GEODIFFUSION: SPATIO-TEMPORAL OFFSET-AWARE LATENT DIFFUSION FOR IRREGULAR TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world spatio-temporal sequences often consist of irregularly sampled trajectory observations, where spatial displacements are coupled with unequal time intervals, creating a mismatch between sequence adjacency and the scale of physical transitions. This challenges a basic assumption of standard sequence modeling: adjacent tokens correspond to directly comparable physical transitions. Sequence indices alone cannot distinguish these transition scales. To address this problem, we propose GeoDiffusion, a spatio-temporal offset-aware latent diffusion framework that builds on diffusion models' ability to model joint trajectory distributions. Specifically, spatio-temporal offset encoding jointly embeds the spatial displacement and time interval between consecutive observations. Learnable latent queries then aggregate position and offset features, enabling the denoising network to reconstruct latent representations that incorporate both trajectory locations and transition scales. Derived from the trajectories themselves, the offset features provide a transition-oriented inductive bias. Experiments on five maritime and urban trajectory datasets show favorable aggregate prediction and imputation performance across the evaluated settings, with exceptions in individual comparisons. Jaccard comparisons and averaged POR, DTW, and ADE measurements characterize visited-grid overlap, sequence alignment, and coordinate accuracy.

1 INTRODUCTION

Spatio-temporal trajectory modeling concerns the evolution of moving objects through geographic space and physical time. It underpins the prediction of future locations and the reconstruction of missing movement records in maritime monitoring and urban mobility analysis (Capobianco et al., 2021; Rong et al., 2022; Solatorio, 2023; Hsu et al., 2024). In these settings, trajectories are typically observed at irregular intervals because of heterogeneous acquisition frequencies and interrupted observations. Irregular sampling creates a mismatch between sequence adjacency and physical transition scales. Identical index separations can correspond to substantially different elapsed times, while similar spatial displacements can reflect different rates of movement.

Since consecutive records encode spatial displacements over unequal durations. Modeling such data requires a representation of both the observed positions and the spatial–temporal relationships underlying their transitions. Temporal representations and continuous-time models already address irregular observations. Time2Vec encodes time as a learned vector (Kazemi et al., 2019); multi-time attention and ContiFormer model observations at nonuniform timestamps (Shukla & Marlin, 2021; Chen et al., 2023). CSDI uses observed data and temporal information for conditional diffusion imputation (Tashiro et al., 2021), and stochastic process diffusion directly models continuous functions (Biloš et al., 2023). A temporal index-based positional representation specifies observation order but does not, by itself, distinguish these transition scales.

Addressing this mismatch requires a generative model that accounts for heterogeneous physical transitions and their dependencies across the trajectory. A displacement and its duration characterize an individual transition, while the surrounding observations constrain how that transition fits into the overall movement. Diffusion models provide a suitable formulation for learning these dependencies through the reconstruction of a joint trajectory representation at multiple noise levels (Ho et al., 2020). With a sequence-wide denoising network, each reverse step can update the representation as a whole, allowing information from different portions of the trajectory to contribute to generation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Following this principle, we propose GeoDiffusion, a spatio-temporal offset-aware latent diffusion framework. Spatio-Temporal Offset Encoding (STOE) jointly embeds the coordinate displacement and logarithmic time interval between consecutive observations. Learnable latent queries attend to both spatial-token features and offset embeddings, aggregating a variable-length trajectory into a fixed-size latent representation (Jaegle et al., 2021). A Transformer denoiser then learns to reconstruct this representation from Gaussian perturbations, following the latent diffusion formulation (Rombach et al., 2022; Lovelace et al., 2023). This architecture assigns complementary roles to representation learning and generation: the encoder makes location and transition information available in latent space, while diffusion learns the distribution of the resulting trajectory representations. The framework is evaluated through forecasting, missing-segment completion, and endpoint-conditioned generation, which expose different subsets of trajectory observations.

We evaluate GeoDiffusion on three maritime AIS datasets and two urban mobility datasets. Across the five observation–target settings within each dataset, GeoDiffusion achieves the highest mean Jaccard similarity for both forecasting and completion, while individual comparisons identify settings in which competing methods perform better. Component ablations examine the contribution of the proposed representation: removing STOE reduces Jaccard similarity from 0.7491 to 0.7100 and increases coordinate ADE from 1.36 to 1.61 on the private-data setting. These results support the contribution of explicit spatial–temporal transition features within the evaluated architecture and show favorable full-model performance across multiple trajectory domains. Establishing whether these gains specifically arise from improved modeling of heterogeneous transition scales requires matched time-aware comparisons and controlled variations in observation intervals.

2 Related Work

Irregular time-series representations. Time2Vec provides a model-agnostic embedding of time (Kazemi et al., 2019). Multi-Time Attention Networks use continuous-time embeddings and attention to obtain fixed-length representations of irregular observations (Shukla & Marlin, 2021), while ContiFormer combines attention with continuous-time modeling (Chen et al., 2023). These methods motivate comparisons between joint spatial–temporal offsets, time embeddings, and continuous-time attention. Informer, Autoformer, TimesNet, and Crossformer are general sequence-modeling baselines (Zhou et al., 2021; Wu et al., 2021; 2023; Zhang & Yan, 2023); TimesNet models temporal variation through two-dimensional structure and is not a diffusion model.

Diffusion for temporal data and trajectories. DDPMs learn a reverse process for Gaussian corruption (Ho et al., 2020). CSDI conditions diffusion on observed values and temporal embeddings and can accommodate irregular intervals (Tashiro et al., 2021). Stochastic process diffusion instead defines a generative process over continuous functions (Biloš et al., 2023). Latent Laplace Diffusion is a recent latent approach to irregular multivariate time series (You et al., 2026). In trajectory generation, DiffTraj models GPS paths (Zhu et al., 2023), and Diff-RNTraj incorporates road-network structure (Wei et al., 2024). Closely related maritime approaches include DiffuTraj, which conditions on vessel history, interactions, and environment (Li et al., 2024), STGDPM, which models vessel interactions with spatio-temporal graphs (Jin et al., 2025), and physics-guided diffusion for vessel imputation (Zhang et al., 2024). GeoDiffusion emphasizes compressed representations with explicit displacement and gap features.

Event generation and conditional sampling. Diffusion point processes predict future events in continuous time (Zhou et al., 2023); spatio-temporal diffusion point processes jointly model event locations and times (Yuan et al., 2023). ADiff4TPP uses asynchronous diffusion schedules (Mukherjee et al., 2025). These settings distinguish generating observation times from predicting locations at specified times. For conditional generation, Diffuser uses trajectory denoising for planning (Janner et al., 2022), and RePaint uses a pretrained diffusion model with a modified inference procedure for inpainting (Lugmayr et al., 2022).

3 Problem Formulation

We study irregular spatio-temporal sequence modeling, where observations are collected at non-uniform temporal intervals and may contain missing segments. Let

$$\mathbf{x} = \{s_1, s_2, \ldots, s_N\} \tag{1}$$

denote a spatio-temporal sequence, where each observation $s_i = (\mathbf{p}_i, \tau_i)$ consists of a spatial state $\mathbf{p}_i \in \mathcal{P}$ and its timestamp $\tau_i \in \mathbb{R}^+$. In regular sequence modeling, adjacent tokens are usually assumed to correspond to comparable temporal transitions. In irregular spatio-temporal data, this assumption no longer holds: two neighboring observations in sequence order may be separated by very different temporal gaps and spatial displacements.

The central difficulty is therefore the mismatch between **sequence adjacency** and **physical transition scale**. For two adjacent observations s_i and s_{i+1}, we define their spatial displacement and temporal gap as

$$\Delta\mathbf{p}_i = \mathbf{p}_{i+1} - \mathbf{p}_i, \quad \Delta\tau_i = \tau_{i+1} - \tau_i. \tag{2}$$

When $\Delta\tau_i$ and $\Delta\mathbf{p}_i$ vary significantly across i, the local transition distribution becomes non-stationary with respect to the sequence index. A model that relies only on token order may assign similar transition dynamics to physically incomparable movements, leading to unstable generation under long gaps, missing observations, and long-horizon forecasting.

To explicitly characterize this irregularity, we define the local spatio-temporal offset as

$$\boldsymbol{\delta}_i = [\Delta\mathbf{p}_i, \log(1 + |\Delta\tau_i|)], \quad i = 1, \ldots, N-1. \tag{3}$$

The complete offset sequence is denoted as $\boldsymbol{\Delta}(\mathbf{x}) = \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N-1}$. Unlike absolute timestamps or positional indices, $\boldsymbol{\Delta}(\mathbf{x})$ represents the physical transition scale between consecutive observations. We therefore formulate irregular spatio-temporal modeling as learning an offset-conditioned generative prior

$$p_\theta(\mathbf{x} \mid \boldsymbol{\Delta}(\mathbf{x})). \tag{4}$$

In this work, the prior is learned through latent diffusion. Given a sequence $\mathbf{x}$, an encoder maps the sequence and its offset structure into a latent representation

$$\mathbf{z}_0 = f_\theta^{\text{enc}}(\mathbf{x}, \boldsymbol{\Delta}(\mathbf{x})). \tag{5}$$

The forward diffusion process corrupts $\mathbf{z}_0$ into $\mathbf{z}_t$ as

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1 - \bar{\alpha}_t)\mathbf{I}\right), \tag{6}$$

where $\bar{\alpha}_t = \prod_{r=1}^{t}(1 - \beta_r)$. The denoising model learns to recover the clean latent state from the noisy latent variable while being conditioned on the offset structure:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x},t,\boldsymbol{\epsilon}}\left[\left|g_\phi\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t, \boldsymbol{\Delta}(\mathbf{x})\right) - \mathbf{z}_0\right|_2^2\right], \tag{7}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This objective encourages the denoising dynamics to be aware of heterogeneous spatial and temporal transition scales, rather than relying solely on sequence order.

Many downstream spatio-temporal tasks can further be formulated as posterior inference under this learned prior. Let $\Omega \subseteq 1, \ldots, N$ denote the constrained indices, and let $\mathbf{x}_\Omega$ be the known observations. The inference target is

$$p_{\theta,\phi}(\mathbf{x} \mid \mathbf{x}_\Omega, \boldsymbol{\Delta}(\mathbf{x})). \tag{8}$$

Different choices of Ω correspond to different tasks: trajectory prediction constrains the prefix, imputation constrains observed tokens, and route planning constrains endpoints or target regions. Therefore, prediction, imputation, and planning can be unified as posterior-constrained sampling from the same offset-aware diffusion prior.

4 METHODOLOGY

We propose GeoDiffusion, an offset-aware latent diffusion framework for irregular spatio-temporal sequence modeling. The goal of GeoDiffusion is to learn a generative prior that is aware of heterogeneous physical transition scales and can be reused for multiple downstream inference tasks. The key design principle is that irregularity should be modeled at the level of local physical transitions rather than only through sequence positions. Given an input trajectory $\mathbf{x}$, GeoDiffusion first constructs relative spatio-temporal offsets between consecutive observations and injects them into the latent trajectory representation. The resulting offset-aware latent representation is then used as the clean signal for diffusion training. During inference, different downstream tasks are realized by imposing different constraint sets during the reverse denoising process.

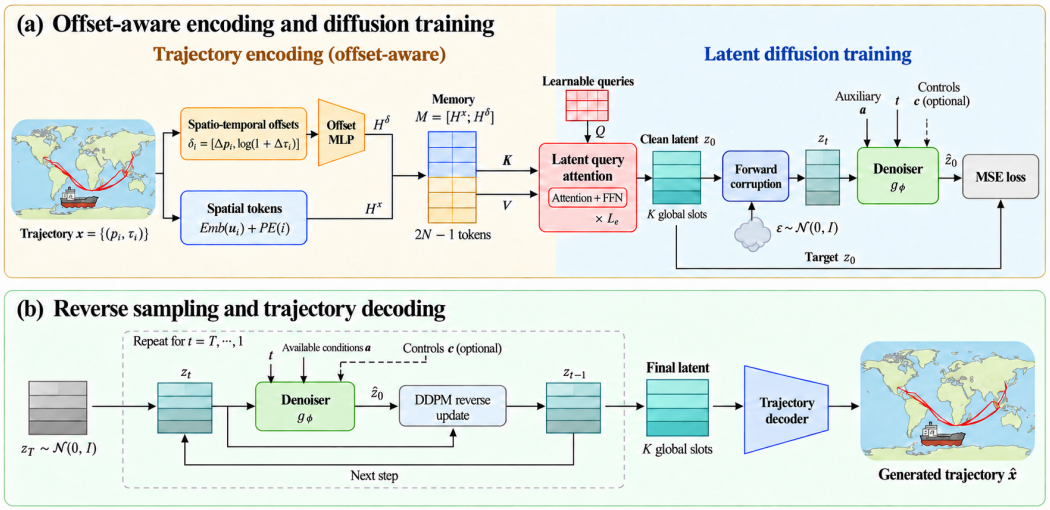


Figure 1: Overview of GeoDiffusion. (a) Spatial-token embeddings and spatio-temporal offset embeddings form a shared memory, which learnable latent queries aggregate into K global slots. The denoiser learns to recover the clean latent representation from its noisy counterpart. (b) Iterative reverse sampling produces a latent representation that is decoded into a trajectory. The diagram uses a for the available offset condition $\mathbf{\Delta}$ and c for optional external controls, embedded as $\mathbf{H}^c$.

Figure 1 summarizes the encoding, diffusion training, and trajectory generation procedures. The available offset information conditions the denoiser, while optional external controls specify additional generation requirements. For downstream tasks, posterior constraints are imposed during reverse sampling as described in Section 4.3.

4.1 Offset-Aware Sequence Representation

Given a trajectory $\mathbf{x}$, we first convert the spatial component into a tokenized representation. In geographic sequence modeling, $\mathbf{p}_i$ can be discretized into a spatial grid token; in other spatio-temporal domains, it can be represented by a continuous state embedding or a domain-specific spatial token. We denote the resulting token sequence as

$$\mathbf{u} = u_1, u_2, \ldots, u_N. \tag{9}$$

A token embedding layer maps each spatial token into a d-dimensional vector:

$$\mathbf{h}_i^x = \mathrm{Emb}(u_i) + \mathrm{PE}(i), \tag{10}$$

where $\mathrm{Emb}(\cdot)$ denotes the spatial token embedding and $\mathrm{PE}(i)$ is a learnable or sinusoidal positional embedding. The positional embedding preserves the order of observations, but it does not describe the physical scale of transitions between adjacent tokens. Therefore, relying on $\mathbf{h}_i^x$ alone is insufficient under irregular sampling.

To capture the physical transition scale, we explicitly construct the local spatio-temporal offset between s_i and s_{i+1} as $\boldsymbol{\delta}_i$ by Eq.(2) and Eq.(3). Each offset vector is projected into the same latent dimension through a two-layer MLP:

$$\mathbf{h}_i^\delta = \mathrm{MLP}_{\theta_\delta}(\boldsymbol{\delta}_i) \in \mathbb{R}^d. \tag{11}$$

The offset embeddings form a transition-level sequence

$$\mathbf{H}^\delta = \{\mathbf{h}_1^\delta, \ldots, \mathbf{h}_{N-1}^\delta\}. \tag{12}$$

Compared with absolute timestamps or generic positional encodings, $\mathbf{H}^\delta$ represents how the system physically evolves between neighboring observations. It therefore provides the model with transition-scale information that is otherwise missing from sequence order.

Spatio-Temporal Offset Encoding via Latent Query Attention. A direct way to incorporate offsets is to concatenate them with token embeddings. However, this treats offset information as an

auxiliary feature and does not explicitly control how transition-scale information enters the latent generative representation. Instead, GeoDiffusion uses a latent query attention mechanism to aggregate observation tokens and offset tokens into a compact offset-aware latent representation.

Let $\mathbf{H}^x = \{\mathbf{h}_1^x, \ldots, \mathbf{h}_N^x\} \in \mathbb{R}^{N\times d}$ denote the observation-token features, and let $\mathbf{H}^\delta \in \mathbb{R}^{(N-1)\times d}$ denote the offset-token features. We concatenate them as the memory sequence:

$$\mathbf{M} = [\mathbf{H}^x; \mathbf{H}^\delta] \in \mathbb{R}^{(2N-1)\times d}. \tag{13}$$

We introduce K learnable latent queries $\mathbf{Q} \in \mathbb{R}^{K\times d}$, which serve as compact latent slots for representing the entire trajectory. Each latent query attends to both observation tokens and offset tokens through multi-head cross-attention:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{M}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{W}_Q(\mathbf{M}\mathbf{W}_K)^\top}{\sqrt{d}}\right)\mathbf{M}\mathbf{W}_V. \tag{14}$$

The latent queries are updated through stacked attention blocks:

$$\mathbf{Z}^{(l+1)} = \mathrm{FFN}\left(\mathrm{MHA}\left(\mathbf{Z}^{(l)}, \mathbf{M}, \mathbf{M}\right) + \mathbf{Z}^{(l)}\right), \tag{15}$$

where $\mathbf{Z}^{(0)} = \mathbf{Q}$. After L_e layers, the encoder outputs the offset-aware latent trajectory representation:

$$\mathbf{z}_0 = f_\theta^{\mathrm{enc}}(\mathbf{x}, \boldsymbol{\Delta}(\mathbf{x})) = \mathbf{Z}^{(L_e)}. \tag{16}$$

This design has two advantages. ❶ The latent representation is conditioned on both observation content and physical transition scale. ❷ By using latent queries, the model learns a compact generative state that is suitable for latent diffusion, rather than directly diffusing over long and irregular raw sequences. Importantly, Spatio-Temporal Offset Encoding is not intended to replace self-attention. Self-attention captures dependencies among trajectory elements, while offset encoding provides explicit information about the physical scale of local transitions. The two mechanisms are complementary: self-attention models relational structure, and STOE informs the model whether adjacent observations correspond to short, long, slow, or abrupt transitions.

4.2 Offset-Conditioned Latent Diffusion

Given the offset-aware latent representation $\mathbf{z}_0$, GeoDiffusion learns a diffusion prior in the latent space. The forward diffusion process gradually corrupts $\mathbf{z}_0$ by adding Gaussian noise:

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{17}$$

where $t \in 1, \ldots, T$ is the diffusion timestep, $\bar{\alpha}_t = \prod_{r=1}^{t}(1-\beta_r)$, and $\beta_{t\,t=1}^{T}$ is the noise schedule. The reverse process is parameterized by a Transformer denoising network g_ϕ. Given a noisy latent state $\mathbf{z}_t$, the diffusion timestep t, and the offset condition $\boldsymbol{\Delta}(\mathbf{x})$, the denoiser predicts the clean latent representation:

$$\hat{\mathbf{z}}_0 = g_\phi(\mathbf{z}_t, t, \boldsymbol{\Delta}(\mathbf{x})). \tag{18}$$

In practice, the offset condition can be injected into the denoising network through cross-attention or adaptive conditioning layers. The denoiser therefore operates not only on the corrupted latent state, but also on the transition-scale structure that produced the latent trajectory. This makes the reverse dynamics aware of irregular sampling patterns. The training objective is

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{\mathbf{x},t,\boldsymbol{\epsilon}}\left[\left|g_\phi\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \boldsymbol{\Delta}(\mathbf{x})\right) - \mathbf{z}_0\right|_2^2\right], \tag{19}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This objective trains the model to recover clean offset-aware latent trajectories from noisy states. The denoising process is explicitly conditioned on relative spatial displacements and elapsed times between trajectory observations.

After reverse diffusion, the final latent representation $\hat{\mathbf{z}}_0$ is mapped back to the trajectory token space through a reconstruction decoder: $\hat{\mathbf{x}} = f_\psi^{\mathrm{dec}}(\hat{\mathbf{z}}_0)$.

Conditional Generation with External Controls. GeoDiffusion supports conditional generation by incorporating external control tokens into the denoising network. Let $\mathbf{c} = c_1, c_2, \ldots, c_M$ denote a set of condition tokens. These tokens are embedded as

$$\mathbf{H}^c = \mathrm{Enc}_c(\mathbf{c}) \in \mathbb{R}^{M \times d}. \tag{20}$$

The denoiser is then extended as

$$\hat{\mathbf{z}}_0 = g_\phi(\mathbf{z}_t, t, \mathbf{\Delta}(\mathbf{x}), \mathbf{H}^c). \tag{21}$$

We implement this conditioning using cross-attention layers inserted into the Transformer denoiser. Self-attention captures dependencies within the latent trajectory, while cross-attention allows the generated trajectory to align with external controls. The conditional training objective becomes

$$\mathcal{L}_{\mathrm{cond}} = \mathbb{E}_{\mathbf{x},t,\boldsymbol{\epsilon}} \left[\left| g_\phi \left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{\Delta}(\mathbf{x}), \mathbf{H}^c \right) - \mathbf{z}_0 \right|_2^2 \right]. \tag{22}$$

To improve controllability, we use classifier-free guidance. During training, the condition tokens are randomly dropped with probability p_{drop}, allowing the model to learn both conditional and unconditional denoising. During sampling, the guided prediction is computed as

$$\tilde{g}_\phi = (1+w) g_\phi(\mathbf{z}_t, t, \mathbf{\Delta}, \mathbf{H}^c) - w g_\phi(\mathbf{z}_t, t, \mathbf{\Delta}, \varnothing), \tag{23}$$

where w is the guidance scale and $\varnothing$ denotes the null condition. This allows GeoDiffusion to balance trajectory realism and condition satisfaction during generation.

4.3 Posterior-Constrained Sampling for Unified Inference

GeoDiffusion further uses the learned diffusion prior for multiple downstream tasks without task-specific retraining. The key observation is that trajectory prediction, imputation, and route planning can all be formulated as posterior inference under different constraint sets. Let $\Omega \subseteq 1, \ldots, N$ denote the constrained indices, and let $\mathbf{x}_\Omega$ be the known observations or specified constraints. The goal is to sample a complete trajectory from

$$p_{\theta,\phi}(\mathbf{x} \mid \mathbf{x}_\Omega, \mathbf{\Delta}(\mathbf{x})). \tag{24}$$

Different tasks correspond to different choices of Ω:

$$\Omega = \begin{cases} \{1, \ldots, m\}, & \text{prediction}, \\ \{i : s_i \text{ is observed}\}, & \text{imputation}, \\ \{1, N\} \text{ or target regions}, & \text{planning}. \end{cases} \tag{25}$$

Instead of training a separate model for each task, GeoDiffusion imposes these constraints during reverse diffusion. We first encode the constrained observations into latent tokens. During each reverse denoising step, the constrained latent components are clamped to remain consistent with the observed or specified tokens, while unconstrained components are updated by the learned denoising dynamics. Let $\mathbf{z}_t^\Omega$ and $\mathbf{z}_t^{\bar{\Omega}}$ denote the constrained and unconstrained parts of the latent state. At each reverse step, GeoDiffusion performs

$$\tilde{\mathbf{z}}_{t-1} = g_\phi(\mathbf{z}_t, t, \mathbf{\Delta}, \mathbf{H}^c), \tag{26}$$

followed by

$$\mathbf{z}_{t-1}^\Omega = \mathrm{Clamp}\left(\tilde{\mathbf{z}}_{t-1}^\Omega, f_\theta^{\mathrm{enc}}(\mathbf{x}_\Omega)\right), \mathbf{z}_{t-1}^{\bar{\Omega}} = \tilde{\mathbf{z}}_{t-1}^{\bar{\Omega}}. \tag{27}$$

This procedure enforces consistency with the constraints throughout the reverse process. The constrained tokens guide the denoising trajectory, while the learned prior generates the missing or future portions.

5 Experiments

5.1 Experiment Setup

Datasets. We conduct experiments on five real-world spatio-temporal trajectory datasets, including three maritime AIS datasets and two urban mobility datasets. For maritime trajectory modeling, we use the DMA AIS dataset, which contains AIS trajectories of vessels operating near the Danish

coastline[1], and the U.S. Coast dataset, which consists of AIS records collected from vessels along the United States coastline[2]. In addition, we use a private maritime dataset, the details of which are provided in Appendix A.2.1. This private dataset contains tanker AIS records collected from real-world maritime operations, covering trajectories of more than 3,500 oil tankers from January to June 2022. To further evaluate the generality of GeoDiffusion beyond maritime scenarios, we include two urban mobility datasets: Geolife Zheng et al. (2009; 2008; 2010) and Porto TaxiDe Brébisson et al. (2015); UCI Machine Learning Repository (2015). Geolife contains GPS trajectories collected from human daily mobility traces across urban environments, while Porto Taxi contains large-scale taxi trajectories collected from real-world urban transportation systems.

Baselines. The baselines covering general sequence modeling and task-specific trajectory methods. For general-purpose spatio-temporal modeling, we include Informer Zhou et al. (2021), Autoformer Wu et al. (2021), TimesNet Wu et al. (2023), Crossformer Zhang & Yan (2023), and GPT-ST Li et al. (2023), all adapted to grid-based trajectories via token embeddings. We further consider task-specific baselines for trajectory prediction (AISFuser Zhang et al. (2025), AIS-Hybrid Zhu et al. (2024)), trajectory imputation (DAISTIN Magnussen et al. (2023), PG-Diffusion Zhang et al. (2024)), and route planning (DiffTraj Zhu et al. (2023), Diff-RNTraj Wei et al. (2024), ShippingMap Liu et al. (2025), Incorporation Li & Yang (2023)).

Evaluation Metrics. We adopt two structure-aware similarity metrics: *Path Overlap Ratio (POR)* Xu et al. (2017) and *Jaccard Similarity*, to evaluate the visited-grid overlap of the generated trajectories. In addition, we report *Dynamic Time Warping (DTW)* Sakoe & Chiba (1978); Berndt & Clifford (1994) and *Average Displacement Error (ADE)* Alahi et al. (2016); Gupta et al. (2018) to assess order-sensitive trajectory similarity and coordinate-level prediction accuracy. DTW measures the temporal alignment cost between generated and ground-truth trajectories, while ADE computes the average pointwise displacement error over all predicted timesteps. Detail are provided in Appendix A.2.2.

Evaluation Tasks. ❶*Trajectory Prediction Task.* Given a historical trajectory segment, trajectory prediction aims to forecast the future sequence of spatial positions in continuous geographic space. This task is important for both maritime navigation and urban mobility analysis, supporting applications such as route forecasting, traffic coordination, collision risk assessment, and mobility behavior understanding. It requires the model to capture complex spatial-temporal dependencies while handling nonlinear dynamics, non-uniform sampling intervals, and heterogeneous movement patterns across different mobility domains. ❷*Trajectory Imputation Task.* Trajectory imputation aims to complete partially observed trajectories by inferring missing spatial-temporal segments from the available observations. This task is crucial for improving the quality and continuity of real-world mobility data, where missing records commonly arise from AIS signal loss, GPS sparsity, device failures, communication interruptions, or irregular data collection. Effective imputation can recover plausible intermediate paths and improve the reliability of downstream applications such as behavior analysis, anomaly detection, route mining, and mobility pattern modeling. ❸*Route Planning Task.* The route planning task generates a trajectory between specified origin and destination locations on the maritime grid. We evaluate the generated routes against reference trajectories using spatial overlap, sequence alignment, and coordinate error.

5.2 Main Results

Overview. We evaluate trajectory prediction and imputation on three maritime AIS datasets and two urban mobility datasets, and route planning on the three maritime datasets. We report average POR, DTW, and ADE across the five datasets for prediction and imputation, and use a Jaccard difference heatmap to compare individual dataset–setting pairs. For maritime prediction, a setting such as "10-5" denotes 10 days of observations followed by 5 days to predict. For urban prediction, the same notation denotes 10 observed tokens followed by 5 target tokens. More detailed experimental results are provided in Appendix A.2.3.

Trajectory Prediction. Figure 2 reports average prediction performance across five datasets. GeoDiffusion achieves the highest average POR and the lowest average DTW and ADE among the methods shown at each evaluated setting. The averaged scores generally worsen as the observed–future window increases, although individual datasets exhibit non-monotonic trends. These comparisons

[1] https://www.dma.dk/safety-at-sea/navigational-information/ais-data

[2] https://hub.marinecadastre.gov/pages/vesseltraffic

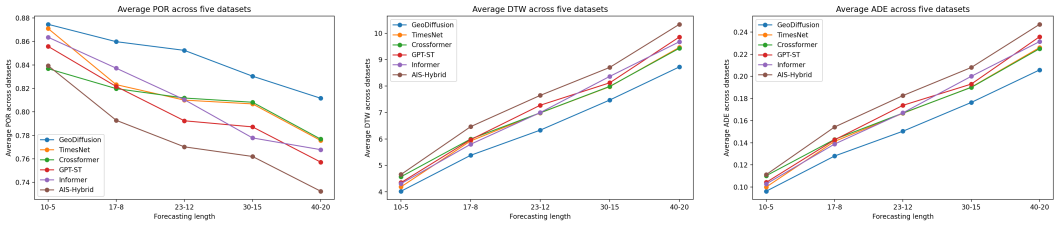


Figure 2: Trajectory prediction performance across five datasets under different forecasting lengths. The three panels show the average POR(↑), DTW(↓), and ADE(↓), respectively.

show favorable aggregate performance across the evaluated prediction windows. POR measures coverage of the reference trajectory's visited grid cells, DTW compares the ordered coordinate sequences, and ADE measures pointwise coordinate error. The lower average DTW and ADE in Figure 2 complement the overlap results by showing improved sequence alignment and coordinate accuracy among the displayed methods.

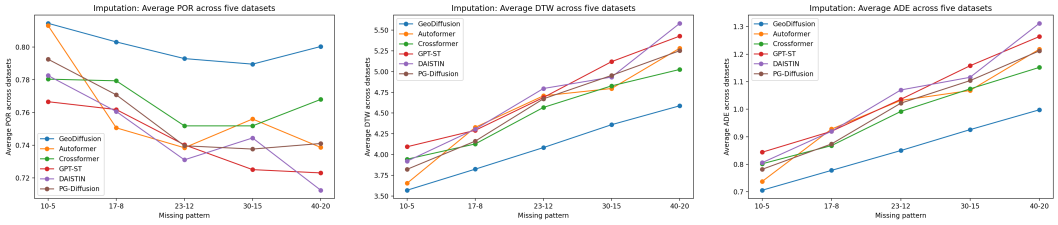


Figure 3: Trajectory imputation results across five datasets. We report average POR (↑), DTW (↓), and ADE (↓) under different missing patterns.

Trajectory Imputation. Figure 3 reports average imputation performance across five datasets. GeoDiffusion achieves the highest average POR and the lowest average DTW and ADE among the methods shown at each evaluated setting. The total–missing lengths range from 10-5 to 40-20: both the observed and missing lengths increase, while the missing fraction remains close to one half. The curves include recoveries between adjacent settings, so their variation reflects changes in sequence length and missing pattern rather than a monotonic increase in the missing fraction. The averaged DTW and ADE results indicate lower sequence-alignment and pointwise coordinate errors for GeoDiffusion among the displayed methods. Together with POR, they provide complementary measurements of the spatial coverage and coordinate reconstruction of incomplete trajectories.

Jaccard Comparison Across Datasets. Figure 4 complements the averaged metrics with the Jaccard difference between GeoDiffusion and the strongest baseline in each dataset–setting pair. GeoDiffusion leads in 23 of 25 prediction settings and 22 of 25 imputation settings, with one tie in imputation on U.S. Coast at 10-5. The four negative differences occur on the Private dataset: prediction at 10-5 and 30-15 (-0.050 and -0.065), and imputation at the same settings (-0.055 and -0.005). Thus, the aggregate improvements coexist with setting-specific exceptions on this dataset.

For urban trajectory imputation, the margin increases from 0.022 to 0.042 on Geolife and from 0.020 to 0.040 on Porto Taxi as the setting changes from 10-5 to 40-20. The maritime datasets show less uniform trends. Full Jaccard scores and baseline identities are in Tables 3 and 4 in the appendix.

Table 1: Route planning results on three datasets with two types of point pairs, including port-to-port pairs and randomly selected point pairs. We report POR (↑), Jaccard Similarity (↑), DTW (↓), and ADE (↓). Higher POR/Jaccard and lower DTW/ADE indicate better performance.

Tasks		GeoDiffusion				DiffTraj				Diff-RNTraj				ShippingMap				Incorporation			
		POR	Jaccard	DTW	ADE	POR	Jaccard	DTW	ADE	POR	Jaccard	DTW	ADE	POR	Jaccard	DTW	ADE	POR	Jaccard	DTW	ADE
Private Dataset	ports	**0.8688**	0.8377	**5.82**	**0.274**	0.7752	0.7418	7.36	0.348	0.7532	0.7169	7.91	0.371	0.7634	0.7288	7.64	0.360	0.7935	0.7547	7.18	0.337
	random	**0.8406**	0.8188	**6.47**	**0.309**	0.7518	0.7150	8.35	0.397	0.7565	0.7355	8.12	0.386	0.7720	0.7438	7.88	0.374	0.7869	0.7651	7.53	0.356
DMA AIS Dataset	ports	**0.9375**	0.9033	**3.74**	**0.181**	0.9213	0.8871	4.28	0.213	0.9099	0.8780	4.61	0.227	0.9327	0.8916	4.12	0.206	0.8975	0.8685	4.83	0.239
	random	**0.8594**	0.8263	**5.96**	**0.286**	0.8265	0.7916	6.89	0.331	0.8196	0.7849	7.12	0.342	0.8297	0.7948	6.77	0.326	0.8312	0.8015	6.64	0.318
U.S. Coast Dataset	ports	**0.9188**	0.8823	**4.38**	**0.209**	0.8884	0.8526	5.27	0.256	0.8667	0.8371	5.71	0.274	0.8811	0.8495	5.44	0.263	0.8790	0.8471	5.52	0.267
	random	**0.8250**	0.7910	**6.72**	**0.322**	0.7950	0.7591	7.58	0.366	0.7846	0.7486	7.91	0.381	0.7928	0.7554	7.66	0.371	0.7874	0.7522	7.82	0.378

Route Planning. Table 1 compares GeoDiffusion with route planning baselines on three maritime datasets. Across the available comparisons, GeoDiffusion achieves the highest POR and Jaccard scores and the lowest DTW and ADE in both endpoint settings. These metrics indicate stronger visited-grid overlap, closer alignment of coordinate sequences, and lower pointwise coordinate

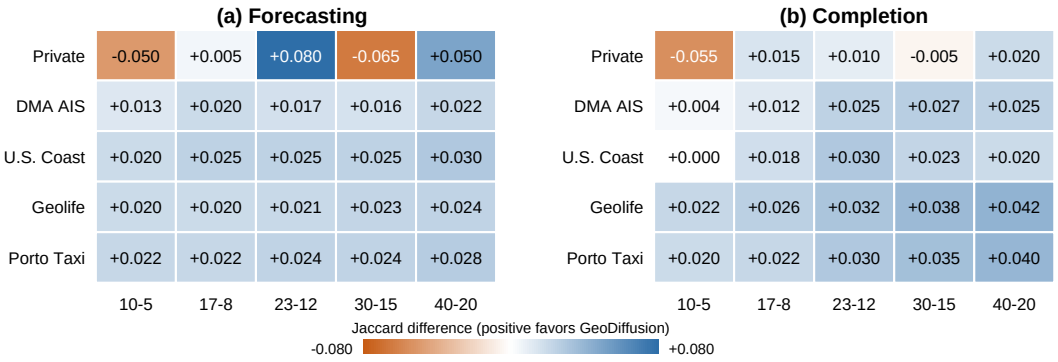


Figure 4: Jaccard differences across five datasets for (a) trajectory prediction (Forecasting) and (b) trajectory imputation (Completion). Each cell reports $\Delta J = J_{\text{GeoDiffusion}} - \max_{b \in \mathcal{B}} J_b$, where $\mathcal{B}$ contains the baselines for the corresponding task and the maximum is taken separately for each dataset and setting. Positive values favor GeoDiffusion; values are absolute score differences. Prediction settings denote observed–future lengths; imputation settings denote total–missing lengths, measured in days for maritime datasets and tokens for urban datasets.

error. The relative margins depend on the dataset and metric. On DMA AIS, GeoDiffusion's POR advantage over the strongest available baseline increases from 0.0048 for port-to-port pairs to 0.0282 for random pairs, and its Jaccard advantage increases from 0.0117 to 0.0248. This pattern does not hold uniformly across datasets or metrics: for example, on the Private dataset the ADE advantage is 0.063 for port-to-port pairs and 0.047 for random pairs. The results support performance across both endpoint settings without implying that random endpoints consistently yield larger gains.

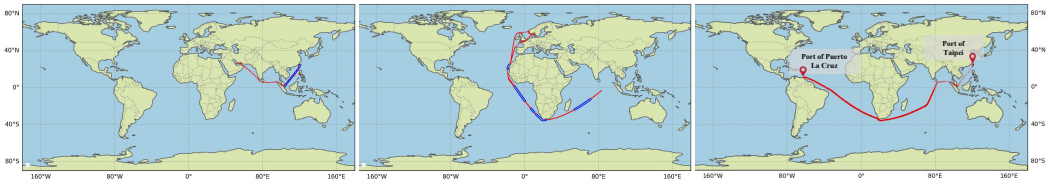


Figure 5: Examples of trajectory prediction (left), trajectory imputation (middle), and route planning from the Port of Taipei to the Port of Puerto La Cruz (right).

Visualization. To further evaluate the modeling capacity of GeoDiffusion, we visualize its outputs on three representative downstream tasks in Figure 5. These examples illustrate generated trajectory shapes across prediction, imputation, and endpoint-conditioned planning. They complement the quantitative comparisons with views of individual model outputs. More visualization results are provided in Appendix A.6.

Ablation Study. Table 2 evaluates component contributions on the Private Dataset under the 60-32 setting. The full model performs best on all four metrics. Removing STOE reduces POR and Jaccard and increases DTW and ADE, supporting the contribution of the combined spatial and temporal offset representation within this architecture. Removing trajectory constraints also worsens all four metrics, and the auto-regressive variant has the lowest scores in this comparison. These ablations support the evaluated components at this setting. Further component comparisons are reported in Appendix A.3.

Table 2: Key ablations on the Private Dataset (60-32 setting). We report Path Overlap Ratio (POR↑), Jaccard Similarity (↑), Dynamic Time Warping (DTW↓), and Average Displacement Error (ADE↓).

Setting	POR	Jaccard	DTW	ADE
Full GeoDiffusion	**0.8463**	**0.7491**	**18.42**	**1.36**
w/o Spatio-Temporal Offset Encoding	0.8122	0.7100	22.87	1.61
w/o Trajectory Constraints	0.8071	0.7103	23.64	1.68
Auto-Regressive Generation	0.7896	0.6932	26.15	1.83

6 Conclusion

We presented GeoDiffusion, a spatio-temporal offset-aware latent diffusion framework for modeling irregular trajectories. By explicitly encoding relative spatial displacements and temporal gaps, GeoDiffusion augments sequence representations with information about the physical scale of local transitions, while latent diffusion captures their global trajectory distribution. A unified posterior-constrained sampling mechanism further enables prediction, imputation, and endpoint-conditioned planning within the same generative framework. Experiments across maritime and urban datasets show favorable aggregate performance in path overlap, sequence alignment, and coordinate accuracy.

AI USE STATEMENT

In this work, we used ChatGPT to assist with drafting and revising parts of the manuscript and with debugging code during implementation. All AI-assisted text and code suggestions were carefully reviewed, verified, and revised by the authors to ensure technical accuracy, methodological correctness, and consistency with the intended presentation of the work. No AI-generated content was used without human verification. The authors take full responsibility for the final content of this work, including the manuscript, implementation, and experimental results.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*, AAAI Technical Report WS-94-03, pp. 359–370. AAAI Press, 1994. URL https://cdn.aaai.org/Workshops/1994/WS-94-03/WS94-03-031.pdf.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 2452–2470, 2023. URL https://proceedings.mlr.press/v202/bilos23a.html.

Samuele Capobianco, Leonardo M Millefiori, Nicola Forti, Paolo Braca, and Peter Willett. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *IEEE Transactions on Aerospace and Electronic Systems*, 57(6):4329–4346, 2021.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. ContiFormer: Continuous-time transformer for irregular time series modeling. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/9328208f88ec69420031647e6ff97727-Abstract-Conference.html.

Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Shang-Ling Hsu, Emmanuel Tung, John Krumm, Cyrus Shahabi, and Khurram Shafique. Trajgpt: Controlled synthetic trajectory generation using a multitask transformer-based spatiotemporal model. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pp. 362–371, 2024.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 4651–4664, 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9902–9915, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Wenzhe Jin, Haina Tang, and Xudong Zhang. STGDPM: Vessel trajectory prediction with spatio-temporal graph diffusion probabilistic model, 2025. URL https://arxiv.org/abs/2503.08065.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2Vec: Learning a vector representation of time, 2019. URL https://arxiv.org/abs/1907.05321.

Changlin Li, Yanglei Gan, Tian Lan, Yuxiang Cai, Xueyi Liu, Run Lin, and Qiao Liu. DiffuTraj: A stochastic vessel trajectory prediction approach via guided diffusion process, 2024. URL https://arxiv.org/abs/2410.09550.

Huanhuan Li and Zaili Yang. Incorporation of ais data-based machine learning into unsupervised route planning for maritime autonomous surface ships. *Transportation Research Part E: Logistics and Transportation Review*, 176:103171, 2023.

Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. Gpt-st: generative pre-training of spatio-temporal graph neural networks. *Advances in neural information processing systems*, 36: 70229–70246, 2023.

Lei Liu, Mingyang Zhang, Cong Liu, Ran Yan, Xiao Lang, and Helong Wang. Shipping map: An innovative method in grid generation of global maritime network for automatic vessel route planning using ais data. *Transportation Research Part C: Emerging Technologies*, 171:105015, 2025.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36: 56998–57025, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2201.09865.

Búgvi Benjamin Magnussen, Nikolaj Bläser, and Hua Lu. Daistin: A data-driven ais trajectory interpolation method. In *Proceedings of the 18th International Symposium on Spatial and Temporal Data*, pp. 75–84, 2023.

Amartya Mukherjee, Ruizhi Deng, He Zhao, Yuzhen Mao, Leonid Sigal, and Frederick Tung. ADiff4TPP: Asynchronous diffusion models for temporal point processes, 2025. URL https://arxiv.org/abs/2504.20411.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042.

Hao Rong, AP Teixeira, and C Guedes Soares. Maritime traffic probabilistic prediction based on ship motion pattern extraction. *Reliability Engineering & System Safety*, 217:108061, 2022.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.10318.

Aivin V Solatorio. Geoformer: Predicting human mobility using generative pre-trained transformer (gpt). In *Proceedings of the 1st International Workshop on the Human Mobility Prediction Challenge*, pp. 11–15, 2023.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

UCI Machine Learning Repository. Taxi service trajectory prediction challenge, ecml pkdd 2015. https://archive.ics.uci.edu/dataset/339/taxi+service+trajectory+prediction+challenge+ecml+pkdd+2015, 2015. Accessed: 2026-05-07.

Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):7940–7953, 2024. doi: 10.1109/TKDE.2024.3460051.

H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th international conference on world wide web*, pp. 1241–1250, 2017.

Zinuo You, Jin Zheng, and John Cartlidge. Latent laplace diffusion for irregular multivariate time series, 2026. URL https://arxiv.org/abs/2605.19805.

Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. Spatio-temporal diffusion point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023. URL https://arxiv.org/abs/2305.12403.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zhiwen Zhang, Zipei Fan, Zewu Lv, Xuan Song, and Ryosuke Shibasaki. Long-term vessel trajectory imputation with physics-guided diffusion probabilistic model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4398–4407, 2024.

Zhiwen Zhang, Wei Yuan, Zipei Fan, Xuan Song, and Ryosuke Shibasaki. Aisfuser: Encoding maritime graphical representations with temporal attribute modeling for vessel trajectory prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 312–321, 2008.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 791–800, 2009.

Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–40, 2010.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Wang-Tao Zhou, Zhao Kang, and Ling Tian. Non-autoregressive diffusion-based temporal point processes for continuous-time long-term event prediction, 2023. URL https://arxiv.org/abs/2311.01033.

Mingda Zhu, Peihua Han, Chunlin Wang, Robert Skulstad, Houxiang Zhang, and Guoyuan Li. An ais data-driven hybrid approach to ship trajectory prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.

Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems*, 36:65168–65188, 2023.

A APPENDIX

A.1 TRAINING AND INFERENCE PROCEDURE

The overall training procedure of GeoDiffusion contains two stages. First, the offset-aware encoder and reconstruction decoder learn a latent representation of trajectories. Second, the latent diffusion model is trained to denoise the offset-aware latent states.

During training, for each trajectory $\mathbf{x}$, we compute the offset sequence $\mathbf{\Delta}(\mathbf{x})$, encode the trajectory into $\mathbf{z}_0$, sample a diffusion timestep (t), and corrupt $\mathbf{z}_0$ with Gaussian noise. The denoising network is then optimized to recover $\mathbf{z}_0$ from the noisy latent state.

During inference, GeoDiffusion starts from Gaussian noise or an intermediate noisy latent state, depending on the task. For unconditional generation, the model directly performs reverse diffusion from $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For constrained tasks, the model initializes constrained latent tokens from the known observations and updates only unconstrained tokens during reverse diffusion. Finally, the decoder maps the generated latent representation back to the trajectory space.

The full sampling process can be summarized as follows:

$$\begin{aligned} \mathbf{z}_T &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \text{for } t = T, \dots, 1: \quad \mathbf{z}_{t-1} &= \mathrm{Denoise}(\mathbf{z}_t, t, \mathbf{\Delta}, \mathbf{H}^c), \\ \mathbf{z}^{\Omega}_{t-1} &\leftarrow \mathrm{Clamp}(\mathbf{z}^{\Omega}_{t-1}, f^{\mathrm{enc}}_{\theta}(\mathbf{x}_{\Omega})), \hat{\mathbf{x}} = f^{\mathrm{dec}}_{\psi}(\mathbf{z}_0). \end{aligned}$$

This algorithm decouples prior learning from task-specific inference. The model learns a general offset-aware trajectory prior during training, while different downstream tasks are specified only through constraint sets during sampling.

A.2 EXPERIMENT DETAILS

Model Configurations. The denoising network in GeoDiffusion is implemented as a 12-layer pre-LayerNorm Transformer with a hidden dimension of 768. The Spatio-Temporal Trajectory Encoder is composed of 2 layers of cross-attention blocks, each equipped with multi-head attention using 8 attention heads. Each cross-attention layer is followed by a feed-forward network with an inner dimension of 1024, residual connections, and LayerNorm applied before both the attention and feed-forward modules.

Training Details. GeoDiffusion is trained using the AdamW optimizer with an initial learning rate of 1e-4 and linear warm-up for the first 10k steps. The diffusion process is discretized into 1000 steps with cosine noise scheduling. Batch size is set to 128, and dropout rate is 0.1. Early stopping is applied based on validation loss.

Evaluation Protocol. Models are evaluated under multiple trajectory forecasting and imputation settings. For each case, we report Path Overlap Ratio (POR) and Jaccard Similarity as main metrics. All reported results are averaged across five independent runs with different random seeds.

A.2.1 PRIVATE DATASET.

GeoDiffusion is pre-trained on a private dataset of tanker AIS records collected from real-world maritime operations. This pre-training dataset covers AIS trajectories of over 3,500 oil tankers from January to June 2022. The AIS messages include position reports, timestamps, speed, heading, and navigational status, sampled at irregular intervals depending on transmission conditions. The vessel trajectories span global shipping routes, encompassing major ocean basins, port approaches, and intercontinental trade corridors.

A.2.2 METRICS.

The POR is defined as the fraction of visited grid cells in the ground-truth trajectory that are also covered by the predicted trajectory: $\text{POR} = \frac{|\text{Predicted Path} \cap \text{Ground Truth Path}|}{|\text{Ground Truth Path}|}$. In addition, we use the

Jaccard Similarity, defined as the ratio between the intersection and union of visited grid sets: $\text{Jaccard} = \frac{|\text{Predicted Path} \cap \text{Ground Truth Path}|}{|\text{Predicted Path} \cup \text{Ground Truth Path}|}$.

Dynamic Time Warping (DTW). DTW measures similarity between coordinate sequences by finding an order-preserving alignment that permits local stretching and compression of sequence indices. The distance below compares spatial coordinates; it does not explicitly include elapsed time.

Given a ground-truth trajectory $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$ and a generated trajectory $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_{T'}\}$, where each point $\mathbf{y}_i$ or $\hat{\mathbf{y}}_j$ denotes a two-dimensional coordinate, DTW computes a cumulative alignment cost through dynamic programming:

$$D(i,j) = d(\mathbf{y}_i, \hat{\mathbf{y}}_j) + \min\left\{D(i-1,j), D(i,j-1), D(i-1,j-1)\right\}, \tag{28}$$

where $d(\mathbf{y}_i, \hat{\mathbf{y}}_j)$ is the Euclidean distance between two trajectory points:

$$d(\mathbf{y}_i, \hat{\mathbf{y}}_j) = \left\|\mathbf{y}_i - \hat{\mathbf{y}}_j\right\|_2. \tag{29}$$

The final DTW distance is given by:

$$\text{DTW}(\mathbf{Y}, \hat{\mathbf{Y}}) = D(T, T'). \tag{30}$$

A lower DTW value indicates closer alignment of the coordinate sequences.

Average Displacement Error (ADE). ADE evaluates the average coordinate-level displacement error between a generated trajectory and the corresponding ground-truth trajectory. It is a standard metric for trajectory prediction and directly measures the pointwise spatial accuracy of the generated path.

Given a ground-truth trajectory $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$ and a generated trajectory $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T\}$ with the same length, ADE is defined as:

$$\text{ADE} = \frac{1}{T}\sum_{t=1}^{T}\left\|\hat{\mathbf{y}}_t - \mathbf{y}_t\right\|_2. \tag{31}$$

For two-dimensional geographic coordinates, this can be written as:

$$\text{ADE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(\widehat{\text{lon}}_t - \text{lon}_t)^2 + (\widehat{\text{lat}}_t - \text{lat}_t)^2}. \tag{32}$$

A lower ADE value indicates that the generated trajectory is closer to the ground-truth trajectory at the coordinate level. While POR and Jaccard focus on route-level spatial overlap, ADE provides a more direct measure of pointwise prediction accuracy.

A.2.3 Additional Experiment Results

Trajectory Prediction Results. Table 3 reports Jaccard similarity for trajectory prediction across five datasets. GeoDiffusion achieves the highest score in 23 of the 25 settings, indicating greater overlap with the reference trajectories' visited grid sets in those comparisons.

On DMA AIS and U.S. Coast, GeoDiffusion has the highest Jaccard score at every evaluated prediction setting. On the Private dataset, it leads at 17-8, 23-12, and 40-20, while competing methods lead at 10-5 and 30-15. These results show that the aggregate advantage includes setting-specific exceptions.

GeoDiffusion also has the highest Jaccard score at every evaluated setting on Geolife and Porto Taxi. These results extend the evaluation to urban human mobility and taxi trajectories, in addition to maritime AIS data.

Jaccard trends vary across datasets and methods as the observed–future window changes. Several curves decrease, while the Private dataset includes increases between adjacent settings. GeoDiffusion has the highest mean Jaccard across the five settings within each dataset.

Detailed Prediction Analysis. Figure 6 provides a detailed comparison of trajectory prediction performance on five datasets under different forecasting lengths. Overall, GeoDiffusion achieves

Table 3: Jaccard results on five datasets under different forecasting lengths.

Tasks		GeoDiffusion	Informer	Autoformer	TimesNet	Crossformer	GPT-ST	AISFuser	AIS-Hybrid
Private Dataset	10-5	0.770	0.805	0.785	**0.820**	0.728	0.760	0.785	0.798
	17-8	**0.785**	0.780	0.710	0.720	0.735	0.750	0.710	0.698
	23-12	**0.820**	0.735	0.705	0.713	0.740	0.680	0.692	0.670
	30-15	0.735	0.657	0.678	0.775	0.795	**0.800**	0.742	0.710
	40-20	**0.800**	0.700	0.720	0.728	0.750	0.702	0.702	0.664
DMA AIS Dataset	10-5	**0.890**	0.872	0.864	0.877	0.852	0.870	0.858	0.850
	17-8	**0.865**	0.845	0.835	0.840	0.828	0.810	0.830	0.805
	23-12	**0.852**	0.825	0.812	0.822	0.835	0.812	0.820	0.792
	30-15	**0.828**	0.805	0.800	0.812	0.805	0.790	0.798	0.780
	40-20	**0.812**	0.780	0.775	0.785	0.790	0.770	0.765	0.748
U.S. Coast Dataset	10-5	**0.870**	0.840	0.825	0.835	0.832	0.850	0.830	0.820
	17-8	**0.840**	0.815	0.800	0.810	0.805	0.798	0.785	0.772
	23-12	**0.825**	0.800	0.785	0.795	0.785	0.780	0.770	0.755
	30-15	**0.805**	0.780	0.770	0.775	0.772	0.765	0.758	0.748
	40-20	**0.785**	0.755	0.745	0.752	0.748	0.742	0.735	0.722
Geolife	10-5	**0.846**	0.817	0.804	0.826	0.812	0.820	0.798	0.786
	17-8	**0.818**	0.790	0.778	0.798	0.784	0.787	0.768	0.754
	23-12	**0.796**	0.766	0.754	0.775	0.765	0.758	0.742	0.730
	30-15	**0.775**	0.742	0.733	0.752	0.748	0.736	0.720	0.708
	40-20	**0.752**	0.718	0.710	0.728	0.722	0.710	0.696	0.682
Porto Taxi	10-5	**0.834**	0.804	0.792	0.812	0.798	0.808	0.782	0.770
	17-8	**0.806**	0.776	0.764	0.784	0.772	0.776	0.752	0.740
	23-12	**0.784**	0.750	0.740	0.760	0.752	0.746	0.726	0.714
	30-15	**0.762**	0.728	0.718	0.738	0.732	0.722	0.704	0.692
	40-20	**0.738**	0.704	0.694	0.710	0.708	0.698	0.680	0.668

lower ADE and higher POR than most competing methods across both maritime AIS datasets and urban mobility datasets. Since ADE measures coordinate-level forecasting error and POR evaluates route-level spatial overlap, these results indicate that GeoDiffusion improves both pointwise prediction accuracy and path-level fidelity.

The detailed prediction curves vary across datasets as the observed–future window increases. Some scores worsen, while others improve between adjacent settings. These comparisons characterize performance at the evaluated windows rather than a uniform degradation law.

On the maritime datasets, GeoDiffusion consistently maintains strong POR values, showing that its predictions better preserve realistic navigation corridors and spatial movement patterns. It also achieves lower ADE in most forecasting settings, indicating more accurate coordinate-level future trajectory estimation. These results suggest that the proposed offset-aware representation helps capture heterogeneous spatial displacements and temporal gaps in AIS trajectories, which is particularly important for long-horizon maritime forecasting.

On Geolife and Porto Taxi, GeoDiffusion also performs favorably on ADE and POR relative to the displayed baselines. These comparisons extend the evaluation from maritime trajectories to urban mobility.

Overall, the detailed prediction results complement the averaged metrics by showing coordinate errors and visited-grid coverage separately for each dataset.

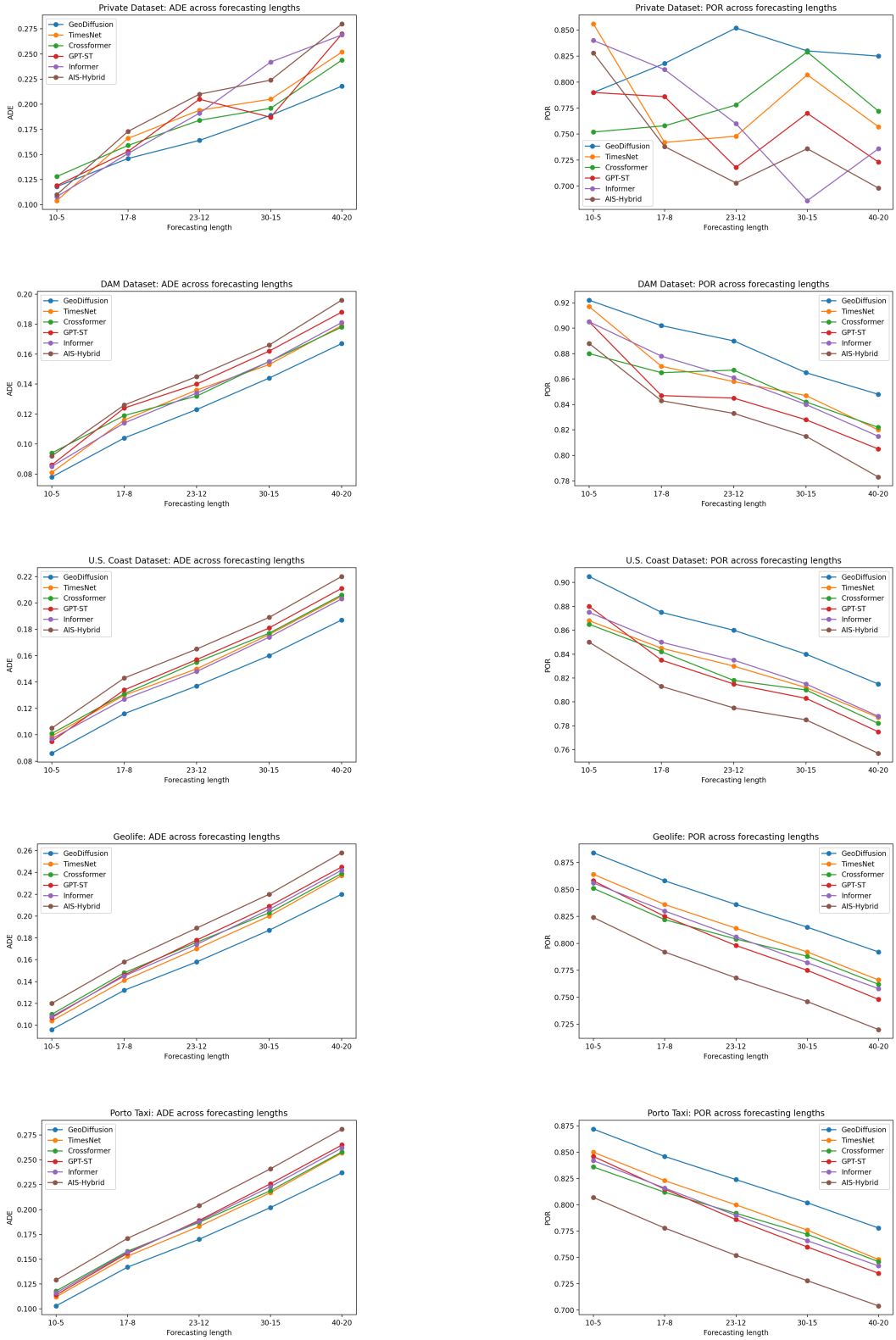


Figure 6: Detailed trajectory prediction results on five datasets under different forecasting lengths. For each dataset, we visualize ADE (↓) and POR (↑) across increasing prediction horizons. The panels compare coordinate error and visited-grid coverage for the displayed methods.

Table 4: Jaccard results for trajectory imputation on five datasets under different missing patterns.

Tasks		GeoDiffusion	Informer	Autoformer	TimesNet	Crossformer	GPT-ST	DAISTIN	PG-Diffusion
Private Dataset	10-5	0.685	0.716	**0.740**	0.610	0.615	0.605	0.710	0.705
	17-8	**0.725**	0.650	0.620	0.653	0.710	0.668	0.668	0.670
	23-12	**0.713**	0.665	0.625	0.703	0.665	0.648	0.644	0.630
	30-15	0.710	0.610	0.700	0.595	0.673	0.618	**0.715**	0.625
	40-20	**0.760**	0.605	0.668	0.625	0.740	0.635	0.618	0.690
DMA AIS Dataset	10-5	**0.812**	0.792	0.808	0.765	0.783	0.760	0.765	0.775
	17-8	**0.772**	0.755	0.738	0.753	0.750	0.738	0.743	0.760
	23-12	**0.765**	0.740	0.723	0.738	0.733	0.725	0.712	0.720
	30-15	**0.785**	0.725	0.758	0.720	0.750	0.723	0.740	0.743
	40-20	**0.800**	0.715	0.740	0.715	0.775	0.728	0.722	0.738
U.S. Coast Dataset	10-5	**0.798**	0.785	**0.798**	0.758	0.772	0.755	0.760	0.770
	17-8	**0.775**	0.750	0.728	0.753	0.747	0.735	0.743	0.757
	23-12	**0.760**	0.730	0.715	0.730	0.723	0.715	0.705	0.712
	30-15	**0.765**	0.710	0.742	0.705	0.738	0.708	0.725	0.730
	40-20	**0.785**	0.700	0.730	0.705	0.765	0.712	0.710	0.725
Geolife	10-5	**0.824**	0.798	0.790	0.802	0.795	0.782	0.768	0.780
	17-8	**0.800**	0.770	0.758	0.774	0.765	0.755	0.748	0.760
	23-12	**0.782**	0.748	0.735	0.750	0.742	0.730	0.720	0.737
	30-15	**0.768**	0.725	0.713	0.730	0.722	0.708	0.700	0.718
	40-20	**0.755**	0.705	0.695	0.713	0.705	0.688	0.682	0.698
Porto Taxi	10-5	**0.805**	0.780	0.768	0.785	0.776	0.765	0.750	0.762
	17-8	**0.782**	0.755	0.742	0.760	0.748	0.738	0.728	0.745
	23-12	**0.765**	0.732	0.720	0.735	0.727	0.712	0.702	0.722
	30-15	**0.750**	0.710	0.700	0.715	0.708	0.692	0.685	0.705
	40-20	**0.738**	0.692	0.682	0.698	0.690	0.675	0.668	0.688

Trajectory Imputation Results. Table 4 reports the Jaccard Similarity results for trajectory imputation across five datasets under different missing patterns. Overall, GeoDiffusion achieves the best or tied-best performance in most settings, demonstrating stronger path-level reconstruction ability than both general sequence models and imputation-specific baselines. Since Jaccard measures the overlap between the generated and ground-truth visited grid sets, higher scores indicate that the imputed trajectories recover more consistent spatial regions and route structures.

On DMA AIS and U.S. Coast, GeoDiffusion achieves the highest or tied-highest Jaccard score across the evaluated imputation settings; it ties Autoformer at 0.798 on U.S. Coast 10-5. On the Private dataset, it leads in three of five settings and trails the strongest baseline at 10-5 and 30-15.

On Geolife and Porto Taxi, GeoDiffusion achieves the highest Jaccard score at every evaluated imputation setting. These comparisons show favorable visited-grid overlap on both urban datasets.

Overall, the Jaccard results measure recovery of the spatial regions visited by the reference trajectories. The POR, DTW, and ADE comparisons provide complementary coverage, sequence-alignment, and coordinate-error measurements.

Detailed Imputation Analysis. Figure 7 compares ADE and POR across five datasets. The total–missing settings vary both trajectory length and the number of missing observations. They do not form a monotonic missing-rate sweep: the missing fraction stays near one half and the number of observed points increases. Performance should therefore be compared within each setting rather than interpreted as a response to progressively fewer observations.

On the maritime datasets, the displayed ADE and POR curves compare coordinate reconstruction and visited-grid coverage across the evaluated missing patterns. GeoDiffusion performs favorably in most comparisons, with dataset-specific variation across settings.

On Geolife and Porto Taxi, GeoDiffusion maintains favorable ADE and POR relative to the displayed baselines. These results show that the evaluated model performs well on urban trajectory imputation as well as on maritime data.

The detailed curves complement the averaged results by showing how performance varies across datasets and settings. Together with the Jaccard tables, they characterize the spatial coverage and coordinate reconstruction achieved by the complete model.

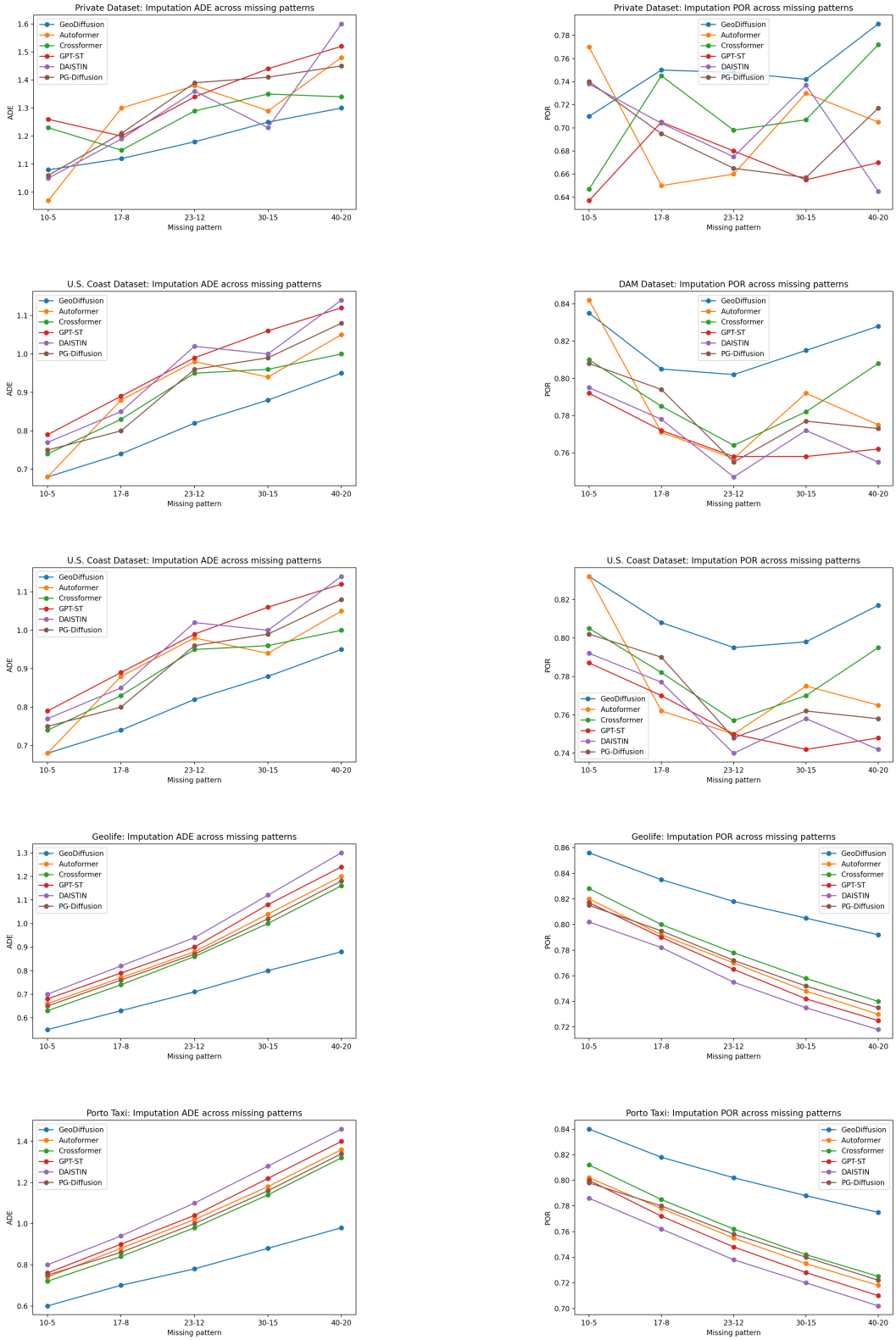


Figure 7: Detailed trajectory imputation results on five datasets under different missing patterns. The left column reports ADE (↓), which measures coordinate-level reconstruction error, and the right column reports POR (↑), which measures path-level spatial overlap. The panels compare coordinate reconstruction and visited-grid coverage across the evaluated total–missing settings.

A.3 Additional Ablation Results

Table 5 reports additional component comparisons. Replacing the Transformer denoiser with a convolutional network reduces POR and Jaccard by approximately 5.18% and 5.58%, respectively, relative to the full model. Removing latent query cross-attention also lowers both scores. These results support the contributions of the evaluated denoiser and latent aggregation components at this setting.

Table 5: Extended ablation results on the trajectory prediction task (Private Dataset, 60-32 setting).

Setting	POR	Jaccard
w/o Transformer (ConvNet Denoiser)	0.8025	0.7073
w/o Latent Query Cross-Attention	0.8250	0.7234

A.4 GeoDiffusion and Baseline Implementation Details

A.4.1 GeoDiffusion Implementation

Grid-based Trajectory Representation. To enable stable training and effective modeling, we discretize the original continuous-space trajectories, represented as sequences of latitude-longitude coordinates, into a finite sequence of grid-based tokens, as shown in Figure 8.

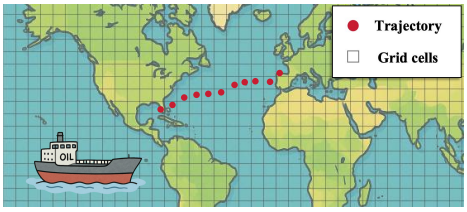


Figure 8: Grid-based representation of vessel trajectories. Each longitude–latitude coordinate is mapped to a discrete grid cell.

This gridification not only transforms the trajectory into a manageable, enumerable representation but also preserves local spatial structure through the design of grid ID assignments, facilitating the learning of transition patterns between neighboring regions. Specifically, each record consists of a longitude–latitude coordinate and a timestamp. We can utilize any spatial discretization function to map each geographic coordinate to a unique grid cell ID.

We implement GeoDiffusion using PyTorch and train it on a single NVIDIA A40 GPU with 48 GB of memory. The model consists of three main components: a Vessel Trajectory Encoder, a Transformer-based denoising network, and a trajectory reconstruction decoder. The encoder processes AIS trajectories represented as sequences of spatio-temporal tokens $\mathbf{x} = \{s_1, s_2, \ldots, s_N\}$, where each token $s_i = (\text{grid}_i, \tau_i)$ combines a grid cell ID and timestamp. Grid tokens are embedded into 512-dimensional vectors, and we introduce a Spatio-Temporal Offset Encoding mechanism to model relative spatial and temporal differences between adjacent tokens. The resulting features are fused with frozen BART encoder outputs via cross-attention, yielding a latent representation $\mathbf{z}_0 \in \mathbb{R}^{k\times 512}$, with $k = 16$. The denoising network is a 12-layer Transformer trained using the DDPM framework with a cosine noise schedule over $T = 1000$ steps. We use a standard noise-prediction objective with a time-weighting term. After denoising, the trajectory decoder transforms the final latent $\hat{\mathbf{z}}_0$ into grid token predictions via an MLP and softmax, which are mapped back to geographic coordinates. We train the model with a batch size of 128, learning rate of 1e-4, AdamW optimizer, weight decay of 0.01, and dropout of 0.1. Gradient clipping (1.0) and linear learning rate warm-up over the first 10k steps are applied. Task-specific conditioning mechanisms are used: for prediction, the known prefix is fixed; for imputation, masked tokens are denoised while observed ones remain fixed; for planning, only endpoints are fixed while intermediate tokens are generated. Training on the private dataset takes roughly 2 days, and inference runs at under 50 ms per trajectory using 50 denoising steps. All results are averaged across five independent runs with different random seeds.

A.4.2 BASELINES

To evaluate the effectiveness of our method across different tasks, we compare GeoDiffusion with a comprehensive set of baselines, categorized according to the nature of each task.

Trajectory Prediction. We select five general-purpose state-of-the-art sequence modeling methods widely used in time-series forecasting and spatio-temporal prediction: Informer Zhou et al. (2021), Autoformer Wu et al. (2021), TimesNet Wu et al. (2023), Crossformer Zhang & Yan (2023), and GPT-ST Li et al. (2023). These methods are adapted to our grid-based trajectory setting by embedding each token into the temporal modeling pipeline. Additionally, we include two task-specific trajectory prediction models: AISFuser Zhang et al. (2025) and AIS-Hybrid Zhu et al. (2024).

Trajectory Imputation. We use the same five general-purpose baselines for sequence modeling as in the prediction task. To account for methods explicitly designed for trajectory completion, we further compare with DAISTIN Magnussen et al. (2023) and PG-Diffusion Zhang et al. (2024).

Route Planning. Due to the structured and goal-directed nature of the route planning task, we select two diffusion-based generative models as general baselines: DiffTraj Zhu et al. (2023) and Diff-RNTraj Wei et al. (2024). We also include two maritime domain planning models: ShippingMap Liu et al. (2025) and Incorporation Li & Yang (2023).

A.5 BASELINE IMPLEMENTATION

Informer Zhou et al. (2021) employs ProbSparse self-attention to improve the efficiency of long sequence modeling and has been widely adopted for time series forecasting tasks. To implement Informer, we utilize their open-source code, available at https://github.com/zhouhaoyi/Informer2020. Following the original configuration, we use the encoder-decoder variant with 3 encoder layers and 2 decoder layers, and set the embedding dimension to 512. The total number of parameters is about 18.36M. We set the number of training epochs to 50, the learning rate to 0.001, and the batch size to 128. Input and prediction sequence lengths are aligned with our grid-based trajectory forecasting protocol, and grid IDs are embedded using a shared trainable embedding layer.

Autoformer Wu et al. (2021) designs a decomposition-based forecasting framework that captures temporal trends and seasonalities. We implement Autoformer using their official repository, available at https://github.com/thuml/Autoformer. We adopt the 3-layer encoder-only variant with an embedding dimension of 512, which shows robust performance in the original paper. The total model size is 19.45M. We set the number of training epochs to 50, the learning rate to 0.001, and the batch size to 128. During preprocessing, each grid token is mapped to a learned embedding and fed into the temporal encoder for sequence-to-sequence forecasting.

TimesNet Wu et al. (2023) models time series data in a 2D variation space to capture local and global temporal dynamics. We follow their official implementation at https://github.com/thuml/TimesNet. Based on the default configuration, we use 3 convolutional blocks with an embedding size of 512 and keep all other hyperparameters unchanged. The total model size is 20.91M. We train the model for 50 epochs with a learning rate of 0.001 and a batch size of 128. We reshape the trajectory grid token sequence as a univariate time series and adapt TimesNet to handle both forecasting and imputation by applying masking strategies consistent with our task setup.

Crossformer Zhang & Yan (2023) introduces cross-dimension dependency modeling for multivariate time series forecasting. We implement Crossformer using the official codebase at https://github.com/Thinklab-SJTU/Crossformer. Following the base configuration, we use 2 cross blocks with an embedding dimension of 512 and maintain all default hyperparameters. The total parameter size is 17.02M. The model is trained for 50 epochs with a learning rate of 0.001 and a batch size of 128. For input adaptation, each trajectory is treated as a multivariate sequence by stacking spatial and temporal token embeddings along the feature axis, enabling Crossformer to model both spatial locality and temporal correlation.

GPT-ST Li et al. (2023) proposes a generative pretraining framework for spatio-temporal graph neural networks, utilizing masked token prediction to capture complex spatial-temporal dependencies. To implement GPT-ST in our trajectory modeling setting, we adopt their official codebase, available at https://github.com/HKUDS/GPT-ST. Since GPT-ST was originally designed for traffic flow data over road networks, we adapt it to our grid-based maritime trajectories by constructing spatial graphs over

grid cells. Specifically, we define undirected edges between adjacent grid cells based on 8-neighbor spatial adjacency, and construct trajectory-induced temporal graphs where each node corresponds to a grid cell at a specific time step.

Following the encoder-only variant of GPT-ST, we treat each trajectory as a sequence of graph node tokens and apply masked modeling to train the model to recover randomly masked grid IDs. For trajectory prediction, the model is trained to predict future tokens given partial input, and for imputation, we randomly mask segments and train the model to fill in the missing positions. We use a 3-layer transformer encoder with hidden size 512, consistent with the original implementation. The total model size is 21.73M. We train GPT-ST for 50 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 128. Masking ratios and sampling strategies follow the default settings from the official implementation.

AISFuser Zhang et al. (2025) is a state-of-the-art transformer-based framework that models maritime graphical representations alongside multi-modal temporal dynamics for vessel trajectory prediction. It integrates both spatial constraints from a maritime traffic network and temporal heterogeneity via a self-supervised learning (SSL) module. To implement AISFuser, we follow the original settings and utilize the publicly available AIS dataset provided by the Danish Maritime Authority. The encoder includes 6 transformer layers with 8 attention heads, and the AIS data is discretized with a spatial resolution of 0.01° for LAT/LON and embedded into a 256-dimensional latent space. Maritime graphical structures (e.g., ship lanes and landscape) within a 50 km radius are encoded using an attention-based polyline graph module. We train the model for 50 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Predictions are made over a variable forecast horizon of up to 10 hours. The SSL loss weight is set to 0.1. We use the official code provided by the authors and evaluate the model using MAE and RMSE metrics on the discretized grid-based trajectory outputs.

AIS-Hybrid Zhu et al. (2024) integrates a seamanship-informed encounter classification module with a two-branch hybrid predictor combining kinematics-based and neural network (NN)-based models. To implement AIS-Hybrid, we follow the methodology described in the original paper and reproduce the two-stage pipeline: an encounter classification phase based on TCPA and COG distributions, followed by a trajectory predictor. For prediction, the NN branch uses a single-layer LSTM encoder-decoder with 128 hidden units and a fully connected output layer, trained for 400–1000 epochs depending on encounter type. The kinematics branch uses historical AIS data and selects a best-matching trajectory via DTW-based similarity retrieval. The final prediction is obtained via an adaptive weighted fusion of the two branches based on real-time error feedback. We discretize AIS positions into grid IDs and resample the data to a 1-second interval to match our grid-based prediction format. Evaluation follows the original paper's metrics, using Mean Prediction Error (MPE) computed over a 3–9 minute prediction horizon. The model is trained and tested on the Oslofjord AIS dataset provided in the original study.

DAISTIN Magnussen et al. (2023) is a data-driven AIS trajectory interpolation method that reconstructs vessel trajectories by searching shortest paths over a directed graph constructed from massive historical AIS data. To implement DAISTIN, we follow the original configuration described in the paper and reproduce the full pipeline, including digraph construction, interpolation via A* pathfinding, and postprocessing. The graph is built from AIS trajectory segments after geometric sampling using a spatial density-aware strategy with 50 quantiles and a sampling rate of 910^6 data points. Each AIS node encodes latitude, longitude, and heading, and edge connections are determined by a spatial neighborhood radius of 20 km and a direction threshold of 45 degrees. During inference, signal gaps exceeding this radius are interpolated via shortest paths, and Douglas-Peucker postprocessing is applied to reduce zigzag noise. For consistency with our grid-based formulation, interpolated positions are mapped to discrete grid IDs. We use the optimal hyperparameter settings reported in the original paper and evaluate DAISTIN using the same SPD and HD metrics. The model is tested on the same IHS Markit tanker AIS dataset used in the original study, filtered to match our imputation protocol with at least 20% missing samples in each trajectory.

PG-Diffusion Zhang et al. (2024) introduces a physics-guided diffusion probabilistic framework designed specifically for long-term vessel trajectory imputation. To implement PG-Diffusion, we follow the original setting and use the official AIS dataset provided by the Danish Maritime Authority, filtered for cargo and tanker vessels. We adopt the original architecture, which includes a pre-trained variational trajectory embedding module (VRNN), a transformer encoder for modeling observed

trajectories, and a Traj-UNet denoising backbone with physics-based constraints. The transformer encoder is configured with 3 layers, 4 attention heads, and 128 hidden units. The number of diffusion time steps is set to 500. During training, we incorporate sinusoidal timestep embeddings and vessel/time ID embeddings into the noise prediction module. The physics-guided discriminator, active only during training, enforces kinematic consistency between consecutive imputed points via Earth motion equations. We set the discriminator step to 20, and the hyperparameters for physics loss terms β_1 and β_2 are set to 0.2 and 0.1, respectively. The model is trained for 50 epochs using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 512. For compatibility with our grid-based formulation, all imputed GPS coordinates are mapped to discrete grid IDs before evaluation.

DiffTraj Zhu et al. (2023) introduces a spatial-temporal diffusion probabilistic model for realistic GPS trajectory generation. To implement DiffTraj, we utilize the official PyTorch implementation provided by the authors, available at https://github.com/Yasoz/DiffTraj. The model includes a forward Gaussian noise process and a reverse denoising process, parameterized by the Traj-UNet architecture with 4 downsampling and upsampling blocks and 2 ResNet blocks per stage. Conditional generation is enabled using a Wide & Deep module that embeds contextual inputs such as departure region and time. To adapt the model for our Route Planning task, we explicitly provide start and end grid tokens as part of the conditional input. The model is trained using a 500-step diffusion schedule with a batch size of 1024 and learning rate of 1×10^{-4}, following the original setting. For evaluation, generated trajectories are decoded into discrete grid sequences and compared against ground truth using Grid Edit Distance (GED) and Path Overlap Ratio (POR) metrics.

Diff-RNTraj Wei et al. (2024) proposes a structure-aware diffusion model designed for road network-constrained trajectory generation. To implement Diff-RNTraj, we follow the authors' official implementation available at https://github.com/wtl52656/Diff-RNTraj. The model consists of three main components: a hybrid RNTraj vectorization module that embeds discrete road segments (via pre-trained Node2vec representations) and continuous movement ratios into a continuous space; a continuous diffusion model trained over 500 denoising steps; and a RNTraj decoder that reconstructs road segment sequences and locations from the generated latent representations. For our Route Planning task, we adapt the conditional sampling pipeline by inserting the specified origin and destination road segments into the diffusion process as soft positional anchors. The denoising backbone is built with 10 residual dilation convolution layers (RDCLs) with kernel size 3 and 512 channels, and we use a batch size of 256, a learning rate of 0.001, and the Adam optimizer. All generated trajectories are decoded to grid-based sequences and evaluated using Grid Edit Distance (GED) and Path Overlap Ratio (POR) against ground-truth routes.

ShippingMap Liu et al. (2025) introduces a hybrid data-driven and model-based framework for global vessel route planning by constructing a static grid-based maritime network from large-scale AIS data. To implement ShippingMap, we follow the original four-step process: AIS trajectory segmentation using spatiotemporal thresholds, clustering of berthing and waypoint areas via CKBA-DBSCAN, maritime shipping network construction using historical trajectory arcs, and double-layer A*-based route search. Specifically, we partition the global water surface into a 1800900 grid system (0.2° resolution) and construct a directed grid network based on historical navigation. Waypoints are detected via the PELT change point algorithm and grouped using directional-aware DBSCAN clustering. Given a source and destination port, we map them to corresponding grid cells and apply A* search on the combined shipping and grid network to retrieve the optimal route. The code is implemented in Python and evaluated on the global 2018 AIS dataset of dry bulk carriers. To align with our grid-based representation, all planned routes are discretized into sequences of grid tokens. We adopt Grid Edit Distance (GED) and Path Overlap Ratio (POR) as evaluation metrics for comparing the planned routes with ground-truth trajectories.

AIS-MASS Li & Yang (2023) proposes an unsupervised route planning framework for Maritime Autonomous Surface Ships (MASS) based on historical AIS data. To implement AIS-MASS, we follow the original methodology, which consists of two key components: movement pattern extraction via Automatic and Adaptive Dynamic Time Warping (AADTW) and unsupervised clustering using the Spectral Clustering with Affinity Feature (SCAF) algorithm. Movement patterns are mined separately for different ship types (e.g., tankers and cargo vessels) and are used to construct representative route libraries. For each planning instance, given a source and destination location, the algorithm identifies the nearest feature centers in the cluster space and retrieves the corresponding optimal pattern route. The system is fully data-driven and does not require any human-labeled samples. In our adaptation

to the grid-based Route Planning setting, we map all AIS coordinates to discrete grid tokens and apply the AADTW+SCAF pipeline to extract cluster-wise centroid routes. During inference, we use a nearest-cluster search based on source-destination grid alignment and evaluate the retrieved routes using Grid Edit Distance (GED) and Path Overlap Ratio (POR). All hyperparameters (e.g., clustering affinity weights and DTW window size) follow the settings recommended in the original implementation.

Evaluation Metrics. We adopt two structure-aware similarity metrics: *Path Overlap Ratio (POR)* and *Jaccard Similarity*, to evaluate the visited-grid overlap of the generated trajectories. The *Path Overlap Ratio (POR)* is defined as the fraction of visited grid cells in the ground-truth trajectory that are also covered by the predicted trajectory: $\text{POR} = \frac{|\text{Predicted Path} \cap \text{Ground Truth Path}|}{|\text{Ground Truth Path}|}$. This metric measures coverage of the visited grid cells in the reference trajectory. In addition, we use the *Jaccard Similarity*, defined as the ratio between the intersection and union of visited grid sets: $\text{Jaccard} = \frac{|\text{Predicted Path} \cap \text{Ground Truth Path}|}{|\text{Predicted Path} \cup \text{Ground Truth Path}|}$. Unlike POR, the Jaccard index penalizes both under-coverage and over-coverage, thus providing a balanced assessment of *path precision* and *path recall*. Together, these two metrics capture both *coverage sufficiency* and *visited-grid overlap*, offering a comprehensive view of performance in grid-based maritime route planning.

A.6 MORE VISUALIZATION RESULTS

This section presents qualitative examples separately from the quantitative settings in the result tables. Red denotes observed trajectory points, and blue denotes model-generated portions. The prediction examples use a fixed five-day target with different observed-history lengths. The imputation examples are organized into five display groups; these groups do not specify the total–missing lengths or missing fractions used in the quantitative evaluation.

Imputation examples, group A.

Figure 10 shows eight examples of observed and generated trajectory portions across different geographic regions.

Prediction: 10 observed days and 5 target days.

Figure 11 shows eight five-day continuations following 10 days of observed history.

Imputation examples, group B.

Figure 12 shows eight examples of observed and generated trajectory portions across different geographic regions.

Prediction: 20 observed days and 5 target days.

Figure 13 shows eight five-day continuations following 20 days of observed history.

Imputation examples, group C.

Figure 14 shows eight examples of observed and generated trajectory portions across different geographic regions.

Prediction: 29 observed days and 5 target days.

Figure 15 shows eight five-day continuations following 29 days of observed history.

Imputation examples, group D.

Figure 16 shows eight examples of observed and generated trajectory portions across different geographic regions.

Prediction: 38 observed days and 5 target days.

Figure 17 shows eight five-day continuations following 38 days of observed history.

Imputation examples, group E.

Figure 18 shows eight examples of observed and generated trajectory portions across different geographic regions.

Prediction: 55 observed days and 5 target days.

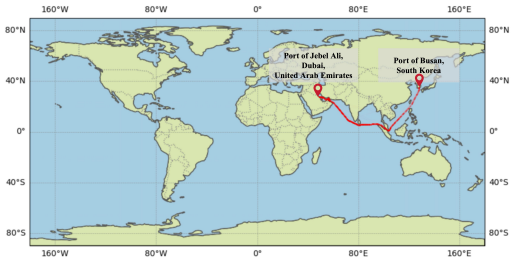
Figure 19 shows eight five-day continuations following 55 days of observed history.

Port-to-port planning.

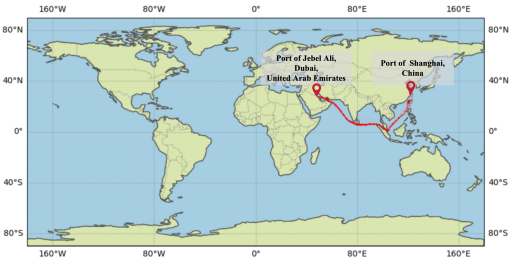
Figure 9 shows generated routes conditioned on origin and destination ports.

A.7 LIMITATIONS

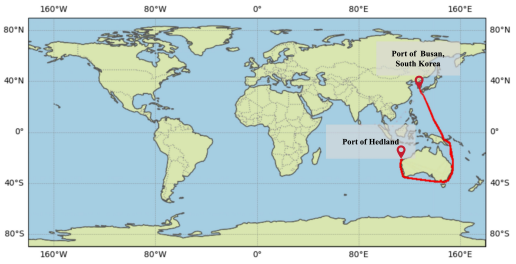
GeoDiffusion has several limitations. First, although we evaluate the model on both maritime AIS and urban mobility datasets, the experiments are still centered on trajectory data. Further validation on other irregular spatio-temporal domains, such as sensor networks, clinical time series, and physical systems, would better demonstrate the generality of the framework. Second, the current implementation relies on spatial tokenization, making the results sensitive to grid resolution and tokenization design. Third, posterior-constrained sampling enforces observed or specified trajectory constraints, but does not explicitly model external physical rules, traffic regulations, or dynamic environmental factors. Finally, diffusion-based inference requires iterative denoising, which is more computationally expensive than single-pass forecasting models. Improving constraint modeling and sampling efficiency are important directions for future work.



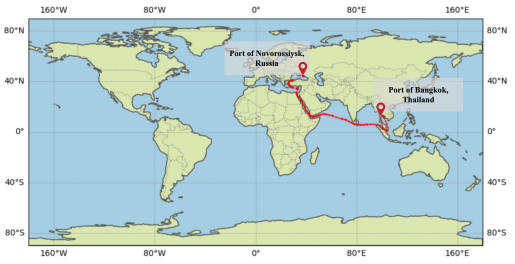
(a) Port of Jebel Ali, Dubai, United Arab Emirates to Busan, South Korea



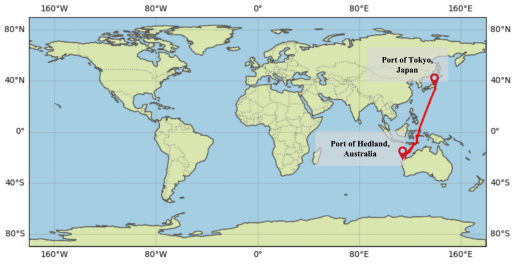
(b) Port of Jebel Ali, Dubai, United Arab Emirates to Shanghai, China



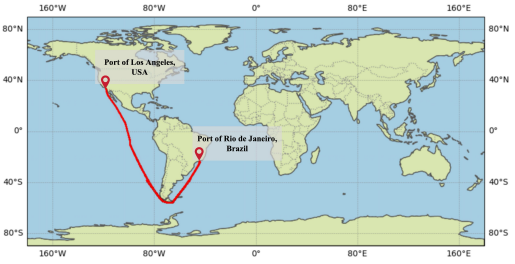
(c) Port of Hedland to Busan, South Korea



(d) Port of Bangkok, Thailand to Novorossiysk, Russia



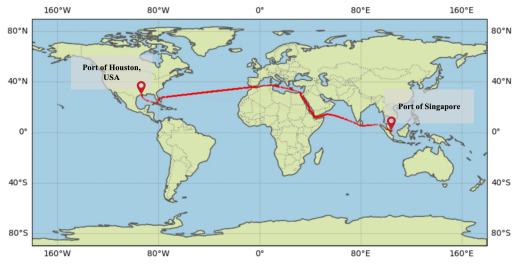
(e) Port of Hedland, Australia to Tokyo, Japan



(f) Port of Los Angeles, USA to Rio de Janeiro, Brazil

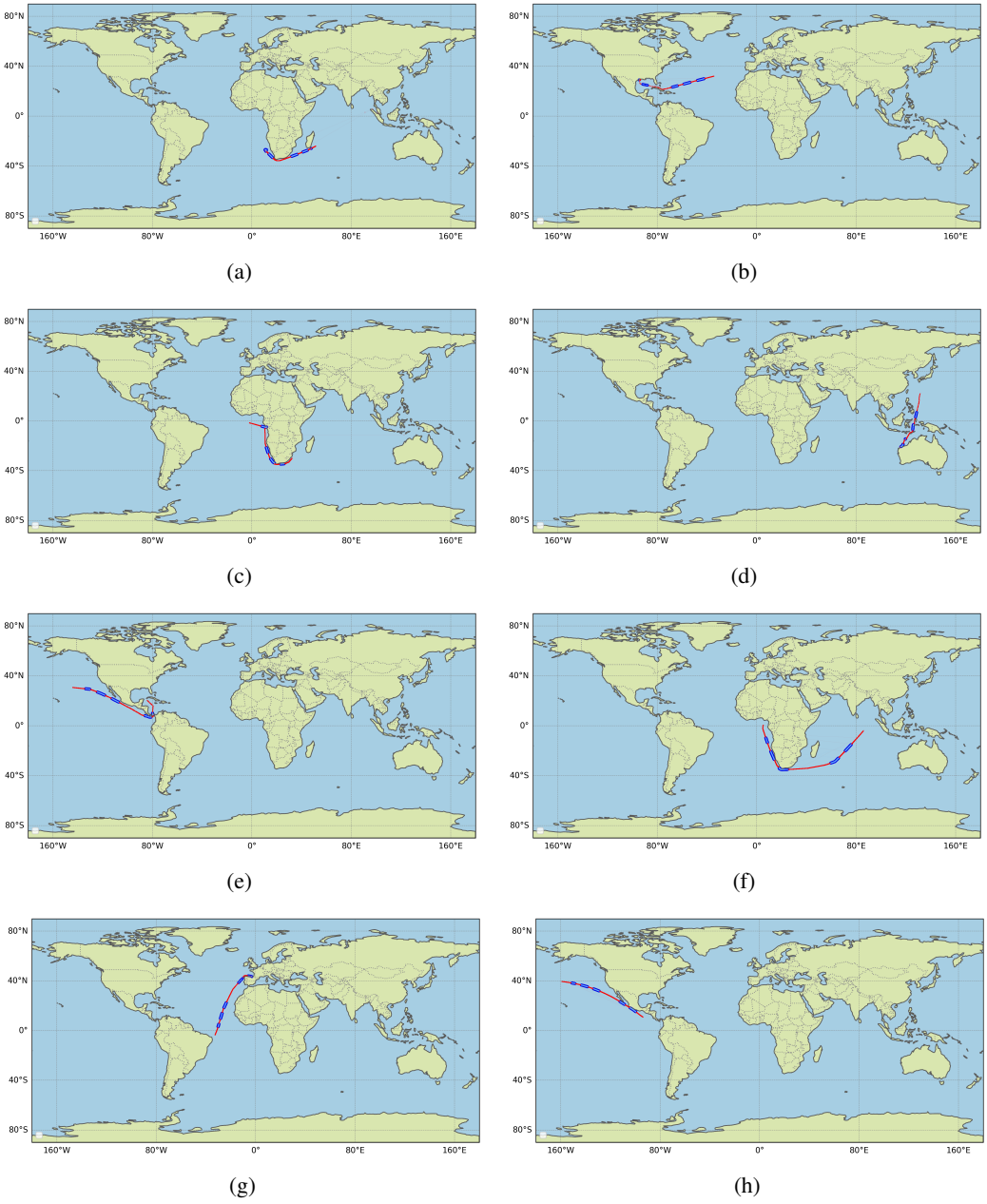


(g) Port of Singapore to Corpus Christi, USA



(h) Port of Singapore to Houston, USA

Figure 9: Port to Port



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 10: Imputation examples, group A. Red: observed points; blue: generated portions.

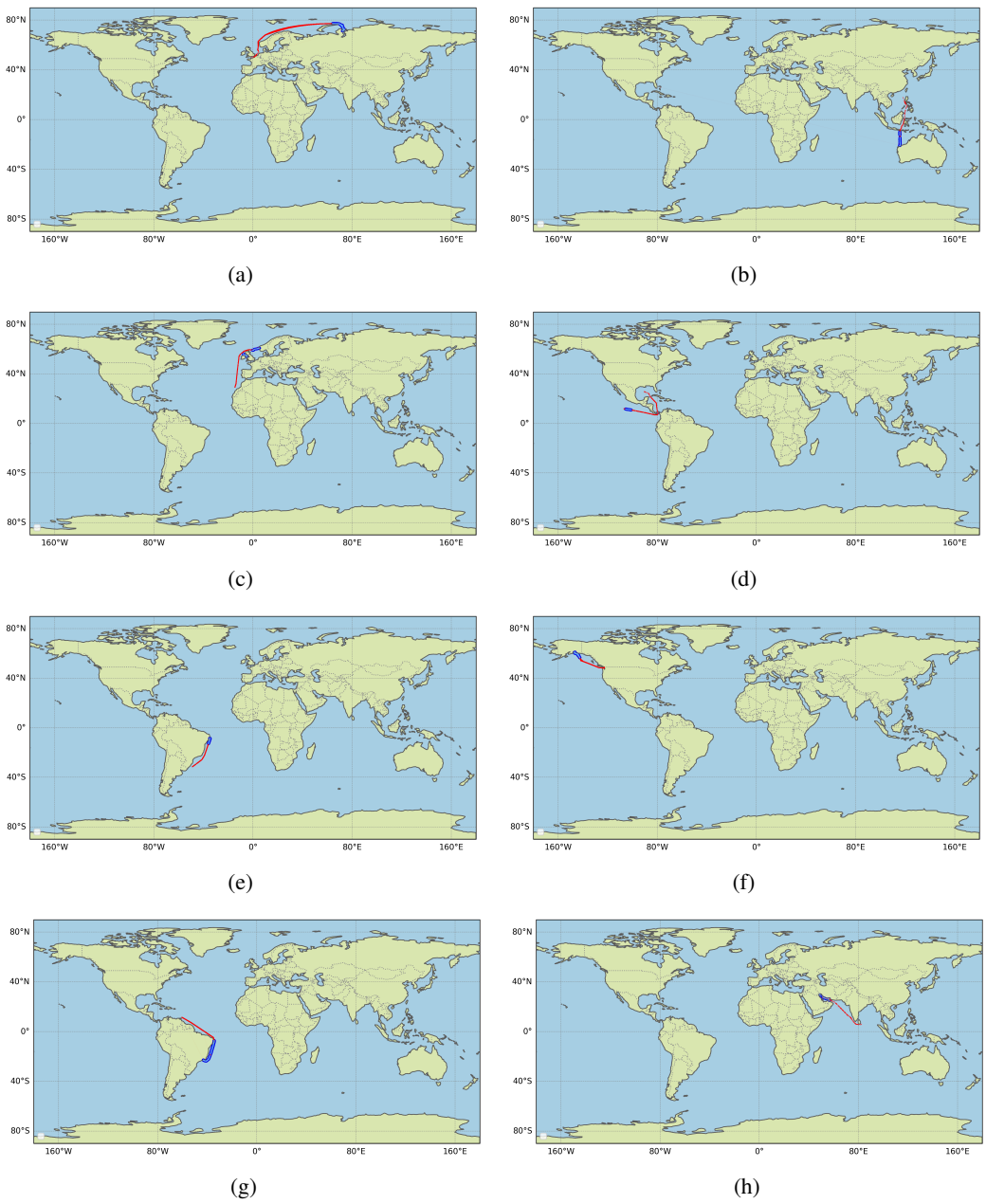
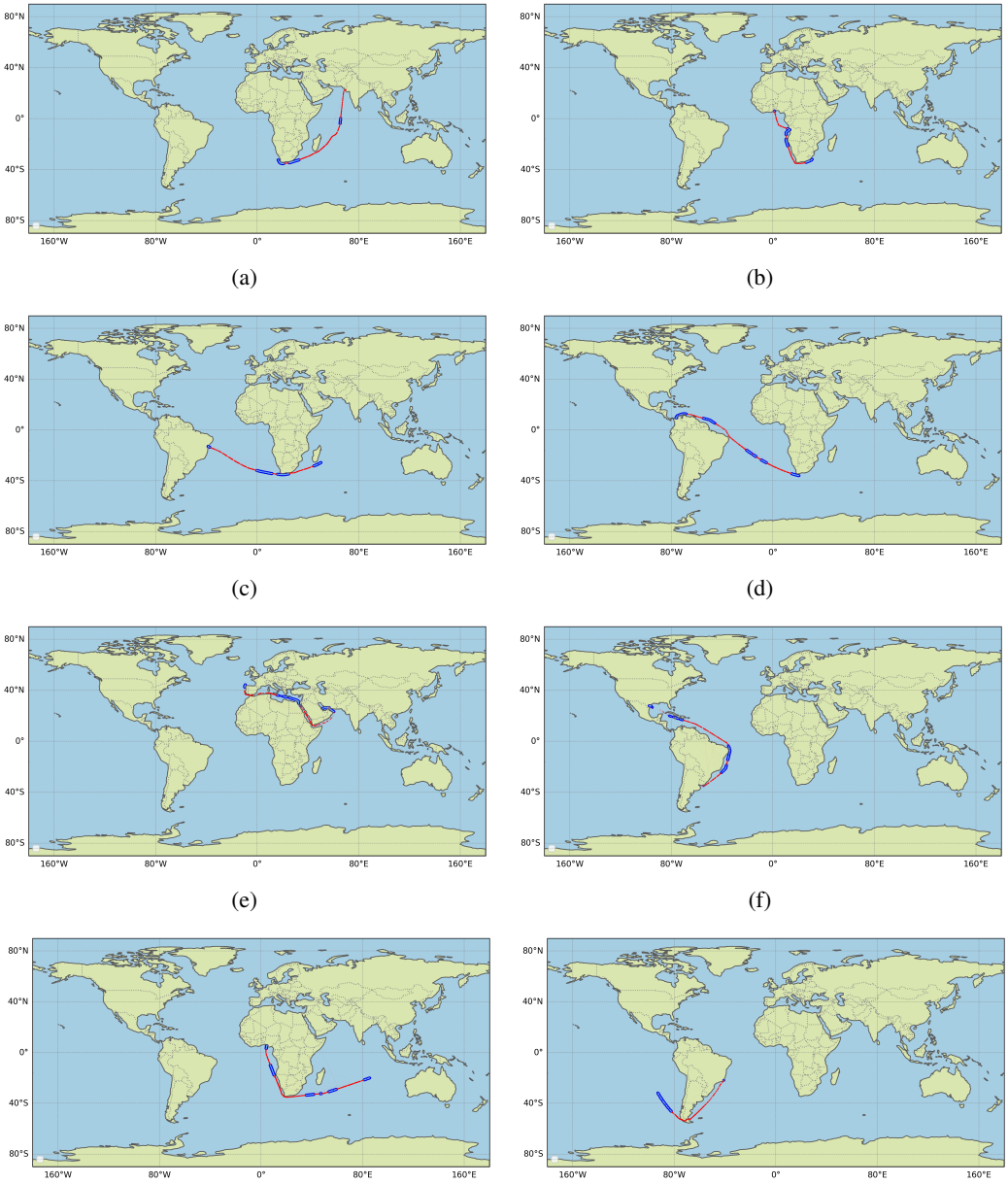


Figure 11: Prediction examples with 10 observed days and 5 target days. Red: observed points; blue: generated portions.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 12: Imputation examples, group B. Red: observed points; blue: generated portions.

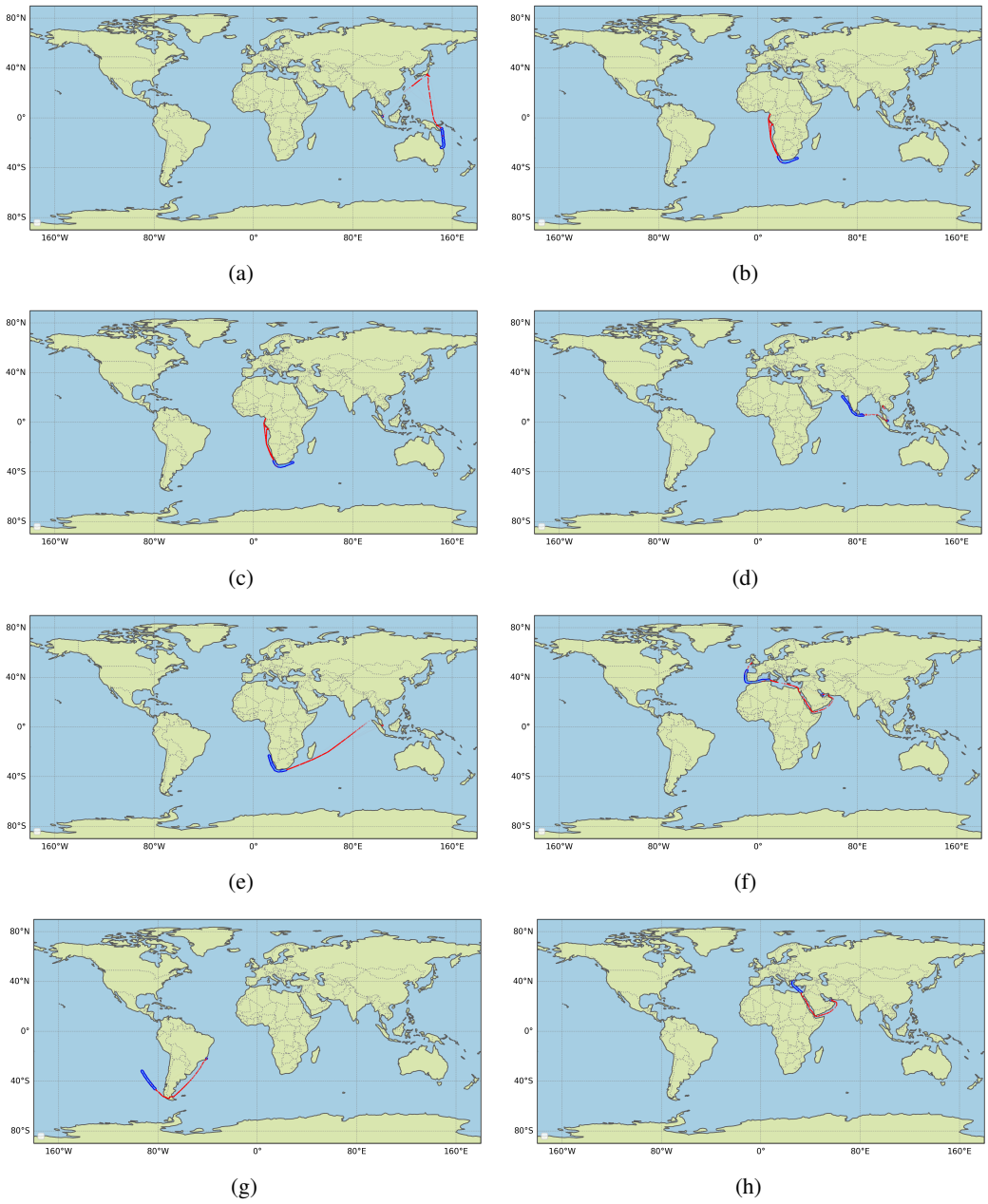


Figure 13: Prediction examples with 20 observed days and 5 target days. Red: observed points; blue: generated portions.

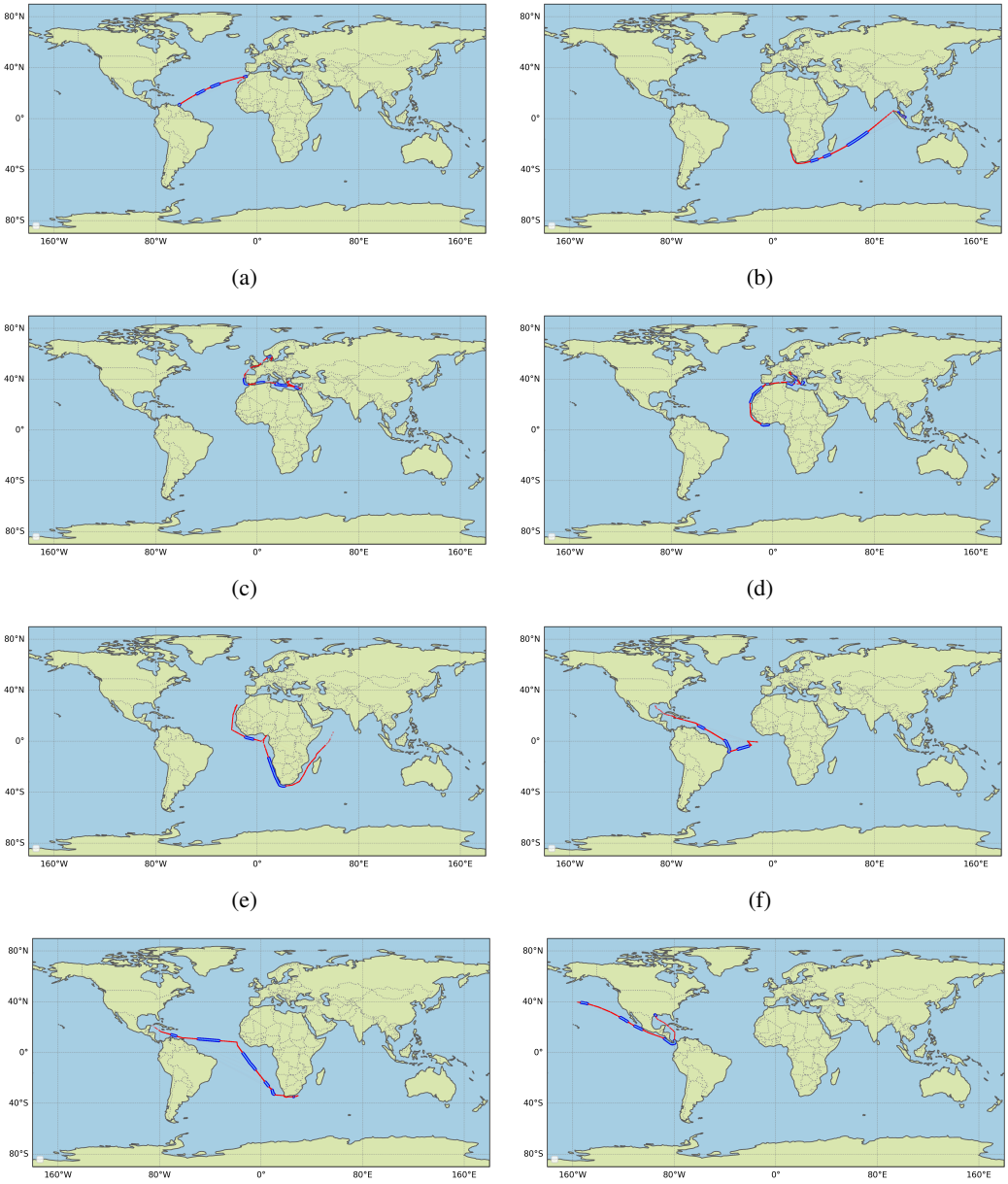


Figure 14: Imputation examples, group C. Red: observed points; blue: generated portions.

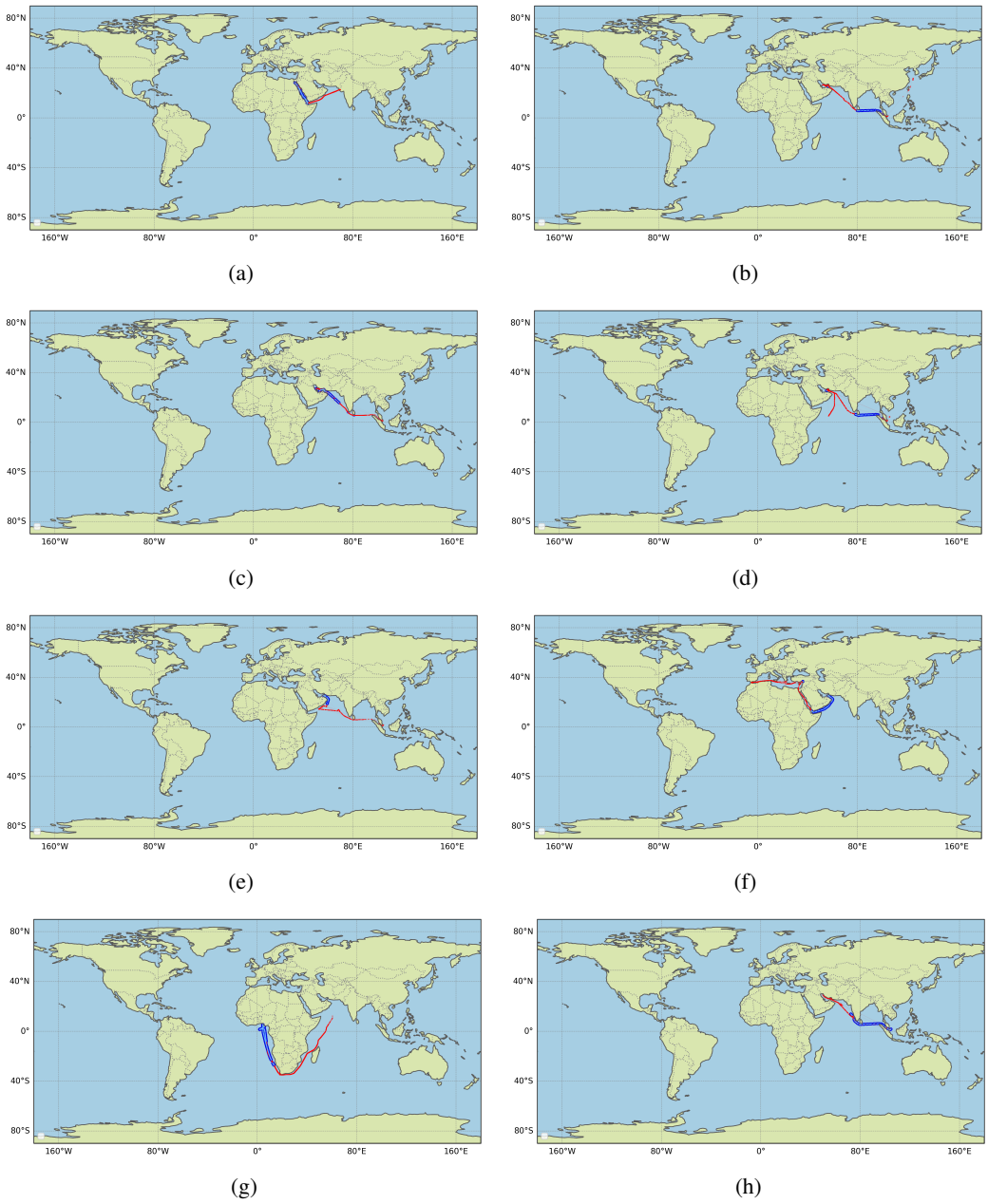


Figure 15: Prediction examples with 29 observed days and 5 target days. Red: observed points; blue: generated portions.

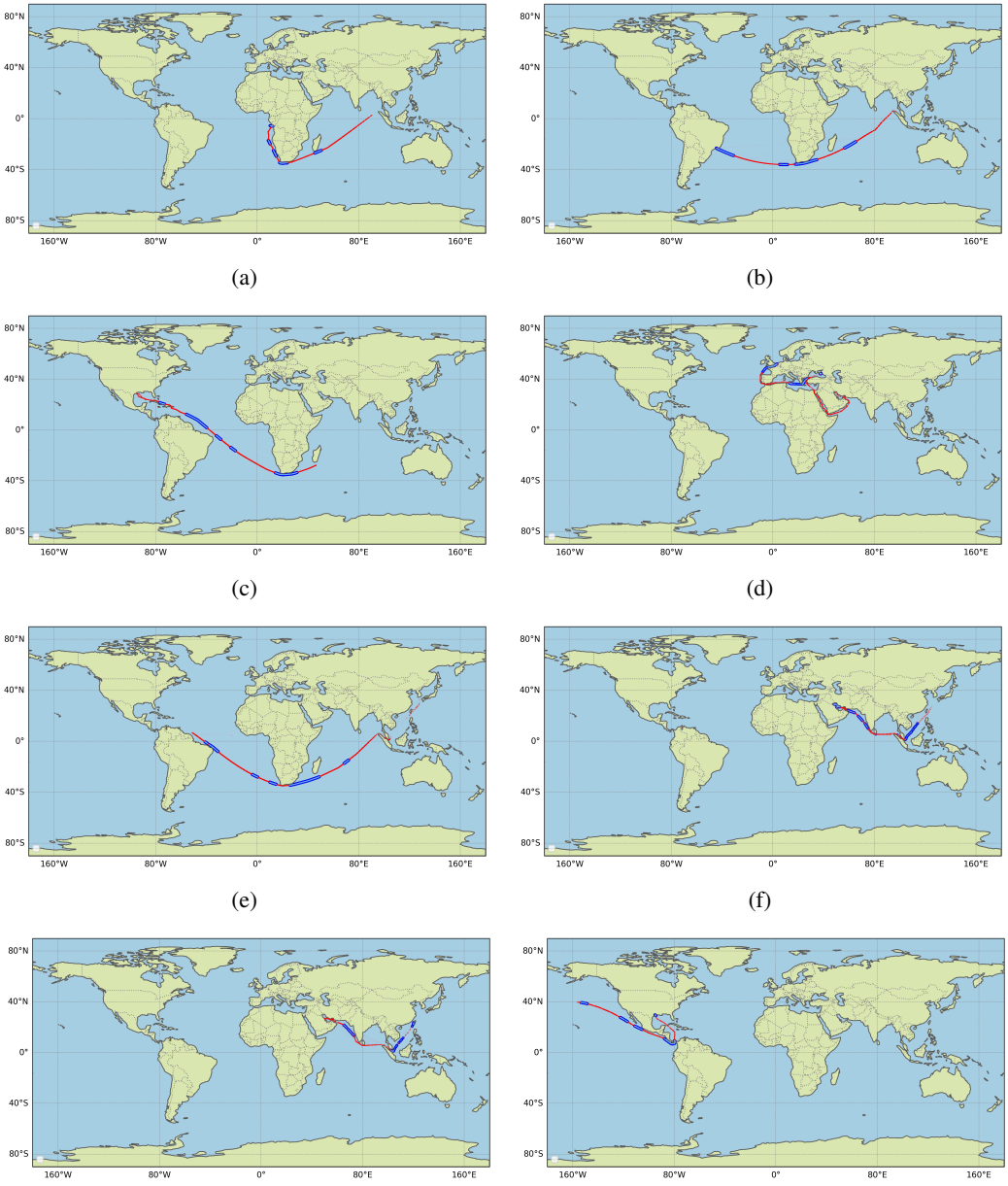


Figure 16: Imputation examples, group D. Red: observed points; blue: generated portions.

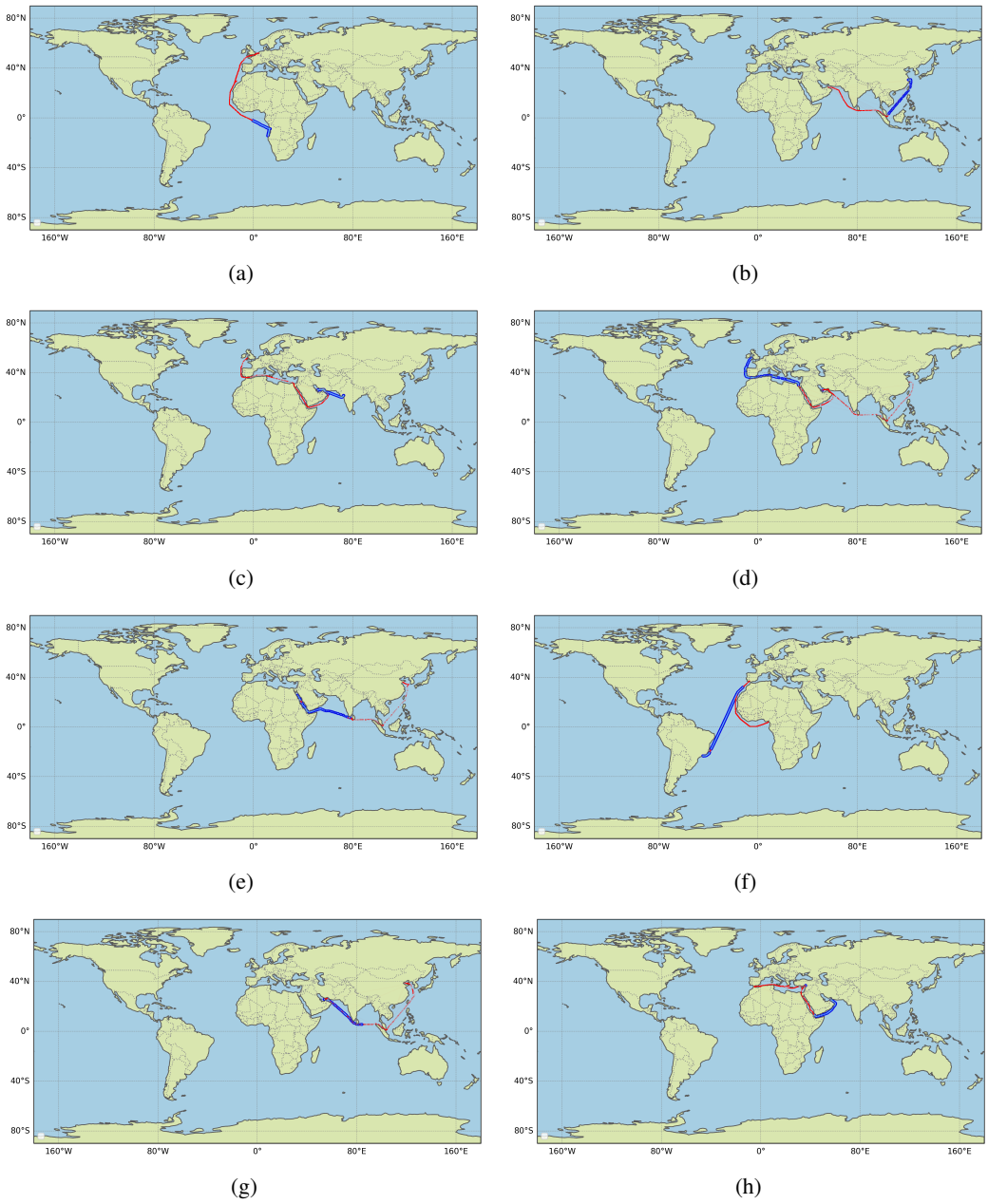


Figure 17: Prediction examples with 38 observed days and 5 target days. Red: observed points; blue: generated portions.

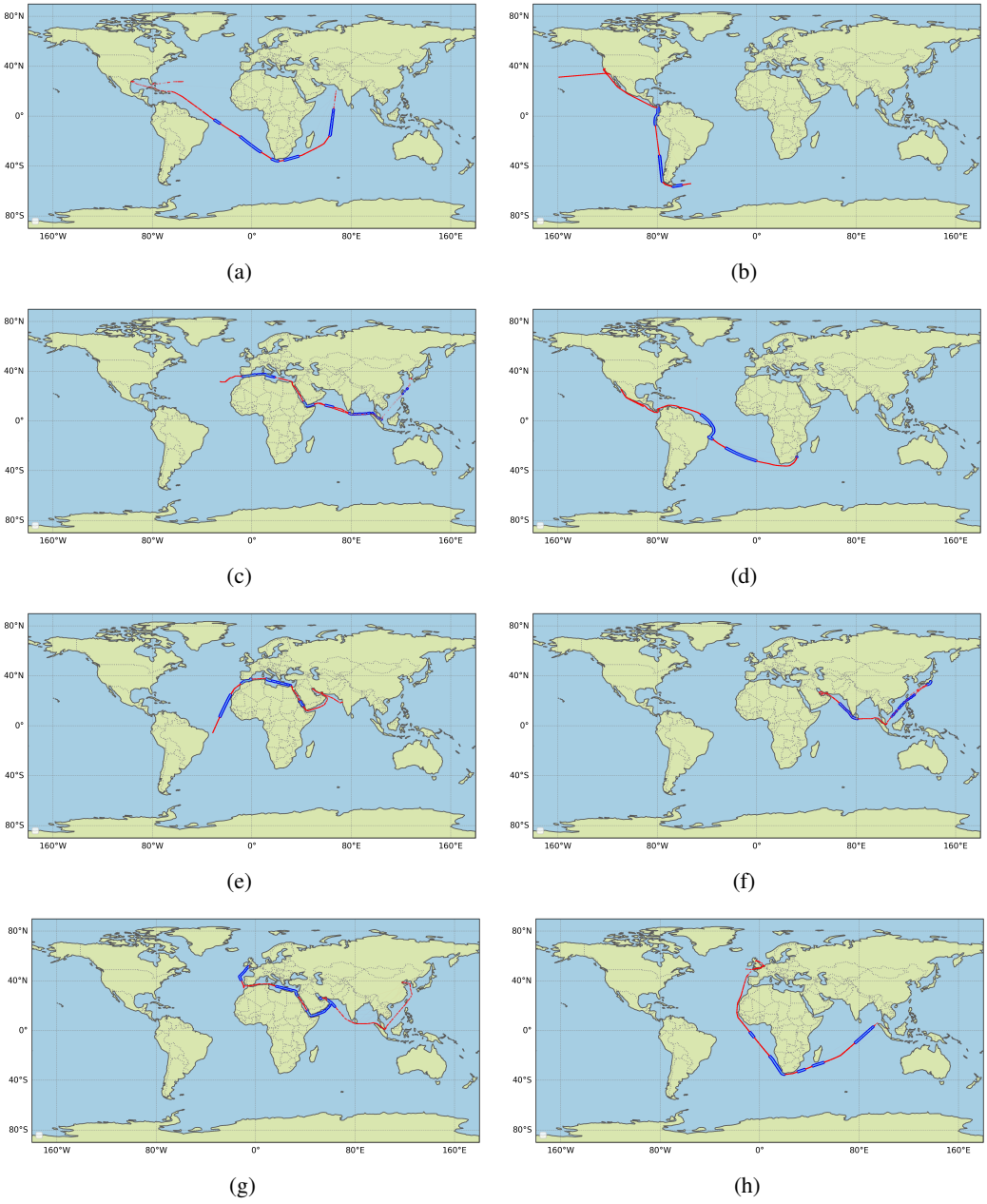


Figure 18: Imputation examples, group E. Red: observed points; blue: generated portions.

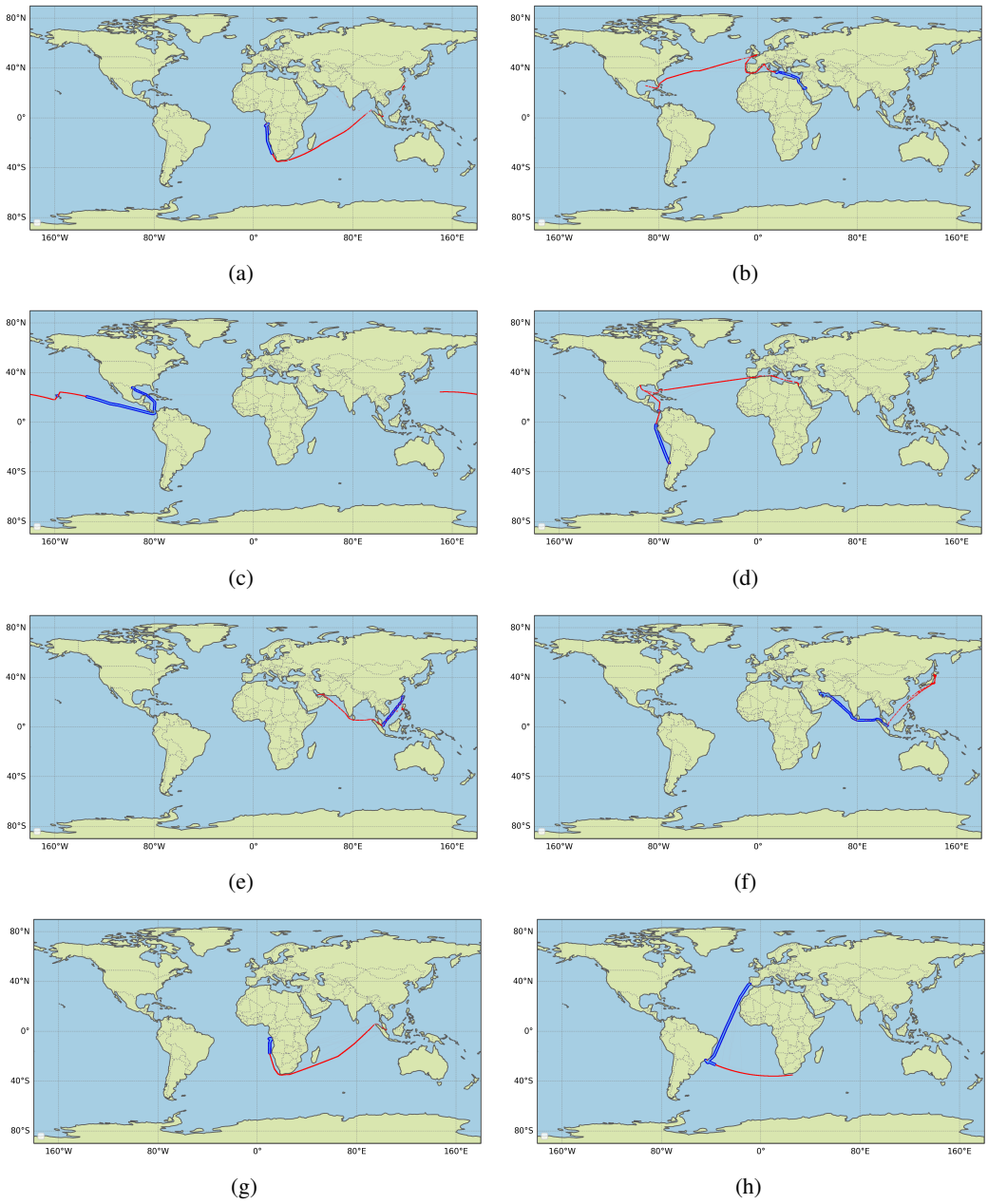


Figure 19: Prediction examples with 55 observed days and 5 target days. Red: observed points; blue: generated portions.