised pretraining, spectral foundation models, and multimodal Earth-observation models provide relevant examples of how reconstruction-based learning is being extended to partially observed, corrupted, or heterogeneous geospatial data. Although these approaches extend beyond the strict trajectory corpus, they are important for understanding robustness requirements in broader autoencoder-based GeoAI.

Data variability can directly affect how autoencoder-based models learn, reconstruct, and score geospatial observations. If the training data contain systematic noise, missingness patterns, atmospheric artifacts, or sensor-specific distortions, the model may learn these effects as part of the normal data distribution. In anomaly detection, reconstruction error may then confuse true abnormal behavior with measurement noise, or fail to identify anomalies that resemble common sensor artifacts. In trajectory recovery, reconstructed paths may be numerically smooth but geographically invalid if road-network, motion, or physical constraints are not respected. In remote sensing and Earth-observation settings, an autoencoder may reconstruct visually plausible imagery while distorting spectral signatures that are essential for downstream tasks such as land-cover classification, change detection, or environmental monitoring. Therefore, reconstruction fidelity alone is not sufficient to establish robustness under controlled corruption, missing data, sensor shift, atmospheric variability, spectral/spatial perturbation, and cross-domain distribution shift.

9.1.3. Cross-region, cross-sensor, and cross-domain generalization

The limited generalization ability of existing methods is closely related to the robustness problem. Most autoencoder-based trajectory models are designed and evaluated on specific datasets, domains, or geographic contexts, such as urban taxi trajectories, pedestrian datasets, maritime AIS routes, aviation trajectories, or autonomous-driving benchmarks. These settings differ substantially in sampling frequency, movement constraints, spatial scale, interaction structure, sensor characteristics, and behavioral semantics. A model trained in one city, transport system, maritime region, or airport environment may therefore perform poorly when applied to another setting, even when the downstream task appears similar.

This limitation reflects a core GIScience issue: geospatial processes are spatially heterogeneous and context-dependent. The same movement pattern may have different meanings across regions and domains. For example, a route deviation may indicate abnormal behavior in a constrained maritime corridor, routine congestion avoidance in an urban road network, or controller intervention in aviation. Similarly, speed, stop duration, turning behavior, and trajectory density are not directly comparable across pedestrian, vehicle, vessel, and aircraft trajectories. In broader GeoAI settings, cross-domain generalization is further complicated by sensor differences, spatial resolution, seasonal variability, atmospheric effects, and differences in land-cover composition.

Existing autoencoder-based methods often learn representations that are effective within a specific dataset but are not explicitly shown to be region-invariant, sensor-invariant, or domain-transferable. This limits their broader applicability in operational GeoAI, where models may encounter cities, countries, sensors, platforms, or environmental conditions that differ from the original training data. The key challenge is determining whether learned latent spaces capture transferable geospatial structures or mainly encode dataset-specific regularities.

9.1.4. Multimodal fusion and representation alignment

Modern GeoAI applications increasingly require the integration of multiple data modalities. In trajectory-oriented GeoAI, movement records may need to be combined with road networks, semantic locations, land-use information, weather variables, environmental measurements, mobility context, remote sensing observations, and interactions with other agents. In adjacent Earth-observation GeoAI settings, multimodal learning may involve optical, SAR, multispectral, hyperspectral, temporal, elevation, and climate-related data. Although these modalities are not all systematically covered in the main trajectory taxonomy, they are increasingly relevant to autoencoder-based GeoAI because real-world geospatial systems often combine movement, imagery, graph, semantic, and environmental information.

Autoencoders provide a natural framework for multimodal representation learning because encoder-decoder architectures can compress heterogeneous inputs into shared or coordinated latent spaces. Several reviewed trajectory studies already move in this direction through context-aware prediction, map-informed recovery, multimodal convolutional fusion, graph-based interaction modeling, or attention-based integration of heterogeneous trajectory attributes. However, multimodal fusion remains difficult because different modalities have different spatial resolutions, temporal frequencies, noise properties, missing-data patterns, and semantic meanings. Simply concatenating features is often insufficient. For example, a trajectory sequence, a road-network graph, a satellite image, and a weather variable do not describe the same phenomenon at the same scale or with the same uncertainty.

This challenge becomes more pronounced under missing or degraded observations. In operational settings, some modalities may be unavailable, delayed, corrupted, or inconsistent with others. A multimodal autoencoder trained under complete and clean inputs may therefore behave unpredictably when deployed in realistic sensing environments. The underlying difficulty is not only combining heterogeneous data,

but also preserving modality-specific information while learning a coherent latent representation across trajectory, image, semantic, and environmental inputs.

9.1.5. Interpretability, uncertainty, and trustworthiness

The interpretability of latent representations remains a significant challenge for autoencoder-based GeoAI. Although many methods use latent embeddings for clustering, classification, anomaly detection, similarity computation, recovery, prediction, or generation, it is often unclear what specific spatial, temporal, semantic, or behavioral factors are encoded in the latent space. This lack of transparency limits the ability of researchers and practitioners to determine whether the model has learned meaningful geospatial structures or spurious dataset-specific correlations.

This issue is especially critical for anomaly detection and safety-related applications. A high reconstruction error may indicate an abnormal trajectory, but it may also result from sensor noise, missing data, rare but valid behavior, domain shift, or preprocessing errors. Without interpretability mechanisms, it is difficult to explain why a trajectory was flagged as anomalous, which part of the movement was responsible, and whether the result corresponds to an operationally meaningful event. Similar concerns arise in prediction and generation, where plausible-looking trajectories may still violate physical constraints, road-network structure, social interaction rules, or domain-specific safety requirements.

Uncertainty estimation is also underdeveloped in many autoencoder-based GeoAI methods. VAE and CVAE models provide a probabilistic latent-space formulation, and some reviewed studies use uncertainty-aware prediction or generative modeling. However, uncertainty is not consistently evaluated, calibrated, or connected to decision-making. In real-world GeoAI systems, uncertainty becomes especially important under noisy observations, sensor shifts, missing data, and unfamiliar geographic contexts. The broader challenge is that accurate reconstruction or prediction does not automatically imply that the model is interpretable, well-calibrated, or trustworthy under operational conditions.

9.1.6. Scalability, standardized evaluation, privacy, and operational deployment

Scalability and deployment remain major challenges for autoencoder-based GeoAI. Many reviewed methods are evaluated in offline experimental settings, often using controlled datasets, fixed preprocessing pipelines, and powerful computing resources. However, practical GeoAI systems must process large-scale and continuously generated data streams from vehicles, vessels, aircraft, mobile devices, IoT sensors, remote sensing platforms, and environmental monitoring systems. In such environments, models must satisfy constraints related to latency, memory usage, energy consumption, communication costs, and reliability under changing data distributions.

The lack of standardized benchmarks and evaluation protocols further limits progress. As shown throughout the reviewed literature, studies differ substantially in datasets, preprocessing strategies, sampling rates, train/test splits, task definitions, metrics, and reporting practices. This makes it difficult to determine whether performance improvements are due to model design, data processing, dataset properties, or evaluation protocols. The issue becomes even more serious for robustness and generalization, where models should be assessed under distribution shifts, noise, missing data, sensor variability, and cross-region transfer. Current benchmarks rarely provide systematic stress tests for these conditions.

Privacy and ethical data usage also remain unresolved. Trajectory data can reveal sensitive information about individuals, routines, locations, and behaviors. Generative autoencoder models may reduce direct exposure by producing synthetic trajectories, but they do not automatically eliminate privacy risks because generated samples may still preserve identifiable or inferable patterns. Similar concerns arise in multimodal GeoAI, where combining trajectories with imagery, semantic locations, or contextual data can increase re-identification risks. Therefore, operational deployment requires more than accurate models. It also requires careful attention to data governance, evaluation transparency, robustness monitoring, and model degradation after deployment.

Overall, these challenges show that the next stage of autoencoder-based GeoAI cannot be assessed solely through task-specific performance improvements. Robustness, uncertainty, interpretability, cross-domain transfer, multimodal alignment, privacy, scalability, and deployment form interconnected requirements for reliable geospatial representation learning.

9.2. Future research directions

The challenges outlined above suggest that future research on autoencoder-based GeoAI should move beyond isolated improvements in reconstruction accuracy or task-specific performance. The next stage of development should focus on models that are robust under realistic geospatial variability, transferable across regions and sensors, interpretable enough for domain use, and efficient enough for operational deployment. Since the main taxonomy of this survey focuses on trajectory-oriented GeoAI, the directions discussed below are centered on trajectory representation learning, while also considering adjacent GeoAI developments, such as remote sensing, Earth observation, foundation models, and multimodal geospatial learning, where they provide relevant methodological insights.

9.2.1. Robust and uncertainty-aware autoencoders

A central future direction is the development of autoencoder-based models that are explicitly robust to noisy, incomplete or degraded geospatial data. In trajectory GeoAI, this includes robustness to GPS noise, irregular sampling, missing points, low-frequency observations, map-matching errors, and sensor-specific inconsistencies in sources such as AIS and ADS-B. In such settings, reconstruction-based models should not only minimize average reconstruction error, but also distinguish between meaningful movement deviations and uncertainty caused by incomplete observations.

Uncertainty-aware autoencoders are especially important for anomaly detection, trajectory recovery, prediction, and generation. In anomaly detection, uncertainty estimates can help distinguish truly abnormal behavior from uncertain reconstructions caused by noise or sparse observations. In trajectory recovery, uncertainty can indicate when reconstructed paths are underdetermined or when multiple route alternatives are plausible. In prediction and generation, probabilistic latent representations can better represent the fact that future movement is often multimodal rather than deterministic. Future work should therefore give more attention to uncertainty calibration, confidence estimation, and the relationship between reconstruction error, latent-space density, and decision reliability.

This direction is also relevant to safety-critical GeoAI domains such as maritime monitoring, aviation safety, autonomous driving, and urban traffic management. In these settings, a model that produces an accurate-looking trajectory or anomaly score may still be insufficient if it cannot express when its output is unreliable. Robust and uncertainty-aware autoencoders should therefore be evaluated not only under clean test data, but also under realistic perturbations, missingness, sensor shifts, and rare-event conditions.

9.2.2. Degradation-aware autoencoders for remote sensing and earth-observation GeoAI

Even though remote sensing and Earth-observation models are not part of the main trajectory-focus of this survey, they are closely related to the broader GeoAI robustness problem. Future autoencoder-based GeoAI can benefit from this adjacent literature because remote sensing data commonly involve degradation, atmospheric interference, cloud

cover, shadows, occlusions, sensor differences, spatial-resolution variation, and spectral or spatial variability during acquisition. These issues create a useful parallel to trajectory-data degradation, where missing observations, noisy measurements, and sensor inconsistencies also affect representation learning.

A promising direction is the development of degradation-aware reconstruction objectives that account for the specific structure of geospatial data. In remote sensing, this may involve learning representations that preserve spectral signatures, spatial texture, temporal consistency, and sensor-specific characteristics rather than only producing visually plausible reconstructions. In trajectory GeoAI, analogous objectives may need to preserve movement continuity, road-network feasibility, speed and heading consistency, and domain-specific constraints. Across both settings, robustness depends on whether the latent representation preserves the information that is meaningful for downstream geospatial analysis, not only on whether the input can be reconstructed with low numerical error.

Masked autoencoding and self-supervised pretraining are particularly relevant in this context. In Earth-observation GeoAI, masked reconstruction has become an important strategy for exploiting large unlabeled datasets and learning transferable representations from partially observed or corrupted inputs. Similar principles could inform trajectory GeoAI, especially for sparse trajectories, partially observed movement sequences, missing map context, or degraded multimodal observations. However, such transfer of ideas should be treated carefully because satellite imagery, hyperspectral data, and trajectory sequences differ in structure, scale, and evaluation requirements.

9.2.3. Physics-guided, GIScience-guided, and foundation model-enhanced GeoAI

Future autoencoder-based GeoAI should more explicitly incorporate physical, geographic, and domain-specific constraints. Many trajectory models learn from observed data alone, but geospatial movement is not unconstrained. Vehicles follow road networks, vessels are affected by navigational corridors and maritime rules, aircraft follow airspace procedures and physical motion constraints, and pedestrians move within social and built-environment contexts. Similarly, Earth-observation data are shaped by sensor physics, atmospheric effects, spatial resolution, spectral response, and acquisition geometry.

Physics-guided and GIScience-guided autoencoders can help ensure that learned representations remain consistent with geographic reality. For trajectory analysis, this may involve incorporating road-network topology, map constraints, movement feasibility, spatial dependence, spatial heterogeneity, and scale effects into the representation-learning process. For broader GeoAI, it may involve encoding knowledge about sensor properties, spatial resolution, spectral structure, or environmental processes. Such constraints are particularly relevant for recovery, anomaly detection, prediction, and generation, where outputs may otherwise be numerically plausible but geographically invalid.

Foundation model-enhanced GeoAI provides another promising direction. As discussed in Section 8, recent foundation models in remote sensing and Earth observation show how large-scale self-supervised pretraining can support transferable representations across heterogeneous geospatial data. For trajectory GeoAI, foundation models may serve as pretrained encoders for spatial context, map information, semantic environments, or multimodal observations. Autoencoders may then function as task-specific reconstruction modules, latent-space adapters, compression layers, or uncertainty-aware refinement components. This type of integration could preserve the efficiency and structure of autoencoder representations while benefiting from the broader contextual knowledge captured by large pretrained models.

9.2.4. Multimodal geospatial representation learning and cross-domain transferability

Another important direction is the development of multimodal autoencoder frameworks that can jointly represent trajectories, maps, remote sensing observations, semantic context, weather variables, environmental measurements, and interaction information. Real-world GeoAI systems rarely rely on a single clean data source. Instead, they combine heterogeneous information with different spatial resolutions, temporal frequencies, uncertainty levels, and missing-data patterns. Autoencoder-based models are well suited to this setting because they can learn shared or coordinated latent spaces, but their effectiveness depends on how well they preserve modality-specific structure while aligning information across modalities.

Future work should pay particular attention to cross-domain transferability. A trajectory representation learned in one city, maritime region, road network, or sensor setting should ideally remain useful when transferred to another region or domain. This requires models that do not simply memorize dataset-specific sampling patterns, local movement habits, or sensor artifacts. In multimodal settings, this challenge becomes even more complex because each modality may shift differently across regions, seasons, sensors, and operational contexts.

Cross-domain transfer should therefore be evaluated as a central capability rather than as an optional experiment. Future studies should consider cross-city, cross-region, cross-sensor, and cross-task evaluation settings, especially for tasks such as anomaly detection, recovery, prediction, and classification. Such evaluation would help clarify whether learned autoencoder representations capture transferable geospatial structure or primarily encode the specific conditions of the training dataset.

9.2.5. Explainable and semantically meaningful latent spaces

Improving the interpretability of latent representations is essential for trustworthy autoencoder-based GeoAI. Many reviewed studies rely on latent spaces for downstream analysis, but the meaning of individual latent dimensions, latent clusters, reconstruction errors, or generated samples often remains unclear. Future work should therefore focus on latent representations that can be related to meaningful geospatial factors, such as route deviation, speed profile, heading change, stop behavior, interaction pattern, maneuver type, road-network context, land-use context, or environmental conditions.

Explainability is particularly important in anomaly detection. A reconstruction error alone does not explain whether a detected anomaly corresponds to an unsafe maneuver, a sensor artifact, an unusual but valid route, a rare weather-related behavior, or a preprocessing error. Similarly, in prediction and generation, plausible outputs should be explainable in terms of motion dynamics, map context, interaction structure, or uncertainty. For remote sensing and Earth-observation GeoAI, interpretability may require connecting latent representations to spectral signatures, spatial structures, land-cover properties, or changes over time.

Hybrid approaches involving autoencoders and LLM-based systems may provide one route toward semantic explanation, provided that explanations remain grounded in the underlying spatial and geometric evidence. Autoencoder representations can provide compact and structured encodings of movement or geospatial observations, while language-based models may support semantic labeling, task guidance, natural-language querying, or user-facing explanations. However, such systems must avoid producing explanations that are linguistically plausible but geographically unsupported. Explainable GeoAI therefore requires both semantic interpretability and spatial grounding.

9.2.6. Hybrid generative and representation-learning frameworks

Section 8 highlighted that emerging trajectory modeling paradigms, including diffusion models, flow matching, and LLM-based methods, should be viewed as complementary to autoencoder-based representation learning rather than as direct replacements. This complementarity suggests an important future direction: hybrid frameworks that combine compact autoencoder latent spaces with stronger generative, semantic, or planning-oriented models.

Autoencoders remain useful for learning efficient trajectory encodings, reconstructing incomplete observations, supporting latent-space similarity, detecting anomalies, and compressing complex geospatial inputs. Diffusion and flow-matching models, in contrast, are more directly suited to multimodal generation and probabilistic forecasting. LLM-based methods contribute semantic reasoning, activity interpretation, task guidance, and explanation. Future hybrid systems could therefore use autoencoder-derived latent representations as structured conditioning signals for diffusion or flow-matching models, while language-based components provide semantic constraints, controllable generation instructions, or explanation layers.

This direction is particularly relevant for tasks where both geometric precision and semantic understanding are required. For example, trajectory generation may need to produce paths that are not only statistically realistic but also consistent with map constraints, activity intent, traffic rules, or interaction context. Similarly, anomaly detection may benefit from combining reconstruction-based deviation scores with semantic descriptions of why a movement pattern is unusual. The main opportunity is to combine the compactness and efficiency of autoencoder representations with the expressiveness of newer generative and reasoning-based GeoAI paradigms.

9.2.7. Large-scale operational deployment and continual monitoring

For autoencoder-based GeoAI to move from research prototypes to operational systems, future work must address deployment constraints more directly. Many models are evaluated offline, but real-world systems often require streaming inference, low latency, limited memory usage, and robustness to changing data distributions. This is especially important for intelligent transportation systems, maritime surveillance, aviation monitoring, autonomous driving, mobile sensing, and large-scale Earth-observation pipelines.

Deployment-oriented research should consider lightweight architectures, efficient latent representations, model compression, pruning, quantization, edge and fog computing, cloud-edge coordination, and federated or privacy-aware learning. These directions are important because many geospatial data sources are distributed, sensitive, and continuously updated. A model deployed in a real environment must not only perform well at training time, but also remain reliable as sensors change, movement patterns evolve, infrastructure is modified, or environmental conditions shift.

Continual monitoring is therefore an important part of operational GeoAI. Autoencoder-based systems should be assessed for model drift, distribution shift, reconstruction degradation, false-alarm changes, and failure under rare or previously unseen conditions. This is closely connected to standardized evaluation: robust deployment requires benchmarks and reporting protocols that measure not only task accuracy, but also latency, memory use, privacy risk, robustness, calibration, and degradation over time.

Overall, future work on autoencoder-based GeoAI should aim toward models that are not only accurate within individual benchmarks, but also robust, transferable, interpretable, uncertainty-aware, multimodal, and deployable under realistic geospatial conditions. This requires treating representation learning as part of a broader GeoAI system, where data quality, geographic context, sensor variability, domain knowledge, privacy, and operational constraints jointly shape model reliability.

10. Conclusions

Trajectory data have become a central modality in GeoAI, enabling the modeling of complex spatio-temporal dynamics across diverse real-world applications. In this survey, we systematically reviewed autoencoder-based approaches for trajectory representation learning, covering a broad range of domains, including urban mobility, maritime navigation, aviation systems, interaction-aware environments, and synthetic validation settings. The analysis shows that autoencoder-based architectures, spanning feed-forward, recurrent, convolutional, variational, transformer-based, graph-based, and hybrid designs, provide a flexible framework for learning compact and expressive latent representations. These representations support a wide range of tasks, including compression, clustering, anomaly detection, similarity computation, recovery, classification, prediction, and trajectory generation, particularly in settings where labeled data are limited or costly to obtain.

At the same time, the reviewed literature shows that several important challenges remain unresolved. These include robustness under noisy, incomplete, degraded, or heterogeneous geospatial observations; limited cross-region, cross-sensor, and cross-domain generalization; difficulty in multimodal representation alignment; privacy and ethical concerns; computational and deployment constraints; lack of standardized evaluation protocols; and limited interpretability of learned latent spaces. These challenges become increasingly important as trajectory analysis is integrated with broader GeoAI modalities, such as road-network context, environmental observations, remote sensing data, and other sensor-derived spatial information.

Future research should therefore move beyond task-specific performance improvements and focus on robust, uncertainty-aware, transferable, interpretable, and deployment-ready autoencoder-based GeoAI. Promising directions include robust and degradation-aware reconstruction, multimodal and cross-domain representation learning, physics-guided and GIScience-guided modeling, foundation model-enhanced GeoAI, semantically meaningful latent spaces, privacy-aware learning, and scalable operational deployment. Overall, autoencoder-based trajectory representation learning remains a rapidly evolving research area with strong potential to support more robust, scalable, and trustworthy spatio-temporal intelligence in modern GeoAI systems.

CRediT authorship contribution statement

Andreas Karathanasis: Writing – original draft, Visualization, Resources, Investigation. **John Violos:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization. **Iraklis Varlamis:** Project administration, Funding acquisition, Conceptualization. **Aris Leivadeas:** Writing – review & editing, Supervision, Project administration, Formal analysis. **Konstantinos Tserpes:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is part of the MUlti-Sensor Inferred Trajectories (MUSIT) project that has received funding from the European Union's HORIZON-MSCA-2023-SE-01 research and innovation programme under grant agreement No. 101182585. The authors would also like to thank the MPhil program in Computer Science and Informatics of Harokopio University of Athens https://mphil.dit.hua.gr/en/home/ for supporting this work.

Data availability

No data was used for the research described in the article.

References

[1] D. Wang, T. Miwa, T. Morikawa, Big trajectory data mining: a survey of methods, applications, and services, Sensors 20 (16) (2020) 4571.
[2] T. Alsahfi, M. Almotairi, R. Elmasri, A survey on trajectory data warehouse, Spat. Inf. Res. 28 (1) (2020) 53–66.
[3] C. Wang, J. Huang, Y. Wang, Z. Lin, X. Jin, X. Jin, D. Weng, Y. Wu, A deep spatiotemporal trajectory representation learning framework for clustering, IEEE Trans. Intell. Transp. Syst. 25 (7) (2024) 7687–7700.
[4] W. Chen, Y. Liang, Y. Zhu, Y. Chang, K. Luo, H. Wen, L. Li, Y. Yu, Q. Wen, C. Chen, et al., Deep learning for trajectory data management and mining: a survey and beyond, arXiv e-prints arXiv:2403.00231, 2024.

[5] W. Li, GeoAI: where machine learning and big data converge in GIScience, J. Spat. Inf. Sci. (20) (2020) 71–77.

[6] G. Mai, Y. Xie, X. Jia, N. Lao, J. Rao, Q. Zhu, Z. Liu, Y.-Y. Chiang, J. Jiao, Towards the next generation of geospatial artificial intelligence, Int. J. Appl. Earth Obs. Geoinf. 136 (2025) 104368.

[7] D. Hu, L. Chen, H. Fang, Z. Fang, T. Li, Y. Gao, Spatio-temporal trajectory similarity measures: a comprehensive survey and quantitative study, IEEE Trans. Knowl. Data Eng. 36 (5) (2023) 2191–2212.

[8] A. Graser, A. Jalali, J. Lampert, A. Weißenfeld, K. Janowicz, MobilityDL: a review of deep learning from trajectory data, Geoinformatica 29 (1) (2025) 115–147.

[9] W. Sheng, X. Li, Siamese denoising autoencoders for joints trajectories reconstruction and robust gait recognition, Neurocomputing 395 (2020) 86–94.

[10] J. Cho, Y. Kang, Context-aware anomaly detection of pedestrian trajectories in urban back streets using a variational autoencoder, ISPRS Int. J. Geo-Inf. 14 (11) (2025) 438.

[11] D. Yao, C. Zhang, Z. Zhu, J. Huang, J. Bi, Trajectory clustering via deep representation learning, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3880–3887.

[12] D. Yao, C. Zhang, Z. Zhu, Q. Hu, Z. Wang, J. Huang, J. Bi, Learning deep representation for trajectory clustering, Expert Syst. 35 (2) (2018) e12252.

[13] X. Li, K. Zhao, G. Cong, C.S. Jensen, W. Wei, Deep representation learning for trajectory similarity computation, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), IEEE, 2018, pp. 617–628.

[14] M. Kölle, S. Illium, C. Hahn, L. Schauer, J. Hutter, C. Linnhoff-Popien, Compression of GPS trajectories using autoencoders, arXiv:2301.07420, Jan 2023, [Online]. Available.

[15] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, J. Wang, Self-supervised trajectory representation learning with temporal regularities and travel semantics, in: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, pp. 843–855.

[16] M. Liatsikou, S. Papadopoulos, L. Apostolidis, I. Kompatsiaris, A denoising hybrid model for anomaly detection in trajectory sequences, in: EDBT/ICDT Workshops, 2021.

[17] Y. Ye, A. Wang, A. Zeb, S. Zhang, J.J. Yu, Map-informed trajectory recovery with adaptive spatio-temporal autoencoder, IEEE Trans. Intell. Transp. Syst. 26 (1) (2024) 102–115.

[18] Z. Wang, S. Zhang, J. James, Reconstruction of missing trajectory data: a deep learning approach, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2020, pp. 1–6.

[19] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, H. Chen, A survey on trajectory-prediction methods for autonomous driving, IEEE Trans. Intell. Veh. 7 (3) (2022) 652–674.

[20] X. Song, K. Chen, X. Li, J. Sun, B. Hou, Y. Cui, B. Zhang, G. Xiong, Z. Wang, Pedestrian trajectory prediction based on deep convolutional LSTM network, IEEE Trans. Intell. Transp. Syst. 22 (6) (2020) 3285–3302.

[21] S. Lu, Y. Xia, Dual supervised autoencoder based trajectory classification using enhanced spatio-temporal information, IEEE Access 8 (2020) 173918–173932.

[22] X. Chen, J. Xu, R. Zhou, W. Chen, J. Fang, C. Liu, Trajvae: a variational autoencoder model for trajectory generation, Neurocomputing 428 (2021) 332–339.

[23] X. Shi, D.-Y. Yeung, Machine learning for spatiotemporal sequence forecasting: a survey, arXiv preprint arXiv:1808.06865, 2018.

[24] S. Chakri, N. Mouhni, F. Ennaama, Exploring the frontiers of trajectory outlier detection: an in-depth review and comparative analysis, Int. J. Electr. Comput. Eng. (2088-8708) 14 (5) (2024).

[25] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P.S. Yu, L. He, Deep clustering: a comprehensive survey, IEEE Trans. Neural Netw. Learn. Syst. 36 (4) (2024) 5858–5878.

[26] A. Makris, I. Kontopoulos, P. Alimisis, K. Tserpes, A comparison of trajectory compression algorithms over AIS data, IEEE Access 9 (2021) 92516–92530.

[27] X. Chu, X. Tan, W. Zeng, A clustering ensemble method of aircraft trajectory based on the similarity matrix, Aerospace 9 (5) (2022) 269.

[28] S. Song, X. Zhao, Z. Zhang, M. Luo, A data compression method for wellbore stability monitoring based on deep autoencoder, Sensors 24 (12) (2024) 4006.

[29] W. Zeng, Z. Xu, Z. Cai, X. Chu, X. Lu, Aircraft trajectory clustering in terminal airspace based on deep autoencoder and Gaussian mixture model, Aerospace 8 (9) (2021) 266.

[30] Z. Wang, G. Yuan, H. Pei, Y. Zhang, X. Liu, Unsupervised learning trajectory anomaly detection algorithm based on deep representation, Int. J. Distrib. Sens. Netw. 16 (12) (2020) 1550147720971504.

[31] W. Zhang, M. Hu, J. Du, An end-to-end framework for flight trajectory data analysis based on deep autoencoder network, Aerosp. Sci. Technol. 127 (2022) 107726.

[32] X. Olive, L. Basora, Identifying anomalies in past en-route trajectories with clustering and anomaly detection methods, in: ATM Seminar 2019, 2019.

[33] X. Olive, J. Grignard, T. Dubot, J. Saint-Lot, Detecting controllers' actions in past mode S data by autoencoder-based anomaly detection, in: SID 2018, 8th SESAR Innovation Days, 2018.

[34] B. Peralta, R. Soria, O. Nicolis, F. Ruggeri, L. Caro, A. Bronfman, Outlier vehicle trajectory detection using deep autoencoders in Santiago, Chile, Sensors 23 (3) (2023) 1440.

[35] M. Yue, Y. Li, H. Yang, R. Ahuja, Y.-Y. Chiang, C. Shahabi, Detect: deep trajectory clustering for mobility-behavior analysis, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 988–997.

[36] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 791–800, https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[37] Y. Zheng, Q. Li, Y. Chen, X. Xie, W.-Y. Ma, Understanding mobility based on GPS data, in: Proceedings of the 10th International Conference on Ubiquitous Computing, 2008, pp. 312–321, https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[38] Y. Zheng, X. Xie, W.-Y. Ma, et al., GeoLife: a collaborative social networking service among user, location and trajectory, IEEE Data Eng. Bull. 33 (2) (2010) 32–39, https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[39] W. Yu, Q. Huang, A deep encoder-decoder network for anomaly detection in driving trajectory behavior under spatio-temporal context, Int. J. Appl. Earth Obs. Geoinf. 115 (2022) 103115.

[40] B. Murray, L.P. Perera, A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data, Ocean Eng. 209 (2020) 107478.

[41] Y. Xu, H. Zhang, Z. Gao, C. Liu, Z. Zhao, CMAE-traj: a contrastive masked autoencoder framework for trajectory prediction, Robot. Auton. Syst. 199 (2026) 105391.

[42] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, et al., Interaction dataset: an international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps, arXiv preprint arXiv:1910.03088, 2019, https://interaction-dataset.com/.

[43] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J.K. Pontes, D. Ramanan, P. Carr, J. Hays, Argoverse 2: next generation datasets for self-driving perception and forecasting, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 2021, https://www.argoverse.org/av2.html.

[44] J. Lambert, J. Hays, Trust, but verify: cross-modality fusion for HD map change detection, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 2021, https://www.argoverse.org/av2.html.

[45] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C.R. Qi, Y. Zhou, et al., Large scale interactive motion forecasting for autonomous driving: the waymo open motion dataset, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9710–9719, https://waymo.com/open/.

[46] K. Gajamannage, Y. Park, R. Paffenroth, A.P. Jayasumana, Reconstruction of fragmented trajectories of collective motion using hadamard deep autoencoders, Pattern Recognit. 131 (2022) 108891.

[47] T. Wang, C. Ye, H. Zhou, M. Ou, B. Cheng, AIS ship trajectory clustering based on convolutional auto-encoder, in: Proceedings of SAI Intelligent Systems Conference, Springer, 2020, pp. 529–546.

[48] M. Liang, R.W. Liu, S. Li, Z. Xiao, X. Liu, F. Lu, An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation, Ocean Eng. 225 (2021) 108803.

[49] X. Olive, L. Basora, B. Viry, R. Alligier, Deep trajectory clustering with autoencoders, in: ICRAT 2020, 9th International Conference for Research in Air Transportation, 2020.

[50] O. Rákos, S. Aradi, T. Bécsi, Z. Szalay, Compression of vehicle trajectories with a variational autoencoder, Appl. Sci. 10 (19) (2020) 6739.

[51] U.S. Department of Transportation Federal Highway Administration, Next generation simulation (NGSIM) vehicle trajectories and supporting data, Provided by ITS DataHub, 2016, https://doi.org/10.21949/1504477, (Accessed: from 30 March 2026).

[52] L. Li, X. Li, Y. Yang, J. Dong, Indoor tracking trajectory data similarity analysis with a deep convolutional autoencoder, Sustain. Cities Soc. 45 (2019) 588–595.

[53] Y. Qi, J. Yang, D. Xu, R. Shao, Y. Duan, Y. Wang, Vessel trajectory anomaly detection based on multi-scale convolutional autoencoder, Ocean Eng. 343 (2026) 123564.

[54] M. Memarzadeh, B. Matthews, I. Avrekh, Unsupervised anomaly detection in flight data using convolutional variational auto-encoder, Aerospace 7 (8) (2020) 115.

[55] H. Gjoreski, M. Ciliberto, L. Wang, F.J.O. Morales, S. Mekki, S. Valentin, D. Roggen, The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices, IEEE Access 6 (2018) 42592–42604, http://www.shl-dataset.org/.

[56] S. Dabiri, C.-T. Lu, K. Heaslip, C.K. Reddy, Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data, IEEE Trans. Knowl. Data Eng. 32 (5) (2019) 1010–1023.

[57] O. Rakos, T. Becsi, S. Aradi, Adversarial autoencoder for trajectory generation and maneuver classification, in: 2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES), IEEE, 2021, pp. 013–018.

[58] T. Krauth, A. Lafage, J. Morio, X. Olive, M. Waltert, Deep generative modelling of aircraft trajectories in terminal maneuvering areas, Mach. Learn. Appl. 11 (2023) 100446.

[59] H. Cheng, W. Liao, X. Tang, M.Y. Yang, M. Sester, B. Rosenhahn, Exploring dynamic context for multi-path trajectory prediction, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 12795–12801.

[60] P. Kothari, S. Kreiss, A. Alahi, Human trajectory forecasting in crowds: a deep learning perspective, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 7386–7400, https://www.aicrowd.com/challenges/trajnet-a-trajectory-forecasting-challenge.

[61] X. Liu, Y. Wang, K. Jiang, Z. Zhou, K. Nam, C. Yin, Interactive trajectory prediction using a driving risk map-integrated deep learning method for surrounding vehicles on highways, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 19076–19087.

[62] R. Krajewski, J. Bock, L. Kloeker, L. Eckstein, The highd dataset: a drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2118–2125, https://levelxdata.com/highd-dataset/.

[63] S. Li, M. Liang, R.W. Liu, Vessel trajectory similarity measure based on deep convolutional autoencoder, in: 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), IEEE, 2020, pp. 333–338.

[64] X. Wang, Z. Liu, Y. Chen, M. Zhang, W. Mao, VT-EncNet: a multi-modal convolutional autoencoder with attention for vessel trajectory representation and similarity computation, Transp. Res. E Logist. Transp. Rev. 211 (2026) 104857.

[65] Y. Liu, Z. Shi, B. Fu, J. Ke, H. Xu, X. Wang, A deep ship trajectory clustering method based on feature embedded representation learning, J. Mar. Sci. Eng. 14 (1) (2026).

[66] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, Y. Huang, T-drive: driving directions based on taxi trajectories, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010, pp. 99–108, https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

[67] S. Horovitz, G.Y. Cohen, D. Shmirer, S. Boxer, I. Blumenkrantz, M. Lasry, SaveDat: spatio-temporal trajectory compression by LSTM, in: 2022 IEEE 7th International Conference on Intelligent Transportation Engineering (ICITE), IEEE, 2022, pp. 442–450.

[68] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, Predicting taxi–passenger demand using streaming data, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1393–1402, https://archive.ics.uci.edu/dataset/339/taxi+service+trajectory+prediction+challenge+ecml+pkdd+2015.

[69] Y. Suo, W. Chen, C. Claramunt, S. Yang, A ship trajectory prediction framework based on a recurrent neural network, Sensors 20 (18) (2020) 5133.

[70] Y. Dong, L. Wang, S. Zhou, W. Tang, G. Hua, C. Sun, Afc-rnn: adaptive forgetting-controlled recurrent neural network for pedestrian trajectory prediction, IEEE Trans. Pattern Anal. Mach. Intell. 47 (11) (2025) 10177–10191.

[71] S. Pellegrini, A. Ess, K. Schindler, L. Van Gool, You'll never walk alone: modeling social behavior for multi-target tracking, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 261–268, https://vision.ee.ethz.ch/datasets_extra/ewap_dataset/. https://gitlab.tu-clausthal.de/pka20/Trajectory-Prediction-Pedestrian/-/tree/31e2b8cd75ab0e68e9b6cf2470bb885eb82640d2/datasets/ETH.

[72] A. Lerner, Y. Chrysanthou, D. Lischinski, Crowds by example, Comput. Graph. Forum 26 (3) (2007) 655–664. Wiley Online Library, https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data. https://gitlab.tu-clausthal.de/pka20/Trajectory-Prediction-Pedestrian/-/tree/31e2b8cd75ab0e68e9b6cf2470bb885eb82640d2/datasets/UCY.

[73] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning social etiquette: human trajectory understanding in crowded scenes, in: European Conference on Computer Vision, Springer, 2016, pp. 549–565, https://cvgl.stanford.edu/projects/uav_data/.

[74] I. Alhussein, A.H. Ali, Discovering anomalous patterns in flight data at Najaf airport using LSTM auto encoders, in: 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA), IEEE, 2021, pp. 106–110.

[75] R. Quan, L. Zhu, Y. Wu, Y. Yang, Holistic LSTM for pedestrian trajectory prediction, IEEE Trans. Image Process. 30 (2021) 3229–3239.

[76] A. Rasouli, I. Kotseruba, J.K. Tsotsos, Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 206–213, https://data.nvision2.eecs.yorku.ca/JAAD_dataset/.

[77] A. Rasouli, I. Kotseruba, J.K. Tsotsos, Agreeing to cross: how drivers and pedestrians communicate, in: 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 264–269, https://data.nvision2.eecs.yorku.ca/JAAD_dataset/.

[78] A. Rasouli, I. Kotseruba, T. Kunic, J.K. Tsotsos, Pie: a large-scale dataset and models for pedestrian intention estimation and trajectory prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6262–6271, https://data.nvision2.eecs.yorku.ca/PIE_dataset/.

[79] N. Bhujel, E.K. Teoh, W.-Y. Yau, Pedestrian trajectory prediction using RNN encoder-decoder with spatio-temporal attentions, in: 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR), IEEE, 2019, pp. 110–114.

[80] H. Xue, D.Q. Huynh, M. Reynolds, PoPPL: pedestrian trajectory prediction by LSTM with automatic route class clustering, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 77–90.

[81] B. Majecka, Statistical Models of Pedestrian Behaviour in the Forum (Ph.D. dissertation), Master's Thesis, School of Informatics, University of Edinburgh, 2009, https://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/.

[82] M.Á. De Miguel, J.M. Armingol, F. Garcia, Vehicles trajectory prediction using recurrent VAE network, IEEE Access 10 (2022) 32742–32749.

[83] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, Y. Kato, A variational autoencoder based generative model of urban human mobility, in: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2019, pp. 425–430.

[84] Y. Liu, K. Zhao, G. Cong, Z. Bao, Online anomalous trajectory detection with deep generative sequence modeling, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, 2020, pp. 949–960.

[85] A. Demetriou, H. Alfsvåg, S. Rahrovani, M. Haghir Chehreghani, A deep learning framework for generation and analysis of driving scenario trajectories, SN Comput. Sci. 4 (3) (2023) 251.

[86] C. Ma, Z. Miao, M. Li, S. Song, M.-H. Yang, Detecting anomalous trajectories via recurrent neural networks, in: Asian Conference on Computer Vision, Springer, 2018, pp. 370–382.

[87] S. Capobianco, N. Forti, L.M. Millefiori, P. Braca, P. Willett, Recurrent encoder–decoder networks for vessel trajectory prediction with uncertainty estimation, IEEE Trans. Aerosp. Electron. Syst. 59 (3) (2022) 2554–2565.

[88] H. Jiang, L. Chang, Q. Li, D. Chen, Trajectory prediction of vehicles based on deep learning, in: 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), IEEE, 2019, pp. 190–195.

[89] C. Wang, F. Lyu, S. Wu, Y. Wang, L. Xu, F. Zhang, S. Wang, Y. Wang, Z. Du, A deep trajectory clustering method based on sequence-to-sequence autoencoder model, Trans. GIS 26 (4) (2022) 1801–1820.

[90] Y. Zhang, A. Liu, G. Liu, Z. Li, Q. Li, Deep representation learning of activity trajectory similarity computation, in: 2019 IEEE International Conference on Web Services (ICWS), IEEE, 2019, pp. 312–319.

[91] X. Feng, Z. Cen, J. Hu, Y. Zhang, Vehicle trajectory prediction using intention-based conditional variational autoencoder, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 3514–3519.

[92] H. Ren, S. Ruan, Y. Li, J. Bao, C. Meng, R. Li, Y. Zheng, Mtrajrec: map-constrained trajectory recovery via SeQ2SeQ multi-task learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1410–1419.

[93] Q. Chen, M. Le, Trajectory segmentation and clustering in terminal airspace using Transformer–VAE and density-aware optimization, Aerospace 12 (11) (2025) 969.

[94] J. Hou, H. Zhou, M. Grifoll, Y. Zhou, J. Liu, Y. Ye, P. Zheng, A Transformer–VAE approach for detecting ship trajectory anomalies in cross-sea bridge areas, J. Mar. Sci. Eng. 13 (5) (2025) 849.

[95] H. Chen, J. Wang, K. Shao, F. Liu, J. Hao, C. Guan, G. Chen, P.-A. Heng, Traj-mae: masked autoencoders for trajectory prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 8351–8362.

[96] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al., Argoverse: 3D tracking and forecasting with rich maps, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8748–8757, https://www.argoverse.org/av1.html.

[97] X. Wu, R. Zhang, L. Li, Clustering trajectories via sparse auto-encoders, in: 2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2021, pp. 260–266.

[98] L. Shi, L. Wang, S. Zhou, G. Hua, Trajectory unified transformer for pedestrian trajectory prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9675–9684.

[99] Y. Chen, H. Zhang, W. Sun, B. Zheng, Rntrajrec: road network enhanced trajectory recovery with spatial-temporal transformer, in: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, pp. 829–842.

[100] M. Yue, Y.-Y. Chiang, C. Shahabi, Vambc: a variational approach for mobility behavior clustering, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 453–469.

[101] L. Zhang, W. Lu, F. Xue, Y. Chang, A trajectory outlier detection method based on variational auto-encoder, Math. Biosci. Eng. 20 (8) (2023) 15075–15093.

[102] F. Lu, Z. Zhang, C. Shui, Online trajectory anomaly detection model based on graph neural networks and variational autoencoder, in: Journal of Physics: Conference Series, vol. 2816, no. 1. IOP Publishing, 2024, p. 012006.

[103] M. Neumeier, M. Botsch, A. Tollkühn, T. Berberich, Variational autoencoder-based vehicle trajectory prediction with an interpretable latent space, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 820–827.

[104] L. Xie, T. Guo, J. Chang, C. Wan, X. Hu, Y. Yang, C. Ou, A novel model for ship trajectory anomaly detection based on Gaussian mixture variational autoencoder, IEEE Trans. Veh. Technol. 72 (11) (2023) 13826–13835.

[105] Office for Coastal Management, MarineCadastre.gov AIS data, 2024, https://hub.marinecadastre.gov/pages/vesseltraffic., Accessed: [2026-03-30].

[106] H. Cheng, W. Liao, M.Y. Yang, B. Rosenhahn, M. Sester, Amenet: attentive maps encoder network for trajectory prediction, ISPRS J. Photogramm. Remote Sens. 172 (2021) 253–266.

[107] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, L. Eckstein, The ind dataset: a drone dataset of naturalistic road user trajectories at German intersections, in: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2020, pp. 1929–1934, https://levelxdata.com/ind-dataset/.

[108] P. Xu, J.-B. Hayet, I. Karamouzas, Context-aware timewise VAEs for real-time vehicle trajectory prediction, IEEE Robot. Autom. Lett. 8 (9) (2023) 5440–5447.

[109] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: a multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11621–11631, https://www.nuscenes.org/.

[110] M. Lee, S.S. Sohn, S. Moon, S. Yoon, M. Kapadia, V. Pavlovic, Muse-vae: multi-scale VAE for environment-aware long term trajectory prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2221–2230.

[111] B. Ivanovic, K. Leung, E. Schmerling, M. Pavone, Multimodal deep generative models for trajectory prediction: a conditional variational autoencoder approach, IEEE Robot. Autom. Lett. 6 (2) (2020) 295–302.

[112] P. Xu, J.-B. Hayet, I. Karamouzas, Socialvae: human trajectory prediction using timewise latents, in: European Conference on Computer Vision, Springer, 2022, pp. 511–528.

[113] J. Wiederer, A. Bouazizi, M. Troina, U. Kressel, V. Belagiannis, Anomaly detection in multi-agent trajectories for automated driving, in: Conference on Robot Learning, PMLR, 2022, pp. 1223–1233.

[114] F. Zhao, X. Cao, J. Zhao, Y. Duan, X. Yang, X. Zhang, Masked graph autoencoder-based multi-agent dynamic relational inference model for trajectory prediction, Neurocomputing 634 (2025) 129922.

[115] T. Wei, Y. Lin, Y. Lin, S. Guo, L. Zhang, H. Wan, Micro-macro spatial-temporal graph-based encoder-decoder for map-constrained trajectory recovery, IEEE Trans. Knowl. Data Eng. 36 (11) (2024) 6574–6587.

[116] Y. Chen, W. Huang, K. Zhao, Y. Jiang, G. Cong, Self-supervised representation learning for geospatial objects: a survey, Inf. Fusion 123 (2025) 103265.
[117] J. Li, D. Hong, L. Gao, J. Yao, K. Zheng, B. Zhang, J. Chanussot, Deep learning in multimodal remote sensing data fusion: a comprehensive review, Int. J. Appl. Earth Obs. Geoinf. 112 (2022) 102926.
[118] S. Choudhury, A. Vasim, M. Schmitt, B. Banerjee, MIMRS: a survey on masked image modeling in remote sensing, in: IGARSS 2025-2025 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2025, pp. 2309–2314.
[119] S. Lu, J. Guo, J.R. Zimmer-Dauphinee, J.M. Nieusma, X. Wang, P. VanValkenburgh, S.A. Wernke, Y. Huo, Vision foundation models in remote sensing: a survey, IEEE Geosci. Remote Sens. Mag. 13 (3) (2025) 190–215.
[120] C. Huo, K. Chen, S. Zhang, Z. Wang, H. Yan, J. Shen, Y. Hong, G. Qi, H. Fang, Z. Wang, When remote sensing meets foundation model: a survey and beyond, Remote Sens. 17 (2) (2025) 179.
[121] Q. Zhang, H. Wang, H. Wen, C. Long, L. Su, X. He, T. Wu, C.S. Jensen, S.-M. Yiu, H. Yin, A survey of generative techniques for spatial-temporal data mining, Data Sci. Eng. (2026) 1–25.
[122] M.F. Goodchild, Geographical information science, Int. j. geogr. inf. syst. 6 (1) (1992) 31–45.
[123] K. Janowicz, S. Gao, G. McKenzie, Y. Hu, B. Bhaduri, GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond, Int. J. Geogr. Inf. Sci. 34 (4) (2020) 625–636.
[124] G. Mai, K. Janowicz, Y. Hu, S. Gao, B. Yan, R. Zhu, L. Cai, N. Lao, A review of location encoding for GeoAI: methods and applications, Int. J. Geogr. Inf. Sci. 36 (4) (2022) 639–673.
[125] W.R. Tobler, A computer movie simulating urban growth in the Detroit region, Econ. Geogr. 46 (sup1) (1970) 234–240.
[126] M.F. Goodchild, The validity and usefulness of laws in geographic information science and geography, Ann. Assoc. Am. Geogr. 94 (2) (2004) 300–303.
[127] L. Anselin, What is special about spatial data? Alternative perspectives on spatial data analysis (89-4), 1989, https://escholarship.org/uc/item/3ph5k0d4 (Accessed: 15 July 2026).
[128] G. De Marsily, F. Delay, J. Gonçalvès, P. Renard, V. Teles, S. Violette, Dealing with spatial heterogeneity, Hydrogeol. J. 13 (1) (2005) 161–183.
[129] S. Dodge, R. Weibel, A.-K. Lautenschütz, Towards a taxonomy of movement patterns, Inf. Vis. 7 (3–4) (2008) 240–252.
[130] Y. Yuan, M. Raubal, Measuring similarity of mobile phone user trajectories–a spatio-temporal edit distance method, Int. J. Geogr. Inf. Sci. 28 (3) (2014) 496–520.
[131] T. Hägerstrand, What about people in regional science, Papers Reg. Sci. Assoc. 24 (1) (1970) 6–21.
[132] H.J. Miller, Modelling accessibility using space-time prism concepts within geographical information systems, Int. J. Geogr. Inf. Syst. 5 (3) (1991) 287–301.
[133] S. Gao, Geospatial Artificial Intelligence (GeoAI), vol. 10, Oxford University Press New York, 2021.
[134] S. Gao, Y. Hu, W. Li, Introduction to geospatial artificial intelligence (GeoAI), in: Handbook of Geospatial Artificial Intelligence, CRC Press, 2023, pp. 3–16.
[135] T. VoPham, J.E. Hart, F. Laden, Y.-Y. Chiang, Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology, Environ. Health 17 (1) (2018) 40.
[136] X.X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, F. Fraundorfer, Deep learning in remote sensing: a comprehensive review and list of resources, IEEE Geosci. Remote Sens. Mag. 5 (4) (2017) 8–36.
[137] Y. Zheng, Trajectory data mining: an overview, ACM Trans. Intell. Syst. Technol. (TIST) 6 (3) (2015) 1–41.
[138] A. Graser, A.N. Jalali, J. Lampert, A. Weißenfeld, K. Janowicz, Deep learning from trajectory data: a review of deep neural networks and the trajectory data representations to train them, in: EDBT/ICDT Workshops, 2023.
[139] A. Boutayeb, I. Lahsen-Cherif, A. El Khadimi, When machine learning meets geospatial data: a comprehensive GeoAI review, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 18 (2025) 13135–13191.
[140] S. Wang, Z. Bao, J.S. Culpepper, G. Cong, A survey on trajectory data management, analytics, and learning, ACM Comput. Surv. (CSUR) 54 (2) (2021) 1–36.
[141] J.D. Mazimpaka, S. Timpf, Trajectory data mining: a review of methods and applications, J. Spat. Inf. Sci. 2016 (13) (2016) 61–99.
[142] R.D.S. Mello, V. Bogorny, L.O. Alvares, L.H.Z. Santana, C.A. Ferrero, A.A. Frozza, G.A. Schreiner, C. Renso, MASTER: a multiple aspect view on trajectories, Trans. GIS 23 (4) (2019) 805–822.
[143] J. He, H. Chen, Y. Chen, X. Tang, Y. Zou, Variable-based spatiotemporal trajectory data visualization illustrated, IEEE Access 7 (2019) 143646–143672.
[144] X. Kong, M. Li, K. Ma, K. Tian, M. Wang, Z. Ning, F. Xia, Big trajectory data: a survey of applications and services, IEEE Access 6 (2018) 58295–58306.
[145] J. He, H. Chen, Y. Chen, X. Tang, Y. Zou, Diverse visualization techniques and methods of moving-object-trajectory data: a review, ISPRS Int. J. Geo-Inf. 8 (2) (2019) 63.
[146] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (Dec) (2002) 583–617.
[147] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: human trajectory prediction in crowded spaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 961–971.
[148] B. Ivanovic, M. Pavone, The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2375–2384.
[149] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. 26 (1) (2003) 43–49.
[150] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing images using the hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intell. 15 (9) (2002) 850–863.
[151] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[152] Y. Wang, J. Wang, G. Su, X. Xu, Cloud computing for large-scale resource computation and storage in machine learning, J. Theory Pract. Eng. Sci. ISSN 2790 (2024) 1513.
[153] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[154] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.
[155] S. Gomez-Gonzalez, S. Prokudin, B. Schölkopf, J. Peters, Real time trajectory prediction using deep conditional generative models, IEEE Robot. Autom. Lett. 5 (2) (2020) 970–976.
[156] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[157] K. Cho, B. Van Merriënboer, Ç. Gulçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.
[158] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59.
[159] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114, 2013.
[160] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[161] T.N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308, 2016.
[162] R. Jiao, J. Bai, X. Liu, T. Sato, X. Yuan, Q.A. Chen, Q. Zhu, Learning representation for anomaly detection of vehicle trajectories, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, pp. 9699–9706.
[163] H. Su, S. Liu, B. Zheng, X. Zhou, K. Zheng, A survey of trajectory distance measures and performance evaluation, VLDB J. 29 (1) (2020) 3–32.
[164] A. Sepas-Moghaddam, A. Etemad, Deep gait recognition: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 264–284.
[165] C. Wan, L. Wang, V.V. Phoha, A survey on gait recognition, ACM Comput. Surv. (CSUR) 51 (5) (2018) 1–35.
[166] Z. Yu, H. Wu, Z. Yin, K. Liu, R. Zhang, Vessel trajectory segmentation: a survey, in: International Conference on Database Systems for Advanced Applications, Springer, 2023, pp. 166–180.
[167] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[168] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, M. Wilhelm, Bringing up OpenSky: a large-scale ADS-B sensor network for research, in: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IEEE, 2014, pp. 83–94, https://opensky-network.org/.
[169] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237, http://www.cvlibs.net/datasets/kitti/.
[170] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, R. Yang, The apolloscape dataset for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 954–960, http://apolloscape.auto/. https://github.com/ApolloScapeAuto/dataset-api?tab=readme-ov-file#data-download.
[171] L. Codecá, R. Frank, S. Faye, T. Engel, Luxembourg sumo traffic (lust) scenario: traffic demand evaluation, IEEE Intell. Transp. Syst. Mag. 9 (2) (2017) 52–63, https://github.com/lcodeca/LuSTScenario.
[172] P.A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using sumo, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Ieee, 2018, pp. 2575–2582.
[173] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in: Conference on Robot Learning, PMLR, 2017, pp. 1–16.
[174] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, J. Yu, Difftraj: generating GPS trajectory with diffusion probabilistic model, Adv. Neural Inf. Process. Syst. 36 (2023) 65168–65188.
[175] Y. Zhu, J.J. Yu, X. Zhao, Q. Liu, Y. Ye, W. Chen, Z. Zhang, X. Wei, Y. Liang, Controltraj: controllable trajectory generation with topology-constrained diffusion model, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 4676–4687.
[176] C. Chu, H. Zhang, P. Wang, F. Lu, Simulating human mobility with a trajectory generation framework based on diffusion model, Int. J. Geogr. Inf. Sci. 38 (5) (2024) 847–878.
[177] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, J. Lu, Stochastic trajectory prediction via motion indeterminacy diffusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17113–17122.
[178] W. Mao, C. Xu, Q. Zhu, S. Chen, Y. Wang, Leapfrog diffusion model for stochastic trajectory prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5517–5526.
[179] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov, et al., Motiondiffuser: controllable multi-agent motion prediction using diffusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9644–9653.

[180] I. Bae, Y.-J. Park, H.-G. Jeon, Singulartrajectory: universal trajectory predictor using diffusion model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 17890–17901.
[181] Y. Lipman, R.T. Chen, H. Ben-Hamu, M. Nickel, M. Le, Flow matching for generative modeling, arXiv preprint arXiv:2210.02747, 2022.
[182] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, Y. Bengio, Improving and generalizing flow-based generative models with minibatch optimal transport, arXiv preprint arXiv:2302.00482, 2023.
[183] S. Ye, M.C. Gombolay, Efficient trajectory forecasting and generation with conditional flow matching, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2024, pp. 2816–2823.
[184] Y. Fu, Q. Yan, L. Wang, K. Li, R. Liao, Moflow: one-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 17282–17293.
[185] Q. Yan, B. Zhang, Y. Zhang, D. Yang, J. White, D. Chen, J. Liu, L. Liu, B. Zhuang, S. Shi, et al., Trajflow: multi-modal motion prediction via flow matching, in: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2025, pp. 3923–3928.
[186] Y. Luo, Z. Cao, X. Jin, K. Liu, L. Yin, Deciphering human mobility: inferring semantics of trajectories with large language models, in: 2024 25th IEEE International Conference on Mobile Data Management (MDM), IEEE, 2024, pp. 289–294.
[187] L. Gong, Y. Lin, X. Zhang, Y. Lu, X. Han, Y. Liu, S. Guo, Y. Lin, H. Wan, Mobility-llm: learning visiting intentions and travel preference from human mobility data with large language models, Adv. Neural Inf. Process. Syst. 37 (2024) 36185–36217.
[188] P.S. Chib, P. Singh, Lg-traj: LLM guided pedestrian trajectory prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 6802–6812.
[189] M. Peng, X. Guo, X. Chen, K. Chen, M. Zhu, L. Chen, F.-Y. Wang, Lc-llm: explainable lane-change intention and trajectory predictions with large language models, Commun. Transp. Res. 5 (2025) 100170.
[190] Z. Lan, L. Liu, B. Fan, Y. Lv, Y. Ren, Z. Cui, Traj-LLM: a new exploration for empowering trajectory prediction with pre-trained large language models, IEEE Trans. Intell. Veh. 10 (2) (2025) 794–807.
[191] K. Yang, Z. Guo, G. Lin, H. Dong, Z. Huang, Y. Wu, D. Zuo, J. Peng, Z. Zhong, X. Wang, et al., Trajectory-llm: a language-based data generator for trajectory prediction in autonomous driving, in: The Thirteenth International Conference on Learning Representations, 2025.
[192] C. Ju, J. Liu, S. Sinha, H. Xue, F. Salim, Trajllm: a modular LLM-enhanced agent-based framework for realistic human trajectory simulation, in: Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 2847–2850.
[193] Y. Du, J. Feng, J. Zhao, Y. Li, Trajagent: an LLM-agent framework for trajectory modeling via large-and-small model collaboration, Adv. Neural Inf. Process. Syst. 38 (2026) 21595–21625.
[194] Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. Lobell, S. Ermon, Satmae: pre-training transformers for temporal and multi-spectral satellite imagery, Adv. Neural Inf. Process. Syst. 35 (2022) 197–211.
[195] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.
[196] X. Sun, P. Wang, W. Lu, Z. Zhu, X. Lu, Q. He, J. Li, X. Rong, Z. Yang, H. Chang, et al., RingMo: a remote sensing foundation model with masked image modeling, IEEE Trans. Geosci. Remote Sens. 61 (2022) 1–22.
[197] C.J. Reed, R. Gupta, S. Li, S. Brockman, C. Funk, B. Clipp, K. Keutzer, S. Candido, M. Uyttendaele, T. Darrell, Scale-mae: a scale-aware masked autoencoder for multiscale geospatial representation learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4088–4099.
[198] Y. Wang, H.H. Hernández, C.M. Albrecht, X.X. Zhu, Feature guided masked autoencoder for self-supervised learning in remote sensing, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 18 (2024) 321–336.
[199] D. Ibanez, R. Fernandez-Beltran, F. Pla, N. Yokoya, Masked auto-encoding spectral–spatial transformer for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–14.
[200] L. Scheibenreif, M. Mommert, D. Borth, Masked vision transformers for hyperspectral image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2166–2176.
[201] X. Li, D. Hong, J. Chanussot, S2MAE: a spatial-spectral pretraining foundation model for spectral remote sensing data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 24088–24097.
[202] D. Hong, B. Zhang, X. Li, Y. Li, C. Li, J. Yao, N. Yokoya, H. Li, P. Ghamisi, X. Jia, et al., SpectralGPT: spectral remote sensing foundation model, IEEE Trans. Pattern Anal. Mach. Intell. 46 (8) (2024) 5227–5244.
[203] D. Wang, M. Hu, Y. Jin, Y. Miao, J. Yang, Y. Xu, X. Qin, J. Ma, L. Sun, C. Li, et al., HyperSIGMA: hyperspectral intelligence comprehension foundation model, IEEE Trans. Pattern Anal. Mach. Intell. 47 (8) (2025) 6427–6444.
[204] X. Guo, J. Lao, B. Dang, Y. Zhang, L. Yu, L. Ru, L. Zhong, Z. Huang, K. Wu, D. Hu, et al., Skysense: a multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 27672–27683.
[205] Z. Xiong, Y. Wang, F. Zhang, A.J. Stewart, J. Hanna, D. Borth, I. Papoutsis, B.L. Saux, G. Camps-Valls, X.X. Zhu, Neural plasticity-inspired multimodal foundation model for earth observation, arXiv preprint arXiv:2403.15356, 2024.

Author biography

Andreas Karathanasis is a Research Assistant at the Informatics and Telematics Institute (ITI) of the Centre for Research and Technology Hellas (CERTH) and is currently pursuing a Master of Philosophy (MPhil) in Computer Science and Informatics at the Department of Informatics and Telematics, Harokopio University of Athens, Greece. He received his B.Sc. degree in Informatics and Telematics from Harokopio University of Athens in 2024. His research interests focus on GeoAI, geospatial data analysis, deep learning, and machine learning.

John Violos is an Assistant Professor in the Dept. of Geography at Harokopio University of Athens, specializing in Geospatial Databases and Big Geospatial Data. He served as a member of the European Commission Digital Single Market working group on the code of conduct for switching and porting data between cloud service providers. He has held research positions at École de Technologie Supérieure (ETS), Université du Québec, Canada; the Information Technologies Institute, Centre for Research and Technology Hellas (CERTH); and the Institute of Communication and Computer Systems (ICCS), Greece. He has participated in more than 18 internationally funded research projects (EU, Canada, and South Korea), serving as Work Package Leader and Task Leader. He has taught more than 12 courses as an Adjunct Professor at the Department of Informatics and Telematics of Harokopio University of Athens, as well as in the MSc Program in Bioinformatics at National and Kapodistrian University of Athens. He has authored over 80 publications in journals and conference proceedings and has served as a guest editor and reviewer for more than 30 international journals, as well as a program committee member and reviewer for over 30 international conferences. He holds a Diploma in Electrical and Computer Engineering from the National Technical University of Athens, during which he worked as an intern at the Hellenic Public Power Corporation (HPPC), and a PhD from the same university. His awards include the Best Paper Award at IEEE CISOSE 2023, Best Paper Award at IEEE iThings 2021, the Best Course Teaching Award (Adjunct Professor) in 2021, and the Thomaidion Award in 2017.

Iraklis Varlamis is a Professor in the Department of Informatics and Telematics at Harokopio University, specializing in Data Management. He holds a PhD from the Athens University of Economics and Business and an MSc in Information Systems Engineering from UMIST, UK. His research interests focus on data mining and social network analysis, as well as recommender systems with applications in social media and the real world. He has published more than 250 articles in international journals and conferences and has over 5,700 citations to his work. He holds two patents from the Hellenic Industrial Property Organization (OBI) and an international patent from the US Patent Office in the domain of web content management. He is the scientific coordinator at Harokopio University for various European (H2020, ECSEL, REC) and national research projects. He also plays a key role in national projects, collaborating with researchers from other faculties of Harokopio University and other universities.

Aris Leivadeas is a Professor with the Dept. of Software and Information Technology Engineering at ETS. From 2015 to 2018, he was a postdoc in the Dept. of SCE, at Carleton University. In parallel, Aris worked as an intern at Ericsson and then at Cisco in Ottawa, Canada. He received his diploma in ECE from the University of Patras in 2008, his MSc degree in Engineering from King's College London in 2009, and his Ph.D. degree in ECE from NTUA in 2015. His research interests include cloud computing, IoT, and network optimization and management. He received a Best Paper Award at ICPE '18 and iThings'21, and the Best Presentation Award at IEEE HPSR '20.

Konstantinos Tserpes is an Assistant Professor at the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA). Prior to 2024, he was a faculty member in the Department of Informatics and Telematics at Harokopio University of Athens. He holds a PhD in Distributed Systems from NTUA. His research interests include distributed software systems, machine learning, and data analytics. He has authored over 100 scientific publications and has been actively involved in student supervision, teaching, and research projects. He has extensive experience in supervising undergraduate, postgraduate, and PhD students, and has taught a wide range of courses across two academic institutions. In addition to his academic work, he has coordinated several EU and nationally funded research projects, serving as both scientific and general coordinator.