

SalienTime: User-driven Selection of Salient Time Steps for Large-Scale Geospatial Data Visualization

Juntong Chen

jtchen@stu.ecnu.edu.cn

School of Computer Science and Technology
East China Normal University
Shanghai, China

Zhong Peng

zpeng@sklec.ecnu.edu.cn

State Key Laboratory of Estuarine and Coastal Research
East China Normal University
Shanghai, China

Haiwen Huang

hwhuang@stu.ecnu.edu.cn
School of Computer Science and Technology
East China Normal University
Shanghai, China

Huayuan Ye

huayuan221@gmail.com
School of Computer Science and Technology
East China Normal University
Shanghai, China

Chenhui Li

chli@cs.ecnu.edu.cn
School of Computer Science and Technology
East China Normal University
Shanghai, China

Changbo Wang

cbwang@cs.ecnu.edu.cn
School of Computer Science and Technology
East China Normal University
Shanghai, China



Figure 1: SaliensTime User Interface. The dataset displayed here is the significant wave height of a 2-year global wave hindcast data. The contextual visualization we introduced (d, e) and the latent space exploration (b) facilitate efficient exploration and selection of salient time steps from large-scale geospatial data. More details are elaborated in Section 5.

ABSTRACT

The voluminous nature of geospatial temporal data from physical monitors and simulation models poses challenges to efficient data

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '24, May 11-16, 2024, Honolulu, Hawai'i

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
<https://doi.org/https://doi.org/10.1145/3613904.3642944>

access, often resulting in cumbersome temporal selection experiences in web-based data portals. Thus, selecting a subset of time steps for prioritized visualization and pre-loading is highly desirable. Addressing this issue, this paper establishes a multifaceted definition of salient time steps via extensive need-finding studies with domain experts to understand their workflows. Building on this, we propose a novel approach that leverages autoencoders and dynamic programming to facilitate user-driven temporal selections. Structural features, statistical variations, and distance penalties are incorporated to make more flexible selections. User-specified priorities, spatial regions, and aggregations are used to combine different perspectives. We design and implement a web-based interface to

enable efficient and context-aware selection of time steps and evaluate its efficacy and usability through case studies, quantitative evaluations, and expert interviews.

CCS CONCEPTS

- Human-centered computing → Visualization systems and tools; • Information systems → Geographic information systems;
- Computing methodologies → Learning latent representations.

KEYWORDS

Key Time Selection, Geospatial Data, Visualization Design, Large-scale Data Visualization, Need-finding Study

ACM Reference Format:

Junlong Chen, Haiwen Huang, Huayuan Ye, Zhong Peng, Chenhui Li, and Changbo Wang. 2024. SalientTime: User-driven Selection of Salient Time Steps for Large-Scale Geospatial Data Visualization. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, New York, NY, USA, 19 pages. <https://doi.org/10.1145/3613904.3642944>

1 INTRODUCTION

With the rapid advancements in physical monitoring technologies and numerical simulation models, geospatial data has grown exponentially in both ubiquity and volume [69]. Today, there are over 1000 Earth Observation satellites orbiting the globe, producing terabytes to petabytes of observation data daily. A single full-resolution remote-sensing image can take a few hundred megabytes of storage space [72]. Numerical hindcast models such as WAVEWATCH III [62] output gigabytes of global oceanography data hourly. This wealth of data serves as a powerful tool to analyze natural phenomena and geographical characteristics such as water quality assessment, natural disaster monitoring, climate change detection, land use mapping, etc. [8] As the spatial resolution increases and more satellites launch missions on the horizon, the data volume will only get larger in the future [32], bringing new challenges for storing, accessing, processing, and visualization.

The true potential of these geospatial big data is only realized when accessible to users, necessitating an effective approach for data access. As of 2022, the Sentinel project [11] by the European Space Agency (ESA) has published over 60 million datasets, gaining more than 500 PB of downloads since the launch of the operation [15]. Currently, the typical data infrastructure for serving these geospatial data is to store them in a cloud-based storage system [70], organized by spatial resolution and temporal granularity, allowing users to access via authenticated HTTP/FTP servers, or APIs. In the traditional workflow, researchers often faced lengthy periods to download the full-sized data to the local workstation before performing analytical tasks. A recent shift towards more user-friendly approaches has been observed. Many organizations now offer a data access hub: a map-centered web platform that facilitates interactive data exploration and visualization directly on maps. Notable examples of such platforms include NASA Worldview [42], NOAA View Global Data Explorer [44], USGS EarthExplorer [65], Copernicus Open Access Hub [16], etc. One common feature of these platforms is to create time-lapse playbacks [42, 52], allowing

users to observe the temporal variation of the data. In such workflows, users begin by choosing a dataset, then visualize a segment of the data to determine its appropriateness. After confirmation, they specify regional and temporal constraints to download the full dataset. With these data exploration platforms, users are empowered to navigate efficiently and assess the dataset's usability without extensive downloads.

However, due to limited network bandwidth and the voluminous nature of geospatial data, visualizing a single frame often demands several seconds. When users switch to a different time step, they confront another waiting period. Though platforms like NASA Worldview have attempted to address this issue with buffer queues, which pre-downloads subsequent data when users inspect data at a certain time step, temporal data exploration remains cumbersome. Moreover, users often navigate between time steps randomly without adequate contextual information, leaving them in the dark about potential patterns and correlations. Ziegler et al. [78] identified it as one of the major challenges in geospatial systems: users struggle to find desired temporal constraints for data. Therefore, if a representative subset of time steps from the original dataset can be selected for prioritized download and visualization, users can swiftly refine their time selections and identify desirable download targets efficiently.

This problem defines the scope of our research: selecting an informative subset of time steps from large-scale geospatial temporal data to assist users with temporal selection. Existing research primarily aims to minimize the reconstruction error of the *piece-wise linear interpolated* [68] data as the goal for selection. That is, the data between two selected time steps are linearly interpolated to generate full-duration data as the reconstructed data for comparison. To achieve this, dynamic programming-based methods [63, 76] and deep learning-based methods [34, 49] are two main approaches. These methods underscore the *summerizability* of the selected time steps, employing metrics like RMSE and SSIM to quantify the quality of the selection. However, our need-finding study shows that under different datasets and analytics tasks, users have different requirements for time step selection. Users are not solely interested in the summerizability of the selected time steps. They also value the amount of information brought by the subset of data in different aspects. Moreover, they anticipate supplemental contexts to guide their selection. Current research that primarily focuses on reconstruction error does not apply to these specific needs, and a user-driven approach that allows users to define priorities for time selection is highly desirable.

In this paper, we first establish a multifaceted definition for salient time steps in the context of geospatial big data via an extensive need-finding study with domain exports. We then introduce a novel approach that leverages autoencoders and Dynamic Programming (DP) algorithm to facilitate user-driven selections of salient time steps. Our network is designed to capture the intrinsic structure of the data, generating meaningful latent codes for measuring structural similarity. Statistical variation and distance penalty are incorporated in the DP process to make more flexible selections. Different perspectives of selection are combined with users-specified priorities, spatial regions and aggregation methods. Additionally, we design and implement an interactive system that

enables users to efficiently identify salient time steps from large-scale geospatial data. The efficacy of our approach is demonstrated via case studies on algal bloom and hurricane data and quantitative evaluations on 5 diverse geospatial datasets.

In summary, the contributions of this paper are:

- We establish a multifaceted definition for salient time steps in the context of geospatial big data via an extensive need-finding study with domain experts.
- We propose a novel approach that employs autoencoders and dynamic programming to select salient time steps from geospatial data with user-specified priorities. We incorporate our approach in an interactive system, facilitating efficient temporal navigation and data exploration.
- We conduct two case studies, quantitative evaluations on diverse real-world datasets and expert interviews to evaluate the efficacy of our approach and the usability of our system.

2 RELATED WORK

2.1 Time Selection from Spatiotemporal Data

Selecting a subset of representative time steps automatically from temporal data has been an active field of research in scientific visualization. Judicious selection of salient time steps can lower the network bandwidth requirement and facilitate rendering time allocation [5]. One approach is clustering, where Akiba et al. [1] group similar time steps into clusters and select one time step from each group. However, this fails to capture the chronological information and may lead to repeated selection clustered in a short period. Another prevalent approach is to employ specific metrics to quantify the variation between time steps and select frames that are most dissimilar from their predecessors. Common metrics include entropy-based measures, where Wang et al. [5] employ mutual information, Zhou et al. [76] and Dutta et al. [12] use *variation of information*, which is based on conditional entropy. Tong et al. [63] use Dynamic Time Wrapping (DTW), which measures the distance between two sequences to map the entire sequence onto a user-specified number of key time steps. In terms of algorithms, greedy-based methods [5, 67, 68] and dynamic programming-based methods [63, 76] are commonly employed. The greedy approach considers local variation and looks for maximizing the variation of the current time step, thus having no optimality guarantees [68]. Dynamic programming-based methods consider all combinations and are guaranteed to find the global optimal results. Zhou et al. [76] propose an approximation approach, in which a multi-pass algorithm with sliding windows is employed to reduce the computation needed for the cost function. Other techniques include a flow-based method proposed by Frey et al. [17], starting with random samples and employing minimum-cost flow to progressively adjust time steps that maximize the coverage with the original data. Pulido et al. [50] applies non-negative Tucker tensor decomposition to extract latent time features and map these features to their maximal values.

Recently, another branch of deep learning-based method has been proposed. Porter et al. [49] uses an Autoencoder network to encode each time step to latent space and use a combination of arc-length-based and angle-based selection to choose frames in the dimension-reduced latent space. In the same year, Liu et al. [34]

proposed to employ Deep Metric Learning (DML) that utilizes a Siamese deep neural network to select key time steps in a supervised way.

To evaluate the quality of selection results, one common approach is to reconstruct data from the selected time steps via linear interpolation and quantify total reconstruction errors. This is typically carried out using image quality metrics such as Root Mean Square Error (RMSE) [49, 50, 76], Peak Signal-to-Noise Ratio (PSNR) [49, 50], Structural Similarity Index Measures (SSIM) [50], and Total Absolute Error (TAE) [50]. Another approach is to directly evaluate the cost metric used for frame selection, such as the Dynamic Time Wrapping distance by Tong et al. [63], and the Flow-based distance by Frey et al. [17]. In our study, we use linear interpolation as an intuitive method to reconstruct the full data, and aligning previous studies, employ RMSE and SSIM to measure reconstruction quality.

Leveraging the feature representation ability of neural networks, deep learning-based methods can effectively capture the structural variations in the data. Meanwhile, DP-based methods offer high explainability, flexible cost definitions and global optimality. Our approach combines the two, selecting informative salient time steps that address user needs.

2.2 User-driven Approaches for Geospatial Visualization Systems

Visualization systems for geospatial data are an active field of research due to their significance in helping researchers understand and analyze the pattern, trend, uncertainty and correlation of data [7]. Numerous systems are proposed for various applications, such as climatology[9], aerodynamics[20], social media analysis[30], and urban computing [6, 54], as comprehensively reviewed in surveys by Chen et al. [7] and Deng et al. [10]. However, bias in the understanding of users' needs for spatial and temporal selection in geospatial systems is widespread and challenging to address [64, 78]. The concept *user-driven approaches* in geospatial systems encompasses quite a broad scope. It refers to the idea that we allow users to define their specified constraints or priorities to perform selections, rather than making presumptions about their preferences. For instance, MixMap [38] focuses on place-based similarity, allowing users to assign weights to different regional characteristics like age, education, and race, to identify similar regions. Zhou et al. [77] introduce a user-driven sampling model where users select representative points. A CNN network then learns these preferences to guide sampling across other spatial regions. Liu et al. propose TPFlow [33] that allow users to select informative spatiotemporal slices from large-scale spatiotemporal data. Based on tensor decomposition, users can progressively drill down different data dimensions to gain insight. The term also implies that access to spatial data should align with users' interactions and activity history [58]. For instance, Guo et al. [19] samples a subset of spatial objects based on different zoom levels and panning locations, facilitating efficient selection from large-scale geospatial data. Pan et al. [45] leverages collective user behaviors to predict which tiles should be prefetched or replaced in the cache to improve geospatial system performance. In terms of selecting salient time steps, *user-driven* also refers to providing contextual information with visualizations to assist interactive temporal selections, following

the focus + context visualization scheme [55]. For example, TransGraph [74] provides hierarchical visualizations of the time steps, allowing users to select time steps at different granularity levels. Dissimilarity maps [63] are 2D heatmaps that illustrate the differences between the current selection and the original data. Time Activity Curves (TACs) [60, 61, 67] provide visualizations about the magnitude of variation, offering insights into temporal data patterns.

We adopt the aforementioned user-driven ideologies by offering users adjustable priorities for selection, ensuring our approach applies to different tasks. We also incorporated contextual visualizations, including temporal and relative trends, to assist users in refining their current selection. Additionally, our system takes into account the user's previous manual labeling when computing selections.

2.3 Spatial Feature Representation

Feature representation plays an important role in various domains, such as dimensionality reduction [27, 36, 37], anomaly detection [14, 24], data steganography [73, 75], etc. Early approaches are commonly based on traditional machine learning methods. Principal Component Analysis (PCA) [27] projects the original data onto a low-dimensional space, retaining the main features of data variation. Linear Discriminant Analysis (LDA) [25] uses a supervised learning method to find a linear combination of features that characterizes different classes of objects. Papadimitriou et al. [46] proposed to leverage Singular Value Decomposition (SVD) to implement feature extraction, which is later widely used in recommender system [31, 48]. There are also some feature representation methods based on domain transform, e.g., Wavelet Transform [57] and Fourier Transform [56].

Recently, a variety of deep learning-based feature representation schemes have been proposed and achieve impressive performance. These methods are generally based on autoencoders. Autoencoder is a kind of neural network that is designed to efficiently learn a low-dimensional representation (which is usually called latent code) for a set of unlabelled data. The learned latent code can be used for numerous kinds of downstream tasks, such as image reconstruction and representation [21, 26, 66], sentiment prediction [59], etc. There are some studies [18, 22, 51] that combine autoencoder and adversarial training to implement image generation, which is generating new images using the learned latent code representation. Kingma et al. proposed Variational AutoEncoder (VAE) [29] for generation tasks by adding conditional constraints on latent code.

In this paper, we propose a new method utilizing autoencoders to efficiently learn feature representation of spatiotemporal data and quantify the structural variation of data frames based on the learned latent code.

3 NEED-FINDING STUDY

To gain insights into what characteristics make time steps *salient* to users and require prioritized visualization, we conducted a need-finding study with domain experts through observational studies and semi-structured interviews. In this section, we first describe

the participants and structure of our interview, then elaborate on our findings.

3.1 Methodology

3.1.1 Participants. We engaged with 5 domain experts from both academia and industry (age $\mu = 32.6$, $\sigma = 6.8$) via email contact, direct messages and verbal communication. Expert 1 (E1) has held the position of Principal Oceanographer in a research institute for more than five years, and has transitioned into academia, being a professor specializing in Coastal Dynamics at University E. Expert 2 (E2) is a senior engineer in an ecological institute, currently in charge of a project about algal bloom monitoring and early warning. Expert 3 (E3) is a software engineer working in the data visualization department of a leading company. Expert 4 (E4) is a professor at University N, studying urban data visualization, pervasive computing, and human-computer interaction. Expert 5 (E5) is an associate professor at University E specializing in data visualization and spatiotemporal data mining. The participants' domains and backgrounds are summarized in Table 1.

Table 1: Domain and experience of the experts in our need-finding study.

ID	Domain	Background	Years of Exp.
E1	Earth Science	Academia	>10
E2	Earth Science	Industry	5-10
E3	Data Visualization	Industry	5-10
E4	Data Visualization	Academia	5-10
E5	Data Visualization	Academia	>10

3.1.2 Platforms for Observation. To observe the participant's usage pattern for temporal selection within geospatial systems, we use three widely-used platforms: NASA Worldview [42], Sentinel Hub EO Browser [52] and MetOceanView [39]. These platforms are selected as they are frequently used by geoscience and visualization researchers to retrieve data, as suggested by E1, E2 and E5. They also exhibit similar functionality. NASA Worldview enables real-time interaction with over 1,000 global, full-resolution satellite imagery layers, updated hourly to support time-critical applications. Sentinel Hub EO Browser provides a complete archive of Sentinel, Landsat, and other ESA satellite products, with customizable visualizations and highlight recommendations for earth events. MetoceanView specializes in oceanography hindcast and forecast data, allowing animated visualizations across hundreds of datasets and models. Figure 2 shows screenshots of these platforms.

These platforms exhibit three core commonalities: 1). They serve as the web-based access portal to an extensive amount of data; 2). They provide categorical, spatial and temporal constraints to facilitate dataset discovery; 3). They offer map-based visualizations and time-lapse video creation capabilities. Users will undergo multiple rounds of filter, exploring and visualization iterations to find the download target, allowing us to understand the decision-making process involved.

3.1.3 Session Structure. Each session was conducted in a face-to-face setting, lasting about 60 minutes. The session starts with a

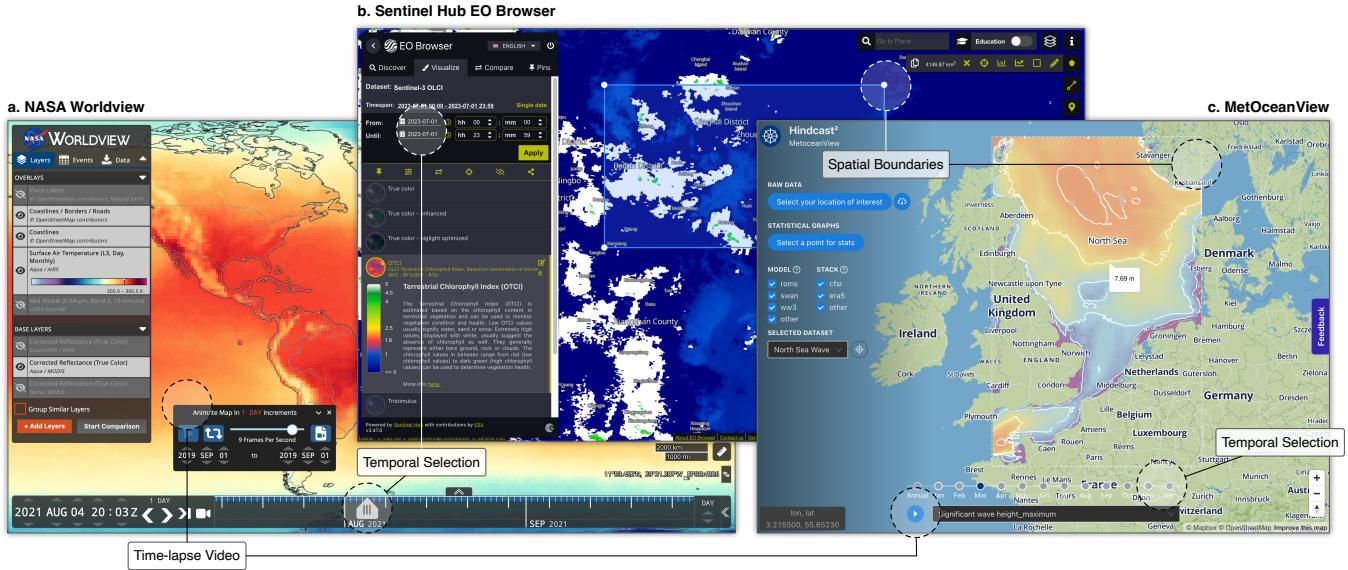


Figure 2: The platforms [39, 42, 52] we used for the observational study.

10-minute introduction that outlines the research context and objectives. This is followed by a 25-minute free exploration phase and a 25-minute semi-structured interview. During the introduction phase, we asked for the participants' consent about screen recording for further analysis. In the exploration phase, we observed participants' behavior and encouraged them to describe their ongoing intentions and task status. They were first instructed to explore all three platforms' functionality in terms of dataset selection, spatial and temporal navigation, and time-lapse video creation and then perform at least one full data utilizing insights provided by previous actions. We do provide explicit directives on which dataset to download due to the participants' different backgrounds and domain knowledge. Allowing participants to choose their tasks also helps us to get a more authentic understanding of real-world conditions and more diverse insights.

The interview phase is structured around the following outline:

- (1) **Motivation & Outcome:** What dataset and the time range were selected, and what motivated the selection of the dataset and time range?
- (2) **Decision-making Factors:** What contextual information from the platforms influenced their decisions in terms of spatial and temporal constraints when exploring datasets?
- (3) **Temporal Selection:** How do they navigate between time steps for visualization, and how is the selection of the time range for download made?
- (4) **Interaction Remarks:** What were the participants' assessments of the interactions involved within the data exploration and visualization workflow?

We also incorporate our observations and may revisit screen recordings based on user feedback. We concurrently take notes to capture key insights.

3.2 Findings

We organize our findings in the Figure 3. We synthesize common opinions from notes and list other insights in the “additional remarks” section. According to our study, participants initiate their tasks by identifying target datasets. In this phase, spatial boundary visualizations (E1, E3, E5), preview images and metadata catalogs provide valuable assistance. Upon dataset selection, we find that participants struggle to navigate between time steps due to the absence of contextual guidance (E1-E5). They are “*not certain about which time step to jump after inspecting current one*”(E3) and “*frequently go back and forth between time steps for comparison*”(E5). As a result, most participants (E2-E5) primarily use random navigation between time steps. Only those with explicit intentions, such as investigating specific earth event occurrences (E1, E2), resort to targeted navigation (E1, E2, E5), where they explicitly input dates and times in the temporal selector.

After the initial selection of time ranges, they use sequential navigation (E1, E4) and generate time-lapse videos (E1, E2, E4, E5) to check their selections before downloading. These videos are useful for inspecting the dynamic variation and ensuring the datasets meet their needs (E2). In this phase, some participants (E2, E4) report slow response times of more than 10 seconds due to the large volume of data needed. This long period of waiting time also makes the navigation between time steps difficult (E2, E5). Additionally, some participants (E1, E3) note the absence of more flexible temporal granularity options, with E1 stating that “*solely providing monthly previews are insufficient since the rate of change differs across datasets*”.

3.2.1 Salient Time Step Definition. Our study reveals that the users' priority for time selection varies in different datasets and tasks; a singular definition fails to cater to users' flexible requirements. For instance, for the task of investigating the algal bloom situations (E2), the users expect to locate the time steps that are most severe

Motivation & Outcome	E1	E2	E3	E4	E5
To analyze storm tides by examining significant wave height data for the North Sea area.	<i>MetoceanView</i> North Sea Wave, Jan 2017	To investigate algal bloom conditions in the Yangtze River Estuary using Sentinel-3A OLCI data.	To produce high-quality tiled maps for web applications using high-resolution satellite imagery.	To explore the distribution of global precipitation rates on an annual basis.	To inspect urban air pollution levels and generate heatmap datasets for subsequent research.
<i>MetoceanView</i> North Sea Wave, Jan 2017	<i>EO Browser</i> Sentinel - 3 OLCI, 2016	<i>EO Browser</i> Landsat 8-9 L2, June 2023	<i>NASA Worldview</i> GCOM W1/AMSR2, 2021	<i>NASA Worldview</i> MODIS/Aerosol Optical Depth May 2020	
Decision-making Factors	Spatial Boundary Visualizations ✓		✓	✓	
	Earth Event Occurrence ✓	✓			
	Time-lapse Video ✓	✓		✓	
	Dataset Visualization ✓	✓	✓	✓	✓
Temporal Selection	Sequential Navigation ✓			✓	
	Random Navigation	✓	✓	✓	✓
	Stepped Navigation ✓	✓		✓	
	Targeted Navigation ✓	✓			✓
Interaction Remarks	Slow Response Time ✓		✓	✓	
	Lack Contextual Guidance ✓	✓	✓	✓	✓
	Lack Temporal Granularity ✓		✓		✓
	Hard to Navigate Between Time ✓			✓	
Additional Remarks	<ul style="list-style-type: none"> Since the change rate differs across datasets, solely providing monthly previews are insufficient. The maximum value within a region is particularly important for extreme weather conditions. 	<ul style="list-style-type: none"> Though no temporal comparison provided, the spatial comparison helped the selection process. For multiple data sources, they need to be combined to perform analysis. 	<ul style="list-style-type: none"> Visualizing a tile of data for one time step is taking too much time. Not certain about which time is better for the task, leading to multiple random navigations. 	<ul style="list-style-type: none"> It will be good if additional information about data variation is provided. Prefer sequential preview at smaller zoom levels and stepped preview on larger zoom levels. 	<ul style="list-style-type: none"> Aggregated data summaries will be helpful to determine anomaly and patterns. Frequently go back and forth between time steps due to the absence of contextual information.

Figure 3: The compiled results of our need-finding study. The participants' domains are illustrated in backgrounds with green (Earth Science) or blue (Data Visualization) color.

or degrading with the fastest rate; for the task of exploring the distribution of precipitation rates (E4), the users expect to get a quick overview of the data with minimum frames of download. Therefore, we define *salient time steps* from the following three perspectives:

DF1. Summarizability. Salient time steps are those that can effectively summarize the critical information and trends contained in the dataset. This includes significant alternation in the data's spatial features over time, or rapid changes in data values that signal crucial dynamics.

DF2. Anomaly. Salient time steps are those that represent anomalies or outliers in the dataset. These steps reveal the deviation from the norm, such as rare occurrences of unusual precipitation or temperature - structurally different frames from the rest.

DF3. Extremum. Salient time steps are those extreme values within a certain spatial or temporal region of the data. These could be maximum or minimum, highlighting key moments of peak activity or low points that require additional attention.

By proposing these three dimensions, we aim to provide complementary, rather than mutually exclusive multifacet definitions for salient time steps. We intend to integrate these three dimensions in our proposed algorithm in Section 4, as well as the interactive systems detailed in Section 5.

3.2.2 Design Requirements. According to the participants' remarks on interaction, we identified several key design requirements for our system to support users' diverse task and data exploration needs.

DR1. Temporal Context. Users need to be aware of the structure of the dataset to make temporal navigation decisions. The system should provide temporal contexts to facilitate navigation. Here *Temporal Context* refers to the auxiliary information informing users about the current state of selections and temporal patterns of data. This could be the visual clues about the availability and loading state of data, the visualizations for the variation trends, or the similarities between data frames.

DR2. Global & Regional. The characteristics between the global region, i.e., the entire data-available region of datasets, and regional features, i.e., the users' area of interest differ. Local variations can often be obscured by overarching global changes, hindering users from identifying time steps. Therefore, our system should support the selection of different spatial analysis regions, providing temporal context at varying spatial scales.

DR3. Dynamic & Static. Time-lapse videos are useful to illustrate the general variation of data, outlining the event dynamics, whereas the specific values of pixels in a static raster provide accurate information. The system should support both dynamic and static visualizations to facilitate user exploration.

DR4. Performance. Long periods of waiting times are detrimental to users' exploration experience, making navigating between time steps cumbersome. The system should provide smooth navigation between time steps. Reasonable waiting time to generate visualizations for 1 frame should be no more than 2 seconds [13]. This can be achieved through appropriate caching and pre-fetching mechanisms.

These design requirements will guide the design and implementation of the system.

4 METHOD

Given a series of temporal data, our goal is to select a subset of time steps that is the most *salient* to users. Intuitively, we aim to evaluate the quality of our current selection iteratively to find the optimal solution. At each iteration, we need to determine if selecting frames i and j yields a better outcome than selecting frames i and k . The DP approach is suitable for this setting as it decomposes the problem into multiple non-overlapping sub-problems, solving each one only once, therefore guaranteeing global optimality.

Formally, let $\mathbf{X} \in \mathbb{R}^{t \times n \times m}$ be a 3D tensor representing the temporal geospatial data, where t is the number of time steps, n and m are the number of rows and columns of the spatial domain, respectively. Let \mathbf{X}_t represent the data at time step t . Let the selected salient time steps be denoted as $S = \{s_1, s_2, \dots, s_k\}$, where $s_i \in [1, t]$ and $s_i < s_{i+1}$. k is the number of salient time steps, specified by users and $k \ll t$.

4.1 Overview

We formulate the problem as a DP problem. We initialize a two-dimensional DP table, denoted as D , to store intermediate results. Each entry $D(i, j)$ represents the minimum cost associated with frame sequence S with the number of selected frames $|S| = i$ and the last selected frame $s_i = j$. Note that the first frame is always selected. The recurrence relation of the DP procedure is defined as follows:

$$D(i, j) = \min_{k \in [i-1, j]} \{D(i-1, k) + C(k, j)\}, \quad (1)$$

where $C(i, j)$ denotes the cost function, defined as the linear combination of the structural cost C_{struc} , the statistical variation cost C_{stat} , and the distance cost C_{dis} :

$$C(i, j) = \alpha C_{\text{struc}}(i, j) + \beta C_{\text{stat}} + C_{\text{dis}}. \quad (2)$$

For instance, $D(2, 3)$ denotes the minimum total cost for the time sequence $S = \{s_1, s_2\}$ with $s_1 = 1$ and $s_2 = 3$. In this specific case, $D(2, 3) = C(1, 3)$. Additionally, we use a two-dimensional array p to memorize the previous time step s_{i-1} that leads to the minimum cost $D(i, j)$, which is used to backtrack the optimal solution. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Salient Time Steps Selection

```

Require: X: Temporal geospatial data with  $t$  steps
          k: Number of time steps to select
           $\alpha, \beta$ : Parameters for cost function
Ensure: S: Salient time steps with  $|S| = k$ 
1: Initialize  $D \leftarrow \inf[1 \dots t][1 \dots k]$ ,  $p \leftarrow \text{zeros}[1 \dots t][1 \dots k]$ 
2: Define cost  $C(i, j) = \alpha C_{\text{struc}} + \beta C_{\text{stat}} + C_{\text{dis}}$ 
3: for  $i = 1, \dots, t$  do
4:    $D[i, 1] \leftarrow 0$ 
5:    $D[i, 2] \leftarrow C(i, 2)$            ▷ The first frame is always selected
6:   for  $i = 0, 1, \dots, t$  do
7:     for  $j = 1, 2, \dots, \min(i+1, k)$  do      ▷ Num. of selected steps
8:       for  $t = j-1, j, \dots, i-1$  do
9:         cost  $\leftarrow D[t, j-1] + C(t, i)$ 
10:        if cost  $< D[i, j]$  then
11:           $D[i, j] \leftarrow \text{cost}$ 
12:           $p[i, j] \leftarrow t$ 
13: Initialize  $S \leftarrow \text{zeros}[1 \dots k]$ , cur  $\leftarrow t-1$ 
14: for  $p = k, k-1, \dots, 1$  do           ▷ Backtrack to find the solution
15:    $S[p-1] \leftarrow \text{cur}$ 
16:   cur  $\leftarrow p[\text{cur}, p]$ 
17: return S

```

In our approach, the cost function is critical to the selection result as it quantifies the quality of current selections. When k frames are selected, the data between salient frames is lost, thus compromising the spatial continuity. To minimize information loss, we want each selected time step to be of significant importance and carry as much information as possible. This requires our cost function to be smaller when the co-existence of two frames in the selection result provides valuable insights, and vice versa. In other words, when two frames are different, the cost decreases. Unlike existing methods that measure the difference using information theory [76] or dynamic time wrapping [63], we propose a more expressive and flexible method to define the cost. In the following subsections, we elaborate on the three components of the cost function.

4.2 Structural Features

The structural cost C_{struc} quantifies the structural difference between two frames. This cost aligns with our definition of time steps DF1 (Summarizability) and DF2 (Anomaly), as it encourages selected frames to reveal the variation of inherent patterns, shapes and spatial characteristics within the data. Traditional methods often fall short in capturing these variations, especially when data exhibits different semantic spatial relationships but follows similar value distribution. We employ a CNN-based autoencoder to extract these features from data, where the encoder $Enc(\cdot)$ encodes an input frame $\mathbf{X}_t \in \mathbb{R}^{H \times W}$ into a low dimensional latent code $Z_t \in \mathbb{R}^\phi$. Here ϕ is a hyperparameter, denoting the dimension number. The

decoder $Dec(\cdot)$ reconstructs the frame $\hat{\mathbf{X}}_t$ from the latent code Z_t . The whole procedure can be formulated as: $\hat{\mathbf{X}}_t = Dec(Enc(\mathbf{X}_t))$. Our goal is to optimize the network parameters to minimize the difference between \mathbf{X}_t and $\hat{\mathbf{X}}_t$. Additionally, we employ a GAN-based discriminator for adversarial training, thereby enhancing the feature extraction ability of our network. Figure 4 shows the overall architecture of our network.

4.2.1 Training Dataset. We collected geospatial data primarily from the NASA Earthdata [41] and NOAA Environmental Modeling Center [43]. The data includes air temperature at different heights, global wave-wind field data, pressure of mean sea level, satellite imagery, etc. Each dataset covers approximately a range of 10 years. For the NaN (Not a Number) values such as land areas for oceanography data, we fill with zeros. Finally, we chose about 2,000 time steps from each dataset, generating a diverse collection of more than 11,000 images as our training dataset.

4.2.2 Encoder. The encoder takes a single-channel grey-scale image as input, consisting of 8 convolution layers for multi-scale feature extraction. The feature map size is halved after each convolution operation. Each convolution layer is followed by a batch normalization layer [23] and a LeakyReLU activation layer [35]. The output of each convolution layer is fed into another convolution layer to produce 8 feature vectors. These vectors are then concatenated as the final latent code: $Z_t = \{z_i\}_1^8$. Since Z_t is derived from different levels of feature map throughout the network calculation, it contains both high-level and low-level feature information of the original data.

4.2.3 Decoder. The decoder generates the reconstructed frame $\hat{\mathbf{X}}$ from the latent code Z_t . Each sub-feature vector z_i of Z_t is first fed into 8 deconvolution layers separately to produce 8 feature maps. Then, these feature maps go through a series of deconvolution layers for upsampling. Formally, the first layer takes z_1 as input. Afterward, the output of the i -th deconvolution layer is concatenated with the feature map of z_i and fed into the $(i+1)$ -th deconvolution layer. Each layer is followed by a batch normalization and LeakyReLU except the last layer which uses a Sigmoid activation to ensure the output values are in the range of $[0, 1]$.

The output of the decoder's last layer is the reconstructed frame $\hat{\mathbf{X}}$. We calculate the reconstruction loss with the mean squared error (MSE) to guide the network training:

$$\mathcal{L}_{rec} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \|\mathbf{X}_t(i, j) - \hat{\mathbf{X}}_t(i, j)\|_2, \quad (3)$$

where (H, W) denotes the image size.

4.2.4 Discriminator. The discriminator network $Dis(\cdot)$ serves as an additional supervision to the reconstruction quality. We adapt the design of GANs [18] to perform adversarial training. The discriminator takes \mathbf{X}_t and $\hat{\mathbf{X}}_t$ as input and produces two feature maps indicating the network prediction label. We train $Dis(\cdot)$ to discriminate the reconstructed image thus encouraging the autoencoder to generate a reconstructed image that cannot be distinguished from the original one. Formally, we optimize the network by minimizing:

$$\mathcal{L}_{dis} = \|Dis(\mathbf{X}_t) - \mathbf{I}_r\|_2 + \|Dis(\hat{\mathbf{X}}_t) - \mathbf{I}_f\|_2, \quad (4)$$

where \mathbf{I}_r and \mathbf{I}_f are the ground truth label of real or fake image.

4.2.5 Structural Cost. We measure the structural similarities of frames i and j by calculating the cosine similarities of their latent codes:

$$S(Z_i, Z_j) = \frac{Z_i \cdot Z_j}{\|Z_i\| \cdot \|Z_j\|}. \quad (5)$$

Then, the structural cost is then defined as:

$$C_{struc}(i, j) = \frac{1}{1 + e^{-5[S(Z_i, Z_j) - 0.5]}}. \quad (6)$$

The sigmoid-like function maps the cosine similarities to the range of $[0, 1]$, which is shifted and scaled to enlarge the difference between similar frames, as we discover that the cosine similarities between the frames are usually within the range $[0.5, 1.0]$. The cost is minimized when the two frames are structurally similar.

4.2.6 Training. All training images are first normalized and resized to the network input size of 256×256 . Our model is implemented with PyTorch [47]. For model training, we use the Adam optimizer [28] with an initial learning rate of 10^{-4} and a batch size of 128. The network converges after 350 epochs. The whole training process is conducted on a Linux server with a 64-core Intel Xeon CPU, an NVIDIA GeForce 3090 GPU, and 128 GB of memory.

4.3 Statistical Variation

The statistical variation cost C_{stat} measures the variation of aggregated data, assisting users in identifying anomalies or obtaining a statistical summary of the dataset. We consider three aggregation operations: MAX, MIN and AVG. These aggregation operations are commonly used in geospatial analysis, as suggested in our need-finding study. The aggregation can be applied either globally on the entire spatial region of \mathbf{X} , or regionally, focusing on user-specified regions. This cost aligns with our definition DF2 (Anomaly) and DF3 (Extremum), as it encourages the selection of frames representing local extrema when using MAX or MIN aggregations, or anomalies when using AVG aggregation, especially when applied to a local region. Let $\mathbf{v} = \{v_i\}_1^t$ denote the aggregated value at time step i . We first normalize the aggregated value:

$$\hat{v}_i = \frac{v_i - \text{nanmin}_t(\mathbf{v})}{\text{nanmax}_t(\mathbf{v}) - \text{nanmin}_t(\mathbf{v})}, \quad (7)$$

where nanmax and nanmin indicate the maximum and minimum values of \mathbf{v} , excluding NaN values. We then define the statistical variation cost as follows:

$$C_{stat}(i, j) = -\tanh(|\hat{v}_i - \hat{v}_j|) + 1. \quad (8)$$

C_{stat} is designed to be minimized when the aggregated value of two frames is substantially different, thereby encouraging the selection of statistically different frames. This is helpful to identify the local extremum and anomalies of the dataset, as detailed in Section 7.3. The tanh function ensures that the range of cost lies between 0 and 1, in alignment with C_{struc} . When selecting frames, users can adjust α , β and the aggregation method to control the priorities for frame selection, where $\alpha + \beta = 1$.

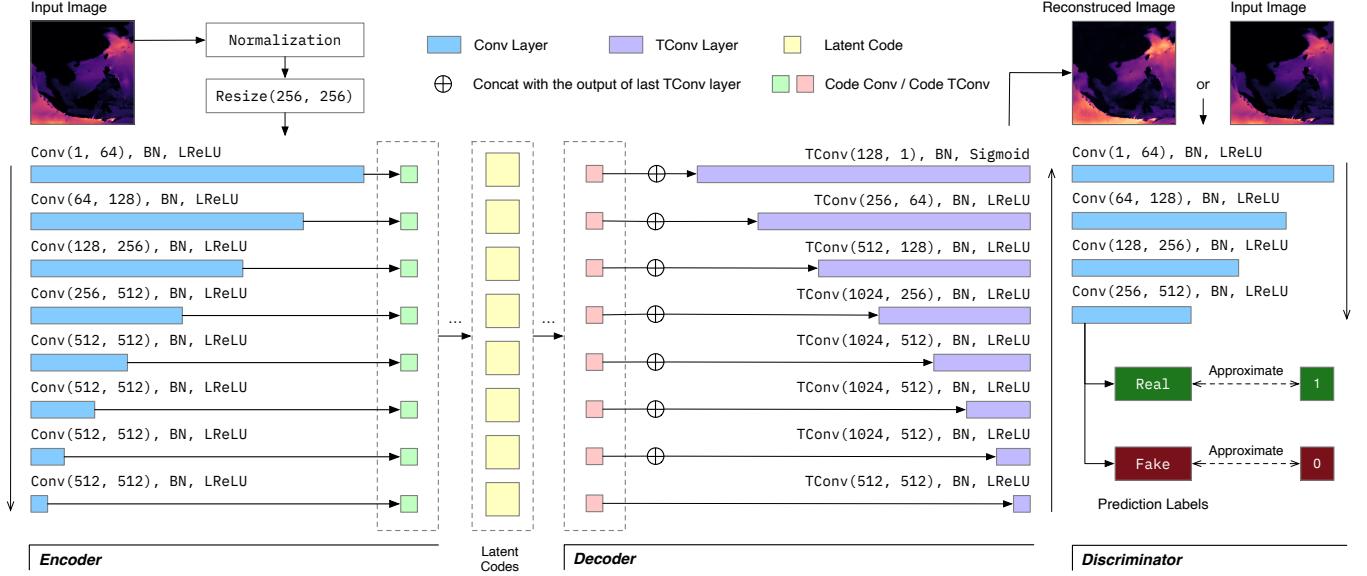


Figure 4: The architecture of our network. The encoder operates from top to bottom while the decoder operates from bottom to top. The flow connecting the output of a decoder layer to the next layer's contact operation is omitted for clarity.

Figure 5 shows an example of cost matrix for C_{struc} and C_{stat} , along with the mixed cost matrix with C_{dis} . It can be observed that the statistical variation, which is the average temperature in this case, exhibits significant changes ahead of the structural features, which encompasses the spatial features of the data. The combination of the two costs allows users to achieve more comprehensive selection results.

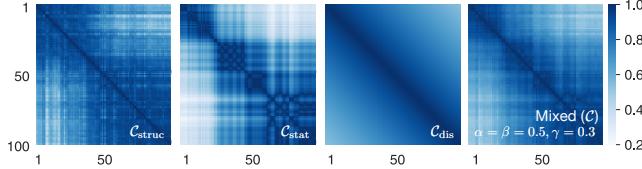


Figure 5: The cost matrix for the first 100 frames from data `tmp2m` (Air Temperature at 2 Meters), using AVG as aggregation method for C_{stat} . Each item (i, j) in the matrix denotes the cost value for frames i and j .

4.4 Distance Penalty

The distance cost C_{dis} is introduced to ensure that the selected frames are well distributed across the temporal domain. The distance cost is defined as follows:

$$C_{\text{dis}}(i, j) = -\gamma \tanh\left(\frac{|i - j|}{\sigma n/k}\right) + 1, \quad (9)$$

where $|i - j|$ is the distance between two frames, n is the total number of frames, k is the number of frames to select, and σ and γ are two hyperparameters. Specifically, γ controls the weight of the distance cost, and σ controls the rate of decay as the distance increases. In our

experiments, we set $\gamma = 0.3$ and $\sigma = 1.0$ to add a slight penalty to the distance between two frames. The distance cost acts as a regulatory term that penalizes the close clustering of selected frames, thereby encouraging a more balanced and informative selection. When solely using the distance cost ($\gamma = 1, \alpha = \beta = 0$), evenly distributed time steps will be selected (Figure 6.d).

The distance cost is essential for preventing the unintended clustering of time steps, a potential drawback of the global optimality in DP. While DP aims to find the optimal solution, it may select consecutive frames when high variation occurs in a short period, as shown in Figure 6.a. This can obscure important temporal patterns. The distance cost mitigates this problem by penalizing consecutive frames, encouraging the selection of more evenly distributed time steps, while still selecting frames with significant variations. As shown in Figure 6.c and d, the distance cost prevented the unintended clustering of frames 66 - 68, while still selecting frames 20 and 21 as salient time steps due to the substantial changes between them.

5 SYSTEM DESIGN

To facilitate interactive exploration of geospatial data and temporal selection, we design and implement a web-based system that integrates our approach with context visualizations. The design requirements from our need-finding study are carefully considered, enabling efficient temporal exploration for geospatial data.

5.1 User Interface

Figure 1 displays the user interface of our system. For clarity, we use bold letters to refer to the labels in Figure 1 unless otherwise specified throughout this section.

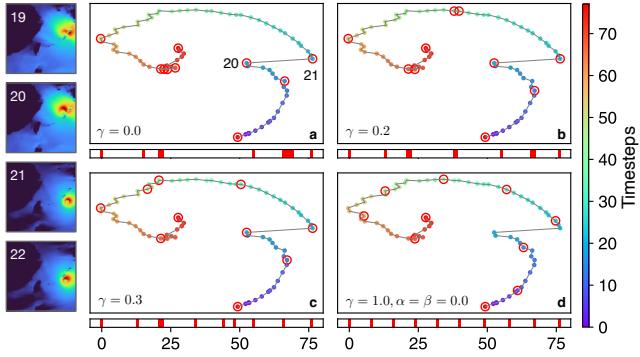


Figure 6: The impact of the distance cost parameter γ on the selection of salient time steps. Subfigures a, b, c, and d show the results with different γ s. Each dot represents a frame's latent code Z_i , dimension-reduced with t-SNE [36] and colored in chronological order. Salient time steps are marked in red circles. On the left are the colorized frames 19 to 22, where drastic changes occur between frames 20 to 21, as reflected in the latent space. The selection is performed on 80 frames of hs (significant height of wind and swell waves) data of the East China Sea area with $k = 10$.

5.1.1 Map View. The map view displays heat maps for geospatial data transformed through color maps, providing users with insights about the current data frame. On mouse hover, the data value corresponding to the coordinates is shown. The frames' global max, min and average values are displayed at the bottom of the Control View (c). Users can select a specific spatial boundary to enter the regional mode for analysis (g), in line with Design Requirement 2 (DR2). When in regional mode, the salient frame calculations and context visualizations are performed within the selected region.

5.1.2 Timeline View. The Timeline View serves as the cornerstone of our system, offering temporal selection and exploration functionalities. At its center is the time series of the dataset. Users first need to drag across a span of time to select the *focus range*, representing the time range they are currently interested in. The selection of salient times will be executed within the range. This design stems from our observation that users typically don't aim to explore the whole available range of the dataset.

Different visual encodings are employed to represent the data frame status, as illustrated in Figure 7. Salient frames are marked in white, while frames visited by users or preloaded are marked in gray. A blinking mark indicates a frame that's currently being loaded. Given that we preload frames adjacent to the salient frames, these loading indicators can help inform users of the loading progress.

To address DR1, we incorporate 2 context visualizations on the timeline. The line chart on the top (d) represents the *Temporal Trend*, indicating the data's temporal variations. The line chart on the bottom (e) represents the *Relative Trend*, indicating the differences of other frames relative to the current frame, updated every time the user navigates to a new frame. Both context visualizations can be configured to display the structural variations or three types of statistical variations.

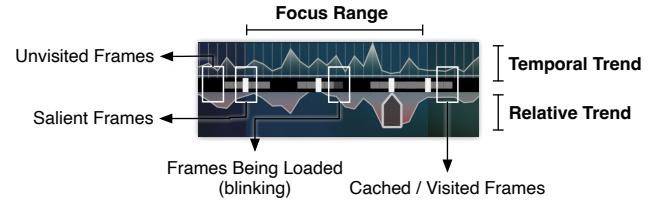


Figure 7: Visual encodings for the frame status.

Three sections of the function panel positioned below the Timeline View (f) provide a comprehensive set of auxiliary tools to assist temporal selection. Figure 8 lists their functionalities. The Navigation section (Figure 8.I) assists users in navigating between time steps, while the Playback section (Figure 8.II) facilitates the creation of time-lapse videos on the map. The interpolated playback utilizes locally cached frames, creating data playback by smoothly transitioning between these frames. The full playback requires full data to be downloaded, potentially incurring significant waiting time. The Refinement section (Figure 8.III) allows users to manually label or unlabel salient frames. When performing salient frame selections, frames manually labeled by the user are guaranteed to be selected, while unlabeled frames will be excluded from the selection result. Both user-labeled and system-generated salient frames are displayed above the timeline for reference (i).

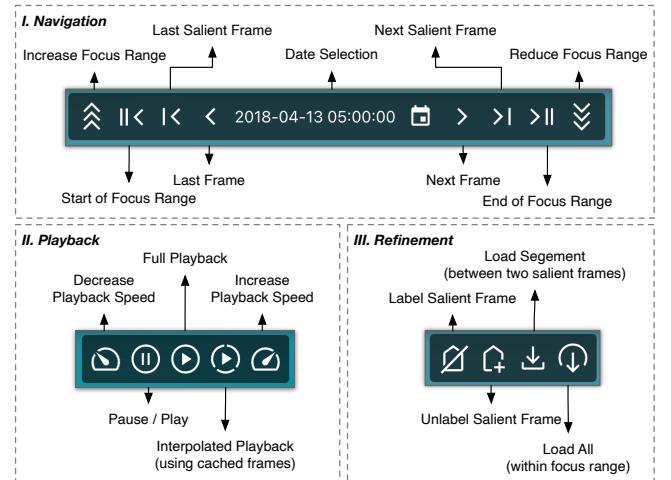


Figure 8: Functions provided by the Timeline View.

5.1.3 Control View. The Control View allows users to select the dataset and specify the parameters α , β , and k for salient frame selection (a). A circular indicator indicates the current parameter states. The visualization of the dimension-reduced latent space (b) within the focus range provides additional structural variation information for the data, assisting users in identifying crucial moments of data variation. This visualization is the same as we provided in Figure 6. To reduce visual clutter, we constrain the maximum number of points visualized in the latent space to 500. If the number

of frames in the focus range exceeds 500, we sample time steps in the latent space evenly. All salient time steps will be displayed, irrespective of the constraint. Salient frames are highlighted here with red circles, aligning with the information on the timeline. Users can navigate to specific frames by clicking on a frame in the latent space. Additionally, users can adjust the colormap and mapping functions at the bottom (c).

5.1.4 Playback View. The Playback View (h) offers additional dynamic information for the data, addressing DR3. It utilizes locally cached data frames to cyclically playback the data within the focus range, allowing users to simultaneously inspect both static and dynamic information. This view is independent of the previously mentioned playback functionalities on the map.

5.2 Implementation

Our system is developed using modern web technologies including React.js, Webpack, and TypeScript. We use D3 [2] for data visualization and Mapbox GL JS for map rendering. We employ the GPU-accelerated t-SNE [4] to accelerate visualization for latent codes. The system is deployed on the same server as described in Section 4.2.6. Addressing DR4, we minimize the computation cost by caching the latent code for different spatial queries and employing paralleled aggregation computation for better performance.

6 CASE STUDY

To validate the efficacy of our system, we apply it to two geospatial datasets: RDI [53] (Red Tide Detection Index) for algal bloom detection and Cloud Moisture Imagery [40] for hurricane analysis. The user study was conducted with Expert 2 (E2) and Expert 5 (E5) from our need-finding study, after which we collected user feedback and comments.

6.1 Algal Bloom

The first case study was conducted with E2. Algal blooms, especially harmful ones (HABs) are one of the major threats to the marine ecosystem and human health. Early detection and warning of HABs is crucial for environmental management and public health. RDI is introduced to monitor HABs from ocean color satellite imageries, employing a three-band blended reflectance model that uses data from MERIS, MODIS and GOCI [53]. The dataset we used is a 3-year time series that covers the Yangtze River Estuary in China, with a spatial resolution of 500 m and a temporal resolution of 5 - 10 days. The dimension of the data is $486 \times 4152 \times 2642$.

Initially, the user employed a smaller zoom level and quickly selected several focus ranges on the timeline using the default number of time steps $k = 12$. The salient frames were generated almost instantly (DR4). Within the following seconds, the salient frames and their adjacent frames were sequentially downloaded. Since a higher RDI value indicates a more severe algal bloom situation, the user added the weight of statistical variation β to 0.2 and set the aggregation to MAX, resulting in a different set of salient frames. Then, the user tried setting β to 1, obtaining the frames that show the peaks and troughs of RDI maximum values. Subsequently, the user reset $\alpha = 0.6$, $\beta = 0.4$ and increased k to 15 to get a more balanced selection (Figure 9.a), navigating between the time steps to obtain data overview.

Then, the user selected MAX on the Temporal Trend above the timeline and observed an annual cyclical trend in the RDI values (Figure 9.b): Algal blooms primarily occur between April and August each year, peaking around June. The user navigated to one of the peak Frames 156 (Figure 9.d), which is a recorded algal bloom event in May 2009. The user then selected *Structural* in the Relative Trend to compare the spatial characteristics of this event with other frames. Afterward, the user zooms in and examines the data around this period. The latent space within this range is shown in Figure 9.c. By observing the patterns in the latent space, the user further navigated data frames at turning points for observation. Eventually, the user chooses to request the full data for frames 950 - 1200.

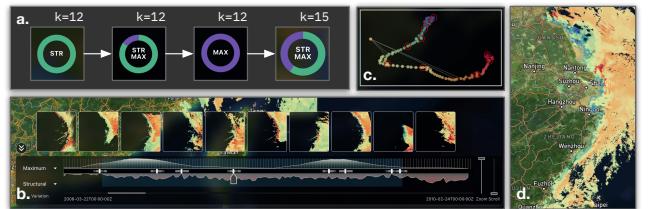


Figure 9: User interaction and results involved in the case study for RDI data. The user's choices for selection parameters are enumerated in (a), with the extent of the green arc in the hollow circle representing the value of α .

The user remarked that our system offers a flexible approach to temporal navigation. With the swift response speed of salient time selection, the user experience enhanced significantly compared with existing data access portals. The user also appreciated the value of contextual visualizations in making informed navigation between timesteps and identifying important events (DR1), and expressed the “*desire to integrate this approach into the workflow of existing systems*”.

6.2 Hurricane

In our second case study with E5, we focus on Hurricane Dorian, a category 5 hurricane that struck the Bahamas and the Southeastern US in 2019. The data was produced by the Advanced Baseline Imager (ABI) instrument from the GOES-16 satellite. We use the Cloud Moisture Imagery Product (CMIP), which is a 16-band dataset that covers the entire Western Hemisphere. We use the Continental US area, Band 13 of the product, which is the Clean Infrared Window, used to monitor clouds, storm systems, and hurricane intensity [40]. We populate data from August to September 2019 in a 20-minute temporal resolution. The dimension of the data is $1653 \times 994 \times 1532$.

The imagery value represents the Top-Of-Atmosphere (TOA) brightness temperature, which drastically decreases in hurricanes and storms due to their deep convection. The user first selected MIN in the temporal trend to view overall brightness temperature changes over the two months, then quickly navigated to the tropical wave on August 24 and the formation of the hurricane on August 28. The user then selected a focus range, finding salient frames in between to discern the evolution of the hurricane eye. Figure 10.d

displays this selection result. The data variations during the early stages of tropical waves are relatively stable, leading to the selection of only two salient frames. As the hurricane rapidly expands and moves northward after its formation, the selections in the latter half of the focus range become much denser.

The user then observed the interpolated preview in the upper right corner, which dynamically displays the hurricane activity (Figure 10.b). Initially, with fewer locally cached frames, the playback relies heavily on the interpolated values, transitioning between cached frames. As the user navigated the timeline and more pre-loaded frames became available, the preview became smoother, revealing the data's original dynamics (DR3). Loading full data for playback took roughly 40 seconds, but the playback was available immediately, progressively loaded and continuously increasing fidelity, eliminating wait times (DR4).

Subsequently, the user selected the spatial boundary above the Bahamas to inspect the hurricane eye and observed that on September 1, at 5 pm, the brightness temperature near the hurricane eye had dropped to 203 Kelvin (Figure 10.c). The user zoomed in on the timeline for a granular view of the hurricane's development within the spatial boundary and navigated to other similar frames with the help of relative trend visualizations (DR2), eventually requesting the download of frames from 800 to 1000.

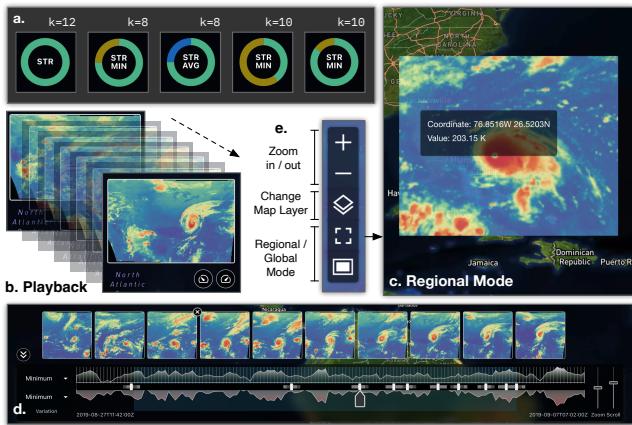


Figure 10: User interaction and results involved in the case study for CMIP data. (d) shows the selection results between August 28 to September 6, with $\alpha = 0.8, \beta = 0.2$, using MIN aggregation. (e) describes the functions provided by the toolbar on the map's top-left corner.

The user pointed out that our interpolated playback mechanism reduces the waiting time and greatly enhances the user experience for creating time-lapse videos. The diverse contextual visualizations, combined with the capability to select spatial boundaries help users in making more informed temporal navigations and identifying important regional events.

7 EVALUATION

In this section, we first conduct quantitative experiments on diverse geospatial datasets, thereby demonstrating our approach's ability to identify salient characterized by summarizability, anomaly and

extremum. Then, we conduct expert interviews to assess the usability of our system compared with existing systems mentioned in Section 3.

7.1 Datasets and Metrics

7.1.1 Datasets. Five datasets are collected for evaluation. Temperature at 2 Meters (tmp2m) and Air Pressure at Mean Sea Level (prmsl) are generated from the Climate Data Assimilation System [71], an observational information reconstructed from historical and present states of earth climate. Significant Wave Height (hs) and Mean Wave Length (lm) are extracted from wave hindcast data from WAVEWATCH III [62] model, where lm data covers East China Sea area and hs covers global range. Cloud and Moisture Imagery Product (cmipc) [40] is the same dataset used in our hurricane case study. Table 2 lists the dimensions, temporal and spatial coverage of the datasets. During the experiment, we are not using the entire time span but using a subset of timesteps as the *focus range*, in alignment with our interactive system.

7.1.2 Method. For each dataset, we conduct two phases of experiments. The first phase evaluates the summarizability of the selected salient time steps compared with even selection and arc-based selection [49]. Even selection sample k time steps uniformly across the whole time span. Arc-based selection selects time steps from the t-SNE dimension reduced latent space via path simplification with Euclidean distance and accumulated angle threshold. For this experiment, we only consider the structural cost by setting $\alpha = 1$ and $\beta = 0$, as the C_{struc} aims to maximize the structural difference between the frames. We quantify the summarizability of selection results by comparing the piece-wise linear interpolated data with the original data. Let selected time steps be $S = \{s_i\}_i^t$ and original data be X , the piece-wise linear interpolated data \tilde{X} is defined as:

$$\tilde{X}_t = \begin{cases} X_t & \text{if } t \in S \\ X_{s_i} + \frac{t-s_i}{s_{i+1}-s_i}(X_{s_{i+1}} - X_{s_i}) & \text{otherwise} \end{cases} \quad (10)$$

Following Porter et al. [49], we use PSNR (in dB) and RMSE to measure the quality of the interpolation. Starting with $k = 5$, we select different k values and compare the interpolated data against the results obtained using even selection. In alignment with practical applications, k does not exceed 30% of the total time steps. Since arc-based selection employs different thresholds to generate salient frames, the number of time steps k is unpredictable. Therefore, we apply a specific threshold (distance threshold $\epsilon = 0.3$, angle threshold $\theta = \pi/4$, mixing factor $\alpha = 0.5$) to get k' frames and use the same k' with our method and even selection for comparison.

The second phase focuses on the impact of statistical variation cost C_{stat} . We set k as a fixed value k' , and progressively adjust β , the weight of C_{stat} from 0 to 1. The results of different β are displayed, alongside the aggregated data value at each time step.

7.1.3 Meaning of the Plots. Figure 11 shows the experiment results. Each row corresponds to a dataset, containing the following subfigures:

- Colorized data image of the selected k' salient times, juxtaposed on the top.
- Selection matrix for k . The horizontal axis represents the data's time steps, and the vertical axis represents different

Table 2: Datasets used for evaluation. Temporal resolution (Res.) indicates the time interval between two consecutive time steps, and coverage (Cov.) indicates the time span of the dataset.

Code	Description	Source	Spatial Region	Dimension ($t \times n \times m$)	Temporal Res. / Cov.
tmp2m	Air Temperature at 2 Meters	CDAS-1 [71]	Global	1588 × 880 × 1760	1 d / 4 years
hs	Significant Height of Wind and Swell Waves	WAVEWATCH III [62]	Global	4632 × 311 × 720	3 h / 2 Years
prmsl	Air Pressure at Mean Sea Level	CDAS-1 [71]	Global	1463 × 361 × 720	6 h / 1 Year
cmipc	Cloud Moisture Imagery, Band 13 (10.3 μm)	GOES-16 [40]	Continental US	1653 × 994 × 1532	20 min / 20 Day
lm	Mean Wave Length	WAVEWATCH III [62]	East China Sea	234 × 551 × 401	3 h / 1 Month

k values. If a time step is selected under that value, the corresponding cell is colored. As a reference, results for even selection are colored in light blue.

- Quality of reconstructed data measured in RMSE and SSIM, where *Struc.* stands for the metrics using our method, considering only the structural cost and *Even* stands for the metrics using even selection.
- Selection matrix for β and the aggregated data values. In this experiment, $k = k'$. The vertical lines in the aggregated data value denote the selected time steps using only the statistical cost ($\alpha = 0, \beta = 1$).
- Dimension-reduced Latent Space, as explained in the caption of Figure 6.

7.2 Summarizability

As shown in the data reconstruction quality metrics, our method's summarizability not only surpasses even selection but also exhibits greater stability. As k increases, even selection often leads to erratic fluctuations in RMSE and SSIM, as observed in all 5 datasets, especially tmp2m and prmsl. In contrast, our method remains stable. This is because even selection fails to select frames that have a significant contribution to the variation of data. Similar frames may be selected, where important variations in between are omitted. In contrast, our structural cost ensures that the selected frames are structurally different from each other, thus preserving data variation.

Table 3: Summarizability of salient time steps selected with different methods. The RMSE and SSIM are calculated between the original data and the piece-wise linear interpolated data.

Dataset	k	Ours		Arc-based [49]		Even	
		RMSE ↓	SSIM ↑	RMSE ↓	SSIM ↑	RMSE ↓	SSIM ↑
tmp2m	10	0.0286	0.9127	0.0326	0.9055	0.0299	0.9111
	20	0.0258	0.9220	0.0260	0.9234	0.0261	0.9241
	40	0.0214	0.9403	0.0207	0.9439	0.0221	0.9430
hs	10	0.0404	0.9064	0.0448	0.8998	0.0476	0.8946
	20	0.0282	0.9430	0.0298	0.9388	0.0301	0.9413
	40	0.0186	0.9727	0.0171	0.9720	0.0229	0.9577
prmsl	10	0.0654	0.9014	0.0716	0.9006	0.1023	0.8892
	20	0.0571	0.9143	0.0578	0.9153	0.0620	0.9135
	40	0.0454	0.9389	0.0459	0.9365	0.0466	0.9373
cmipc	10	0.1205	0.4473	0.1726	0.4459	0.1809	0.4525
	20	0.0916	0.5745	0.0946	0.5696	0.0967	0.5737
	40	0.0800	0.6436	0.0799	0.6546	0.0815	0.6308
lm	10	0.0638	0.9004	0.0699	0.8842	0.0744	0.8799
	20	0.0416	0.9384	0.0517	0.9175	0.0577	0.9189
	40	0.0310	0.9574	0.0390	0.9588	0.0329	0.9624

Compared with the arc-based method, our method has better reconstruction quality, particularly at small k values, as listed in Table 3. When $k = 10$, our method has the lowest RMSE in all datasets. This indicates that our method is very effective when selecting a small number of frames. At higher k values, the arc-based method may achieve slightly better quality in some cases, as in datasets prmsl and lm. This could be due to its trajectory-based selection in the dimension-reduced latent space. While losing some semantic information, it is less prone to selecting trivial frames. In contrast, our selection employs cosine similarity in the original latent space, which captures more detailed information but may suffer from data noises when selecting more frames.

The trajectory in the latent space further demonstrates the efficacy of our approach. Frames located at turning corners of the trajectory are likely to be selected, as these time steps typically mark the beginning or end of a particular variation. For instance, frame 74 in the cmipc captures the onset of the northward movement of the expanding hurricane. Frame 135 in the hs indicates the beginning of a decline in wave height in the southern hemisphere. As k increases, these pivotal frames continue to be selected, while transitions of finer granularity are included, as illustrated in the selection matrix for k .

For certain k values, the reconstruction quality of even selection may match or exceed our method, as seen in the prmsl and lm. However, this is often followed by a sharp decline in reconstruction quality at the subsequent k value. These instances occur when the even selection results overlap with the results of our method. Given that users can not preemptively determine the optimal k , our method offers consistent and reliable selection results.

7.3 Anomaly and Extremum

The selection matrix for β shows the effect of the statistical cost C_{stat} . Compared with the structural cost, C_{stat} produces more predictable results by emphasizing the changes in maximum, minimum or average values, capturing the anomalies and extrema in the datasets. This can be observed from the line chart of aggregated values when $\beta = 1.0$. For instance, when employing the AVG aggregation on the tmp2m data, several inflection points in the average values are selected. The overall temperature exhibits a trend of initial rise followed by a decline, and our method effectively selected the time steps that mark these points, together with the local fluctuations. In the hs data, users are concerned about the peak wave heights and by using the MAX aggregation, the selected time steps adeptly showcase the evolution of height extrema.

This selection method can also assist users in identifying anomalies in the data. For instance, in the cmipc dataset, frame 39 is

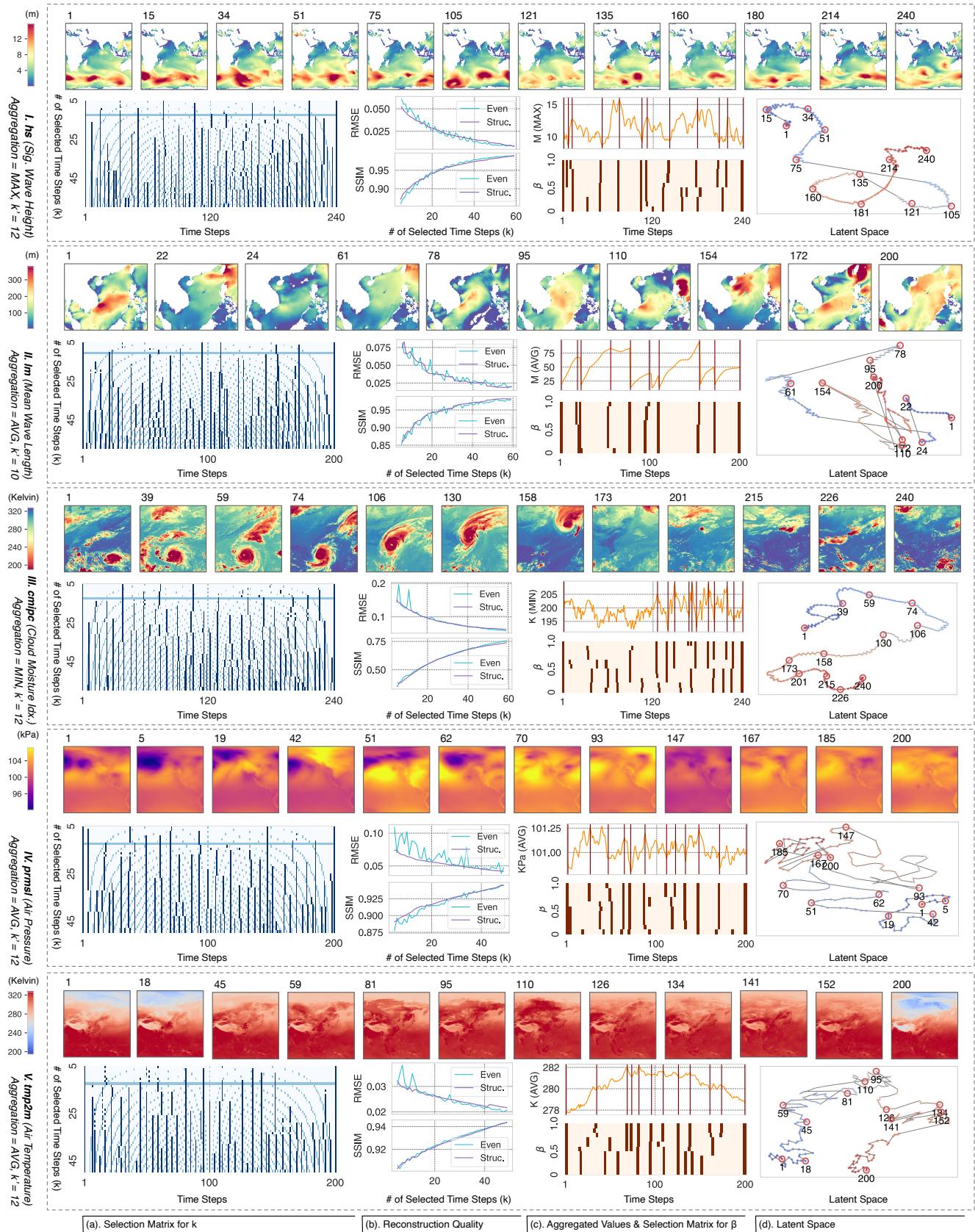


Figure 11: Experiment Results. The meaning of the plots is detailed in Section 7.1.3.

selected, where the brightness temperature of this frame is abnormally low compared to surrounding frames. In the lm dataset, a sharp decline follows frame 154. Upon inspection, we discovered that 2 days of data were missing following this frame due to data quality issues.

Adjustable weights provide users with higher flexibility in selecting salient time steps. As shown in the selection matrix for β , when β transitions from 0 to 1, the selection results gradually shift from prioritizing structural variation to emphasizing statistical variation. This is particularly helpful when the datasets have frames with similar spatial structures but distinct values. For example, frames 167 and 200 in the prms1 dataset are proximate in the latent space, indicating their similar spatial distributions. However, their average values differ by 2.4 hPa. In practice, users can specify their priorities to obtain the selections that align with their expectations.

7.4 Runtime Performance

Computing salient time steps can be divided into three stages: 1). Pre-processing: Reading the pre-stored data memory array, computing the aggregated values and cost function values in advance; 2). Latent Code Generation: Feeding the data within spatial boundaries and focus ranges to the autoencoder to obtain latent codes; 3). Frame Selection: Employing DP to select the optimal salient time steps. The time used for extracting data from NetCDF and Grib data is not considered as it needs to be executed only once for any dataset and is not perceived by users. The experiment is tested on the same server for network training as described in Section 4.2.6.

Table 4 shows the first two stages' running time for each dataset. As shown in the table, even for datasets with thousands of frames, pre-processing and latent code generation can be completed within seconds. The running time for the third stage as k or the length of focus range T increases is plotted in Figure 12. The DP algorithm has a time complexity of $O(T^2k)$, being the dominant contributor to the total running time. We can see that the running time increases linearly with k and quadratically with T . For cases where T is under 2k frames, the selection of 24 salient time steps is completed under a second. Typically, the focus range is limited to several hundred frames, and k rarely exceeds 20 frames. Under these conditions, the computational duration is below 300 ms, making it virtually instantaneous and supporting interactive selection.

Table 4: Running time for the first two stages. Pre. stands for pre-processing. Latent. stands for latent code generation. / F stands for the average time used for processing a single frame.

Dataset	Pre. (s)	Pre. / F (ms)	Latent. (s)	Latent. / F (ms)
tmp2m	1.134	0.714	4.660	2.935
hs	7.040	1.520	7.029	1.517
prms1	1.838	1.256	2.250	1.538
cmipc	2.675	1.651	4.851	2.994
lm	0.197	0.842	0.440	1.880

The results of the first two stages can be cached between time selection requests, as long as the requests' spatial boundary remains unchanged. This is because the aggregation calculation depends on it. When users select a new boundary or switch datasets, they only

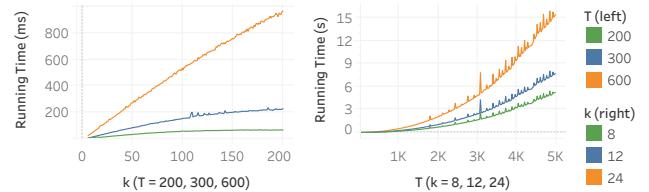


Figure 12: Running time of the DP process as k or length of focus range T increases.

face a few seconds of loading time before being able to perform frequent time selection interactions.

7.5 Expert Interview

To assess the usability of our system compared with existing tools mentioned in Section 3, we conducted an expert interview with experts in our need-finding study (E1 - E5). We employed an adapted System Usability Scale (SUS) [3], a widely recognized tool that offers a comprehensive evaluation of a system's usability through 10 questions, each rated on a 5-point Likert scale. We slightly adjusted the wording to make the scale adapt to our need to focus on the data exploration process in web-based geospatial systems.

Each interview was conducted offline and lasted approximately 60 minutes. It began with a 5-minute introduction, instructing the experts to emphasize temporal navigation and selection over other system features. As there was a three-month gap since the prior study, we first let experts revisit their prior interview notes and selection tasks, allotting 10 minutes for them to familiarize themselves with these systems. Then, we let the experts engage with three systems sequentially, each for 5 to 10 minutes, followed by a SUS questionnaire for that system. After that, we introduced the features and UI components of Salientime and let the experts freely explore the system, performing identical temporal selection and download tasks as they did in the previous systems, as listed in the Outcome section of Figure 3. After 15 minutes, we asked them to fill out the SUS questionnaire for our system and gathered their feedback and overall opinions of our system in terms of usability and functionality.

Figure 13 shows the results of the questionnaire, with the usability score for each question normalized to a 0 - 10 scale. Overall, Salientime achieved a good usability score of 85.5, outperforming the other three systems (77.5, 62.5, and 74.5).

Salient Frames and Temporal Context. Experts remarked on the significance of selected salient frames in identifying crucial data variations, thus aiding decision-making about subsequent selections. Compared to existing systems that solely provide a timeline (Worldview, MetoceanView) or a date picker for inputting data (EO Browser), our timeline design that highlights salient time steps and incorporates contextual visualization offers enhanced usability in their workflows. Most experts (2 strongly agree, 3 agree, Q9) agreed that contextual visualizations on the timeline make them more confident when interacting with the system, and easier to perform temporal navigations (3 strongly agree, 2 agree, Q3).

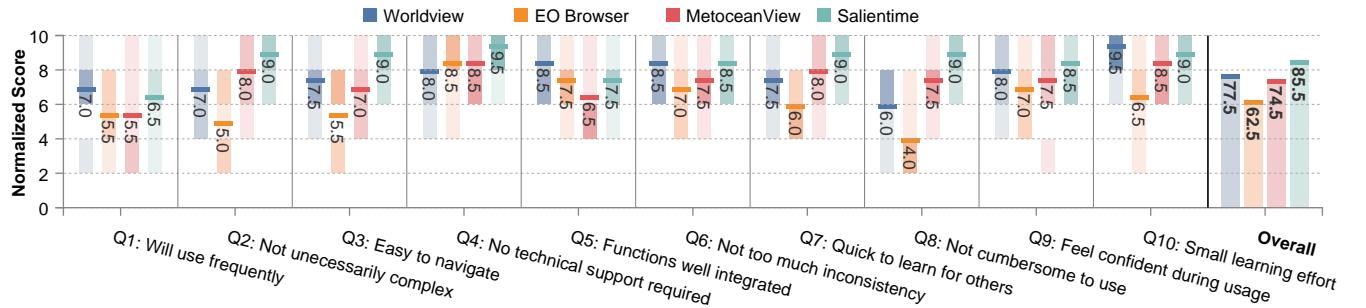


Figure 13: Usability questionnaire results. For consistency, we reverse the points for negative questions in the SUS scale and display the normalized score. The shades of the background color indicate the number of experts who selected that option.

Spatial Features. The dimension-reduced latent space provides additional insight into data variation trends. As a novel approach for providing data overviews compared with existing systems, experts were particularly interested in this view and frequently inspected the frames at the turning points of the latent space trajectory. The combination of the contextual visualization on the timeline and the latent space visualization offers a comprehensive understanding of data variation and cycle information. The integration of this view with other system functions is well-received (1 strongly agree, 3 agree, Q5). Additionally, E1 and E2 predominantly utilized the regional mode to examine specific regions of the data, emphasizing the importance and usefulness of the regional selection features in geospatial systems.

Progressive Loading and Performance. Prioritizing the loading of salient frames and their surrounding frames improved the overall performance of our system (3 strongly agree, 2 agree, Q8). Unlike traditional methods that start frame data download only upon user navigation to a new frame, our approach significantly reduces the user-perceived latency. These salient frames also serve as the initial state for progressive time-lapse video generation. With more frames being loaded, the dynamic information becomes more accurate. This process enables users to rapidly access data dynamics, avoiding the extensive buffer loading time required by Worldview. Experts E3 and E4 recognize this as a useful engineering technique to enhance user experience.

Learning Effort. We also consulted the learning effort of our system, as we introduced extra adjustable parameters α , β , and aggregation during selection. All experts agreed that this selection process was straightforward (3 strongly agree, 2 agree, Q7) and did not require external technical support (4 strongly agree, 1 agree). After a brief introduction to these parameters during the interview, experts can swiftly comprehend their meaning and adjust parameters to fit their needs. Moreover, E4 appreciated the flexibility provided by these parameters, remarking that the impact of different parameters is reasonable, making our system suitable for various tasks. However, E3 highlighted the need for a more detailed explanation of the temporal contexts on the timeline to ensure comprehensive understanding.

Overall, the interview result indicates a high usability of our system. With automatically selected salient frames with user-specified priorities and multiple contextual visualizations indicating data variations, our system provides easier temporal navigation, faster response time and acceptable learning effort, greatly enhancing the user experience for temporal navigation and data exploration of geospatial data.

8 DISCUSSION

8.1 Parameter Selection

We allow users to specify the weight for C_{struc} and C_{stat} in the cost function. However, the weight for distance cost is fixed. We believe the weight for the distance cost γ should not be a user concern, as it tends to guide the selection towards even selection and should be set based on data characteristics. From our observations, $\gamma = 0.3$ serves as a general value for the distance cost. Under such value, any attempt to select two consecutive frames with a structural similarity higher than 0.7 is highly likely to be rejected. Therefore, in datasets with more drastic changes, a larger γ value might be needed to prevent clustering of selection results, while in milder cases, $\gamma = 0.1$ may suffice.

As for the dimension of the latent space ϕ , it determines the feature extraction capability of our network. While a larger ϕ may offer more expressive latent codes, it also increases the risk of overfitting by learning the noise in the data. In contrast, a smaller ϕ may not suffice to encode the data, compromising its generalizability across geospatial datasets. Through multiple experiments, we choose $\phi = 512$ in this trade-off. Nonetheless, we believe it's worth exploring other sizes of ϕ , depending on the volume and complexity of the dataset.

8.2 Complexity and Scalability

As discussed in Section 7.4, the computation is primarily divided into three stages. Both Pre-processing and Latent Code Generation have a time complexity of $O(n)$, where n is the total number of frames. The DP process with a time complexity of $O(T^2k)$ is the bottleneck of our method. Although our experiments indicate fast running time in selections within 2k frames, when T exceeds 5k and k exceeds 20, the waiting time becomes quite lengthy. This can be

seen in datasets spanning several decades or those with high temporal resolution. To scale to these situations, several strategies can be considered. One approach is to pre-compute frames with coarser temporal resolution, reducing the size of T in a stepped manner. Alternatively, the data can be divided into p smaller segments, and the DP process can be run in parallel on each segment, selecting $\lfloor p/k \rfloor$ frames per segment. Additionally, approximate methods of DP that sacrifice global optimality, such as the multi-pass algorithm proposed by Zhou et al. [76], can be considered.

8.3 Limitations and Future Work

First, we currently focus on the raster model of geospatial data, while vector data such as wind direction or ocean currents are equally important. Meanwhile, the interrelationships between variables in multivariable geospatial data remain unexplored. Extending our approach to vector data and multivariable datasets is a potential direction for future research. Second, the DP process with a time complexity of $O(T^2k)$ poses challenges when applied to datasets with more than 5k time steps. Investigating more efficient or approximate methods could enhance the scalability of our approach. Third, there is an inherent loss of spatial continuity when we select a subset of data frames from the full data. To further mitigate this loss, we could incorporate compression-based data transfer that enables the simultaneous transfer of surrounding frames alongside the salient frames. Visual encodings that indicate the extent of information loss can be beneficial as well. Lastly, there's potential to incorporate more comprehensive contextual visualizations. For instance, visualizations that assist users in identifying cyclical patterns in the data, or those that combine temporal and spatial trajectories. Such tools can enable users to make more informed selections.

9 CONCLUSION

In this paper, we offer a new approach and interaction workflow for time step selection in geospatial systems, aiming to enhance user experience and maximize the research value of large-scale geospatial data. Our method is built upon an extensive need-finding study with domain experts, through which we observe their time selection workflows, identify their needs, and establish a multifaceted definition for salient time steps. Leveraging autoencoders and dynamic programming, we select time steps with user-specified priorities in terms of structural and statistical variation. Moreover, we design and implement a web-based interactive system to facilitate context-aware exploration and selection of time steps. We present two case studies to showcase the efficacy of our system and evaluate the effectiveness of our methods on diverse geospatial datasets. The source code is available at <https://github.com/billchen2k/salientime>.

ACKNOWLEDGMENTS

This work was supported by the NSFC under Grants (No. 61802128 and 62072183), the Shanghai Committee of Science and Technology, China (Grant No. 22511104600), and the Yangtze River Delta Science and Technology Innovation Community Project, China (Grant No. 23002400400). Chenhui Li is the corresponding author.

REFERENCES

- [1] Hiroshi Akiba, Nathaniel Fout, and Kwan-Liu Ma. 2006. Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram. In *EUROVIS - Eurographics/IEEE VGTC Symp. Visualization*. <https://doi.org/10.2312/VisSym/EuroVis06/171-178>
- [2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D³ Data-Driven Documents. *IEEE Trans. Vis. Comput. Graphics* 17 (2011), 2301–2309. Issue 12. <https://doi.org/10.1109/TVCG.2011.185>
- [3] J. B. Brooke. 1996. SUS: A 'quick and Dirty' Usability Scale.
- [4] David M Chan, Roshan Rao, Forrest Huang, and John F Canny. 2019. GPU Accelerated T-Distributed Stochastic Neighbor Embedding. *J. Parallel Distrib. Com.* 131 (2019), 1–13. <https://doi.org/10.1016/j.jpdc.2019.04.008>
- [5] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. 2008. Importance-Driven Time-Varying Data Visualization. *IEEE Trans. Vis. Comput. Graphics* 14, 6 (Nov. 2008), 1547–1554. <https://doi.org/10.1109/TVCG.2008.140>
- [6] Juntong Chen, Qiaoyun Huang, Changbo Wang, and Chenhui Li. 2023. SenseMap: Urban Performance Visualization and Analytics via Semantic Textual Similarity. *IEEE Trans. Vis. Comput. Graphics PP* (2023), 1–15. Issue 99. <https://doi.org/10.1109/TVCG.2023.333356>
- [7] Xueyi Chen, Liming Shen, Ziqi Sha, Richen Liu, Siming Chen, Genlin Ji, and Chao Tan. 2019. A Survey of Multi-Space Techniques in Spatio-Temporal Simulation Data Visualization. *Visual Informatics* 3, 3 (Sept. 2019), 129–139. <https://doi.org/10.1016/j.visinf.2019.08.002>
- [8] Xingdong Deng, Penghua Liu, Xiaoping Liu, Ruoyu Wang, Yuanjing Zhang, Jialv He, and Yao Yao. 2019. Geospatial Big Data: New Paradigm of Remote Sensing Applications. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 12, 10 (Oct. 2019), 3841–3851. <https://doi.org/10.1109/JSTARS.2019.2944952>
- [9] Zikun Deng, Di Weng, Jiahui Chen, Ren Liu, Zhibin Wang, Jie Bao, Yu Zheng, and Yingcai Wu. 2019. AirVis: Visual Analytics of Air Pollution Propagation. *IEEE Trans. Visual. Comput. Graphics* (2019), 800–810. <https://doi.org/10.1109/TVCG.2019.2934670>
- [10] Zikun Deng, Di Weng, Shuhan Liu, Yuan Tian, Mingliang Xu, and Yingcai Wu. 2023. A Survey of Urban Visual Analytics: Advances and Future Directions. *Comp. Visual Media* 9, 1 (March 2023), 3–39. <https://doi.org/10.1007/s41095-022-0275-7>
- [11] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini. 2012. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* 120 (May 2012), 25–36. <https://doi.org/10.1016/j.rse.2011.11.026>
- [12] Soumya Dutta, Humayra Tasnim, Terece L. Turton, and James Ahrens. 2021. In Situ Adaptive Spatio-Temporal Data Summarization. In *2021 IEEE International Conference on Big Data (Big Data)*, 315–321. <https://doi.org/10.1109/BigData52589.2021.9671581>
- [13] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler. 2012. Waiting Times in Quality of Experience for Web Based Services. In *2012 Fourth Int. Workshop Qual. Multimedia Exp.*, 86–96. <https://doi.org/10.1109/QoMEX.2012.6263888>
- [14] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recogn.* 58 (2016), 121–134. <https://doi.org/10.1016/j.patcog.2016.03.028>
- [15] ESA (European Space Agency). 2022. Copernicus Sentinel Data Access Annual Report 2022. <https://sentinels.copernicus.eu/web/sentinel/-/copernicus-sentinel-data-access-annual-report-2022>
- [16] ESA (European Space Agency). 2023. Copernicus Open Access Hub. <https://scihub.copernicus.eu/dhus>
- [17] S. Frey and T. Ertl. 2017. Flow-Based Temporal Selection for Interactive Volume Visualization. Flow-Based Temporal Selection for Interactive Volume Visualization. *Comput. Graph. Forum* 36, 8 (Dec. 2017), 153–165. <https://doi.org/10.1111/cgf.13070>
- [18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Adv. Neural Inf. Process. Syst. (NIPS)*. <https://doi.org/10.1145/3426222>
- [19] Tao Guo, Kaiyu Feng, Gao Cong, and Zhifeng Bao. 2018. Efficient Selection of Geospatial Data on Maps for Interactive and Visualized Exploration. In *2018 Int. Conf. Manag. Data. (ICDE)*. ACM, Houston TX USA, 567–582. <https://doi.org/10.1145/3183713.3183738>
- [20] Jun Han, Jun Tao, and Chaoli Wang. 2020. FlowNet: A Deep Learning Framework for Clustering and Selection of Streamlines and Stream Surfaces. *IEEE Trans. Vis. Comput. Graphics* 26, 4 (April 2020), 1732–1744. <https://doi.org/10.1109/TVCG.2018.2880207>
- [21] Geoffrey E. Hinton and Ruslan Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313 (2006), 504–507. <https://doi.org/10.1126/science.1127647>
- [22] Yifei Huang, Chenhui Li, Xiaohu Guo, Jing Liao, Chenxu Zhang, and Changbo Wang. 2020. DeSmoothGAN: Recovering Details of Smoothed Images via Spatial Feature-wise Transformation and Full Attention. In *Proc. ACM Int. Conf. Mult.*

- (MM '20). Association for Computing Machinery, New York, NY, USA, 2655–2663. <https://doi.org/10.1145/3394171.3413958>
- [23] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. Int. Conf. Mach. Learn. (ICML)*.
- [24] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. 2019. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 7842–7851. <https://doi.org/10.1109/cvpr.2019.00083>
- [25] Alan Julian Izenman. 2013. Linear discriminant analysis. In *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer, 237–280. https://doi.org/10.1007/978-0-387-78189-1_8
- [26] Shiqi Jiang, Chenhui Li, Lei Wang, Yanpeng Hu, and Changbo Wang. 2021. LatentMap: Effective Auto-Encoding of Density Maps for Spatiotemporal Data Visualizations. *Graph. Vis. Comput.* 4 (June 2021), 200019. <https://doi.org/10.1016/j.gvc.2021.200019>
- [27] Ian T Jolliffe and BJT Morgan. 1992. Principal component analysis and exploratory factor analysis. *Statistical methods in medical research* 1, 1 (1992), 69–95. <https://doi.org/10.1177/096228029200100105>
- [28] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *Clin. Orthop. (CoRR)* abs/1412.6980 (2014).
- [29] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *Clin. Orthop. abs/1312.6114* (2013).
- [30] Johannes Knittel, Steffen Koch, Tan Tang, Wei Chen, Yingcui Wu, Shixia Liu, and Thomas Ertl. 2022. Real-Time Visual Analysis of High-Volume Social Media Posts. *IEEE Trans. Vis. Comput. Graphics* 28, 1 (Jan. 2022), 879–889. <https://doi.org/10.1109/TVCG.2021.3114800>
- [31] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42 (2009). <https://doi.org/10.1109/mc.2009.263>
- [32] Jae-Gil Lee and Minseo Kang. 2015. Geospatial Big Data: Challenges and Opportunities. *Big Data Res.* 2, 2 (June 2015), 74–81. <https://doi.org/10.1016/j.bdr.2015.01.003>
- [33] Dongyu Liu, Panpan Xu, and Liu Ren. 2019. TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis. *IEEE Trans. Vis. Comput. Graphics* 25 (2019), 1–11. Issue 1. <https://doi.org/10.1109/TVCG.2018.2865018>
- [34] Yang Liu, Yutong Lu, Yueqing Wang, Dong Sun, Liang Deng, Yunbo Wan, and Fang Wang. 2019. Key Time Steps Selection for CFD Data Based on Deep Metric Learning. *Comput. Fluids* 195 (Dec. 2019), 104318. <https://doi.org/10.1016/j.compfluid.2019.104318>
- [35] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. (2013).
- [36] L. Maaten and Geoffrey E. Hinton. 2008. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* (2008).
- [37] L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv e-prints* (Feb. 2018). arXiv:1802.03426 [stat.ML]
- [38] Grant McKenzie, Sarah Battersby, and Vidya Setlur. 2023. MixMap: A User-Driven Approach to Place-Based Semantic Similarity. *Cartogr. Geogr. Inf. Sc.* 0, 0 (March 2023), 1–16. <https://doi.org/10.1080/15230406.2023.2176930>
- [39] MetOcean Solutions. 2008. MetOceanView. <https://dataspace.copernicus.eu/browser>
- [40] NASA (National Aeronautics and Space Administration). 2016. GOES-16 Band Reference Guide.
- [41] NASA (National Aeronautics and Space Administration). 2023. NASA Earthdata. <https://www.earthdata.nasa.gov/homepage>
- [42] NASA (National Aeronautics and Space Administration). 2023. NASA Worldview. <https://worldview.earthdata.nasa.gov/>
- [43] NOAA (National Oceanic and Atmospheric Administration). 2022. NOAA Environmental Modeling Center. <https://polar.ncep.noaa.gov/waves/index.php>
- [44] NOAA (National Oceanic and Atmospheric Administration). 2023. NOAA View Global Data Explorer. <https://www.nvnl.noaa.gov/view/globaldata.html>
- [45] Shaoming Pan, Yanwen Chong, Hang Zhang, and Xicheng Tan. 2017. A Global User-Driven Model for Tile Prefetching in Web Geographical Information Systems. *PLOS ONE* 12, 1 (Jan. 2017), e0170195. <https://doi.org/10.1371/journal.pone.0170195>
- [46] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh S. Vempala. 1998. Latent semantic indexing: a probabilistic analysis. *J. Comput. Syst. Sci.* 61 (1998), 217–235. <https://doi.org/10.1006/jcss.2000.1711>
- [47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst. (NIPS)* 32 (2019).
- [48] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering.
- [49] William P. Porter, Yunhao Xing, Blaise R. Von Ohlen, Jun Han, and Chaoli Wang. 2019. A Deep Learning Approach to Selecting Representative Time Steps for Time-Varying Multivariate Data. In *2019 IEEE Visualization Conf. (VIS)*. IEEE, Vancouver, BC, Canada, 1–5. <https://doi.org/10.1109/VISUAL.2019.8933759>
- [50] Jesus Pulido, John Patchett, Manish Bhattachari, Boian Alexandrov, and James Ahrens. 2021. Selection of Optimal Salient Time Steps by Non-negative Tucker Tensor Decomposition. In *EuroVis 2021 - Short Papers*. The Eurographics Association. <https://doi.org/10.2312/evs.20211055>
- [51] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *Clin. Orthop. abs/1511.06434* (2015). <https://doi.org/10.23919/chicc.2018.8482813>
- [52] SentinelHub. 2020. Sentinel Hub EO Browser. <https://apps.sentinel-hub.com/eo-browser/>
- [53] Fang Shen, Rugang Tang, Xuerong Sun, and Dongyan Liu. 2019. Simple Methods for Satellite Identification of Algal Blooms and Species Using 10-Year Time Series Data from the East China Sea. *Remote Sens. Environ.* 235 (Dec. 2019), 111484. <https://doi.org/10.1016/j.rse.2019.111484>
- [54] Qiaomu Shen, Wei Zeng, Yu Ye, Stefan Müller Arisona, Simon Schubiger, Remo Burkhardt, and Huamin Qu. 2018. StreetVizor: Visual Exploration of Human-Scale Urban Forms Based on Street Views. *IEEE Trans. Visual. Comput. Graphics* 24, 1 (Jan. 2018), 1004–1013. <https://doi.org/10.1109/TVCG.2017.2744159>
- [55] B. Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. IEEE Symp. Visual Lang.* 336–343. <https://doi.org/10.1109/VL.1996.545307>
- [56] Pushpendra Singh, Shiv Dutt Joshi, Rakesh Kumar Patney, and Kaushik Saha. 2016. Fourier-Based Feature Extraction for Classification of EEG Signals Using EEG Rhythms. *Circuits Syst. Signal Process.* 35 (2016), 3700–3715. <https://doi.org/10.1007/s00034-015-0225-z>
- [57] Bedionita Soro and Chaewoo Lee. 2019. A Wavelet Scattering Feature Extraction Approach for Deep Neural Network Based Indoor Fingerprinting Localization. *Sensors* 19 (2019). <https://doi.org/10.3390/s19081790>
- [58] Martina Stockhausen. 2013. User Driven Data Access Mechanisms. In *Earth System Modelling - Volume 6: ESM Data Archives in the Times of the Grid*. Wolfgang Hiller, Reinhard Budich, and René Redler (Eds.). Springer, Berlin, Heidelberg, 33–47. https://doi.org/10.1007/978-3-642-37244-5_5
- [59] Duyu Tang, Furu Wei, Nan Yang, M. Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)*. <https://doi.org/10.3115/v1/p14-1146>
- [60] Teng-Yok Lee and Han-Wei Shen. 2009. Visualization and Exploration of Temporal Trend Relationships in Multivariate Time-Varying Data. *IEEE Trans. Vis. Comput. Graphics* 15, 6 (2009), 1359–1366. <https://doi.org/10.1109/TVCG.2009.200>
- [61] Teng-Yok Lee and Han-Wei Shen. 2009. Visualizing Time-Varying Features with TAC-based Distance Fields. In *IEEE Pacific Visual. Symp. (PacificVIS)*. IEEE, Beijing, China, 1–8. <https://doi.org/10.1109/PACIFICVIS.2009.4906831>
- [62] H. Tolman, M. Accensi, H. Alves, F. Arduhin, J. Bidlot, N. Booij, A. Bennis, Tim Campbell, D. Chalikov, Arun Chawla, J. Filipot, M. Foreman, P. Janssen, F. Leckler, Jian-Guo Li, K. Lind, M. Orzech, R. Padilla-Hernández, W. Rogers, A. Rawat, A. Roland, M. D. Sikiric, M. Szyszka, B. Tracy, G. V. Vledder, A. J. Westhuyzen, and S. Zieger. 2014. *User Manual and System Documentation of WAVEWATCH III (R) Version 4.18*. NOAA (National Oceanic and Atmospheric Administration).
- [63] Xin Tong, Teng-Yok Lee, and Han-Wei Shen. 2012. Salient Time Steps Selection from Large Scale Time-Varying Data Sets with Dynamic Time Warping. In *IEEE Symp. Large Data Anal. Visualization (LDAV)*, 49–56. <https://doi.org/10.1109/LDAV.2012.6378975>
- [64] Carol Traynor and Marian G. Williams. 1995. Why Are Geographic Information Systems Hard to Use? In *Proc. 1995 CHI Conf. Human Factors Comput. Syst.* ACM Press, Denver, Colorado, United States, 288–289. <https://doi.org/10.1145/223355.223678>
- [65] U.S. Geological Survey. 2023. USGS EarthExplorer. <https://earthexplorer.usgs.gov/>
- [66] Pascal Vincent, H. Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proc. Int. Conf. Mach. Learn. (ICML)*. <https://doi.org/10.1145/1390156.1390294>
- [67] J. Woodring and Han-Wei Shen. 2009. Multiscale Time Activity Data Exploration via Temporal Clustering Visualization Spreadsheet. *IEEE Trans. Vis. Comput. Graphics* 15, 1 (Jan. 2009), 123–137. <https://doi.org/10.1109/TVCG.2008.69>
- [68] Mengxi Wu, Yi-Jen Chiang, and Christopher Musco. 2022. Streaming Approach to In Situ Selection of Key Time Steps for Time-Varying Volume Data. *Comput. Graph. Forum* 41, 3 (June 2022), 309–320. <https://doi.org/10.1111/cgf.14542>
- [69] Chaowei Yang, Robert Raskin, Michael Goodchild, and Mark Gahegan. 2010. Geospatial Cyberinfrastructure: Past, Present and Future. *Comput. Environ. Urban Syst.* 34, 4 (July 2010), 264–277. <https://doi.org/10.1016/j.compenurbysys.2010.04.001>
- [70] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, and Yun Li. 2017. Utilizing Cloud Computing to Address Big Geospatial Data Challenges. *Comput. Environ. Urban Syst.* 61 (Jan. 2017), 120–128. <https://doi.org/10.1016/j.compenurbysys.2016.10.010>
- [71] Xiaosong Yang. 2010. Climate Data Assimilation. <https://www.gfdl.noaa.gov/climate-data-assimilation/>

- [72] Xiaochuang Yao, Guoqing Li, Junshi Xia, Jin Ben, Qianqian Cao, Long Zhao, Yue Ma, Lianchong Zhang, and Dehai Zhu. 2019. Enabling the Big Earth Observation Data via Cloud Computing and DGGS: Opportunities and Challenges. *Remote Sens.* 12, 1 (Dec. 2019), 62. <https://doi.org/10.3390/rs12010062>
- [73] Huayuan Ye, Chenhui Li, Yang Li, and Changbo Wang. 2023. InvVis: Large-Scale Data Embedding for Invertible Visualization. *IEEE Trans. Vis. Comput. Graphics* PP (2023), 1–11. Issue 99. <https://doi.org/10.1109/tvcg.2023.3326597> arXiv:2307.16176
- [74] Yi Gu and Chaoli Wang. 2011. TransGraph: Hierarchical Exploration of Transition Relationships in Time-Varying Volumetric Data. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (Dec. 2011), 2015–2024. <https://doi.org/10.1109/TVCG.2011.246>
- [75] Peiying Zhang, Chenhui Li, and Changbo Wang. 2021. VisCode: Embedding Information in Visualization Images Using Encoder-Decoder Network. *IEEE Trans. Vis. Comput. Graphics* 27 (2021), 326–336. Issue 2. <https://doi.org/10.1109/TVCG.2020.3030343> arXiv:2009.03817
- [76] Bo Zhou and Yi-Jen Chiang. 2018. Key Time Steps Selection for Large-Scale Time-Varying Volume Datasets Using an Information-Theoretic Storyboard. *Comput. Graph. Forum* 37, 3 (June 2018), 37–49. <https://doi.org/10.1111/cgf.13399>
- [77] Zhiguang Zhou, Fengling Zheng, Jin Wen, Yuanyuan Chen, Xinyu Li, Yuhua Liu, Yigang Wang, and Wei Chen. 2023. A User-Driven Sampling Model for Large-Scale Geographical Point Data Visualization via Convolutional Neural Networks. *IEEE Trans. Human-Mach. Syst.* (2023), 1–10. <https://doi.org/10.1109/THMS.2023.3296692>
- [78] Parker Ziegler and Sarah E. Chasins. 2023. A Need-Finding Study with Users of Geospatial Data. In *Proc. 2023 CHI Conf. Human Factors Comput. Syst.* ACM, Hamburg Germany, 1–16. <https://doi.org/10.1145/3544548.3581370>