RISE: A Real-time and Interactive Spatiotemporal Kernel Density Visualization System with Streaming and Static Data

Yue Zhong
2310273008@email.szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China

Tsz Nam Chan*
edisonchan@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China

Leong Hou U
ryanlhu@um.edu.mo
Department of Computer and Information Science, University of Macau
Macao, China

Dingming Wu
dingming@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China

Wei Tu
Ruisheng Wang
{tuwei, ruiswang}@szu.edu.cn
School of Architecture and Urban Planning, Shenzhen University
Shenzhen, China

Joshua Zhexue Huang
zx.huang@szu.edu.cn
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China

Abstract

Spatiotemporal Kernel Density Visualization (STKDV) is widely used for analyzing spatiotemporal data in applications such as urban mobility, disease outbreak monitoring, crime hotspot detection, and traffic accident analysis. Existing geographic information systems and software packages cannot generate STKDV in real time on large datasets, provide limited support for interactive exploration, and cannot handle streaming spatiotemporal data.

In this demonstration, we present a Real-time and Interactive SystEm (RISE) for STKDV, a fully web-based platform built upon the fast and accurate compression technique COMP. RISE supports two capabilities: (1) interactive exploratory analysis for static datasets with responsive parameter adjustment and heatmap updates, and (2) real-time dashboard visualization for streaming data with continuous hotspot updates. Users can interactively analyze datasets containing up to 5,286,309 points without programming skills and experience both static exploration and streaming visualization. A demo video is available at https://github.com/YovelaZ/RISE.git.

CCS Concepts

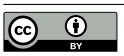
• **Information systems** → **Geographic information systems**.

Keywords

Real-time STKDV, Streaming and Static Data, Interactive System

*Corresponding author.

This work was supported by the Natural Science Foundation of China under grants 23IAA00610, 62572326, and 62372308, Scientific Foundation for Youth Scholars of Shenzhen University, the Science and Technology Development Fund Macau SAR (0003/2023/RIC, 0052/2023/RIA1, 0011/2025/RIC, 001/2024/SKL for SKL-IOTSC), and the Key Technological Innovation Program of Ningbo City under Grant No. 2024Z297. This work was performed in part at SICC which is supported by SKL-IOTSC, University of Macau.



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGSPATIAL '26, Riverside, CA, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2950-8/2026/11
https://doi.org/10.1145/3841645.3843105

ACM Reference Format:
Yue Zhong, Tsz Nam Chan, Leong Hou U, Dingming Wu, Wei Tu, Ruisheng Wang, and Joshua Zhexue Huang. 2026. RISE: A Real-time and Interactive Spatiotemporal Kernel Density Visualization System with Streaming and Static Data. In *The 34th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '26), November 03–06, 2026, Riverside, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3841645.3843105

1 Introduction

The proliferation of IoT devices, mobile services, and social media has shifted spatiotemporal data from static repositories to continuous, high-velocity streams, driving the need for real-time analytical systems. Spatiotemporal Kernel Density Visualization (STKDV) [5, 7–9] addresses this need by generating intuitive hotspot maps that reveal regions of high event concentration and their temporal evolution, with proven applications in urban mobility analysis [17], disease outbreak monitoring [15], traffic accident analysis [16], and crime hotspot detection [14]. Figure 1 presents an STKDV dashboard for visualizing shared bike trip origin hotspots in Downtown Los Angeles on February 6, 2024. The dashboard combines two historical hotspot snapshots (07:09 and 10:39) with a real-time heatmap (13:54), allowing users to observe how spatial activity patterns evolve throughout the day. As shown in the figure, hotspot regions exhibit noticeable shifts over time, indicating that travel demand can change substantially even within a single day. Such changes highlight the importance of real-time hotspot monitoring, as delayed visualizations may lead to outdated decisions.

Despite its analytical advantages, practical systems that support STKDV remain scarce and technically immature. The main challenge lies in the high computational cost of spatiotemporal kernel density estimation, which makes real-time or interactive exploration difficult. Existing desktop GIS platforms, such as QGIS with the **Fast Density Analysis** plugin [10] and ArcGIS Pro with the **Space Time Kernel Density** tool [12], allow users to generate STKDV and identify evolving hotspots over time. In addition, programming libraries such as **LIBKDV** [6] in Python, and **sparr** [11] and **kernstadapt** [13] in R, provide programmatic support for STKDV computation. However, these tools often fail to achieve

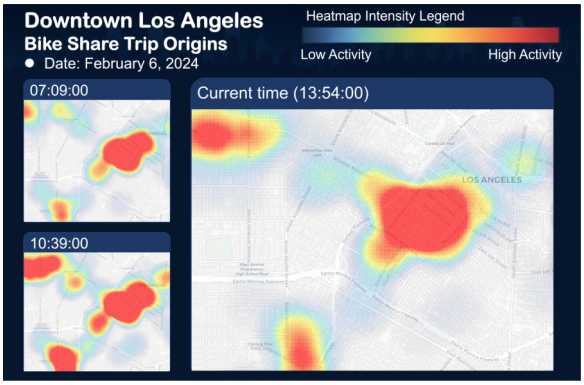


Figure 1: Real-time STKDV dashboard for shared bike trips in Downtown Los Angeles.

real-time heatmap responsiveness, let alone support interactive exploration and real-time streaming data visualization. In addition, Python and R packages require programming skills, limiting accessibility for non-programmers and domain experts without technical backgrounds.

To overcome these limitations, we present a Real-time and Interactive SystEm (**RISE**) for STKDV, a fully web-based platform for efficient hotspot exploration and visual analytics. To address the computational bottleneck, RISE integrates a state-of-the-art block compression technique (COMP [18]), which accelerates STKDV computation by reducing dataset size prior to density estimation while preserving visual fidelity. This acceleration enables real-time responsiveness, allowing users to interactively adjust parameters and instantly observe updated heatmaps. Building on this efficient computational foundation, we further extend the processing pipeline to support streaming spatiotemporal data, enabling continuous data ingestion and dynamic hotspot updates. As a result, RISE supports two complementary scenarios: (1) interactive exploration of static spatiotemporal datasets and (2) real-time dashboard visualization for streaming spatiotemporal data. RISE is among the first web-based systems to integrate both capabilities for STKDV. Table 1 compares existing STKDV systems in terms of computational speed, required user expertise, interactive exploration, and streaming support.

Table 1: Comparison of software tools for STKDV.

Software Tool	Speed	Prior Skill	Interactive	Streaming
Fast Density Analysis [10]	Fast	No	No	No
Space Time Kernel Density [12]	Slow	No	No	No
LIBKDV [6]	Medium	Python	No	No
sparr [11]	Slow	R	No	No
kernstadapt [13]	Slow	R	No	No
RISE (our system)	Real-time	No	Yes	Yes

In this paper, we discuss the technical overview of our system in Section 2 and illustrate RISE in Section 3. Section 4 presents the demonstration plan.

2 Technical Overview

In this section, we first provide the formal problem definition (ϵ-STKDV) in Section 2.1. Then, we introduce block compression in Section 2.2. The extension to support streaming spatiotemporal data is presented in Section 2.3. Finally, we describe the extension to additional kernel functions in Section 2.4.

2.1 Problem Definition

Since evaluating the exact STKDV is time-consuming, we evaluate the approximate version of STKDV (ϵ-STKDV), which is formulated as follows.

Problem 1. *(ϵ-STKDV [18]) Given the visualized region with spatial resolution $X \times Y$, T timestamps $(t_1, t_2, \ldots, t_T)$, a set P of spatiotemporal data points, and an absolute error ϵ, construct an approximate spatiotemporal dataset $\mathcal{S}$ and compute the approximate spatiotemporal kernel density value $A_{\mathcal{S}}(\mathbf{q}, t_i)$ for each pixel-timestamp pair $(\mathbf{q}, t_i)$ such that Equation 1 holds.*

$$|A_{\mathcal{S}}(\mathbf{q}, t_i) - \mathcal{F}_P(\mathbf{q}, t_i)| \leq \epsilon \tag{1}$$

where $\mathcal{F}_P(\mathbf{q}, t_i)$ is the spatiotemporal kernel density estimation value of the pixel-timestamp pair $(\mathbf{q}, t_i)$, defined in Equation 2.

$$\mathcal{F}_P(\mathbf{q}, t_i) = \frac{1}{|P|} \sum_{(\mathbf{p}, t_\mathbf{p}) \in P} K_{space}(\mathbf{q}, \mathbf{p}) \cdot K_{time}(t_i, t_\mathbf{p}) \tag{2}$$

where K_{space} and K_{time} denote the spatial kernel and temporal kernel, respectively. Table 2 lists some commonly used kernel functions.

Table 2: Kernel functions supported in RISE. Here, $u = \|\mathbf{q}-\mathbf{p}\|_2$ or $|t_i - t_\mathbf{p}|$, and b denotes the bandwidth parameter (b_σ for spatial kernels or b_τ for temporal kernels).

Kernel	Function $K(u)$
Triangular	$K(u) = max\left(0, 1 - \frac{u}{b}\right)$
Epanechnikov	$K(u) = max\left(0, \frac{3}{4}\left(1 - \frac{u^2}{b^2}\right)\right)$
Quartic (Biweight)	$K(u) = max\left(0, \frac{15}{16}\left(1 - \frac{u^2}{b^2}\right)^2\right)$
Triweight	$K(u) = max\left(0, \frac{35}{32}\left(1 - \frac{u^2}{b^2}\right)^3\right)$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}\, b} e^{-u^2/(2b^2)}$

2.2 Block Compression

To efficiently compute the approximate values $A_{\mathcal{S}}(\mathbf{q}, t_i)$ in Problem 1, RISE adopts COMP [18], a block compression technique with deterministic absolute error guarantees. By compressing the original spatiotemporal dataset into representative blocks, COMP substantially reduces the number of kernel evaluations during STKDV computation. Here, we first introduce the core idea of COMP.

COMP partitions the spatiotemporal domain into uniform blocks of size $\omega_\mathcal{B} \times \omega_\mathcal{B} \times \lambda_\mathcal{B}$, as illustrated in Figure 2. Within each block $\mathcal{B}$, all points are compressed into a single representative located at the block center $(\mathbf{p}_\mathcal{B}, t_{\mathbf{p}_\mathcal{B}})$, together with a count $\mathbb{C}(\mathcal{B})$ indicating the number of original data points it contains.

Based on this compressed representation, the approximate STKDV value $A_{\mathcal{S}}(\mathbf{q}, t_i)$ is computed using block centers as in Equation 3.

$$A_{\mathcal{S}}(\mathbf{q}, t_i) = \frac{1}{|P|} \sum_{\mathcal{B} \in \mathcal{S}} \mathbb{C}(\mathcal{B}) \cdot K_{space}(\mathbf{q}, \mathbf{p}_\mathcal{B}) \cdot K_{time}(t_i, t_{\mathbf{p}_\mathcal{B}}) \tag{3}$$

where $\mathcal{S}$ denotes the set of non-empty blocks returned by COMP.

To enable controllable approximation quality, COMP [18] determines the block size via a user-specified error parameter ϵ, providing deterministic error guarantees and deriving block size formulas

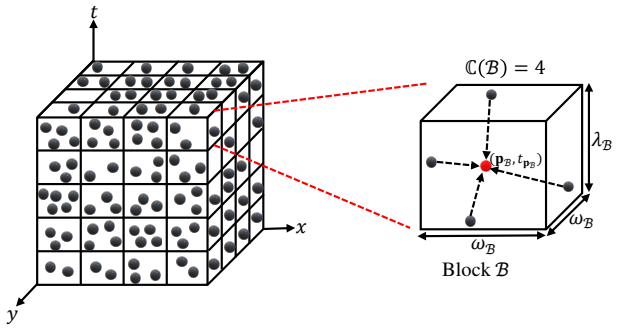


Figure 2: Illustration of block compression (obtained from [18]).

for the Triangular, Epanechnikov, and Quartic kernels (Theorem 1 of [18]), which are extendable to additional kernels (Section 2.4).

2.3 Extension to Streaming Data

Although effective for static datasets, directly applying COMP to streaming spatiotemporal data remains challenging. First, storing all incoming data leads to prohibitive memory overhead. To address this, a sliding temporal window is typically adopted, where data continuously enter and expire from the active range. However, this introduces a new challenge: if COMP is re-executed after each window update, the overlapping portion of data across consecutive windows is repeatedly processed, leading to substantial redundant computation and significantly degrading the efficiency of real-time heatmap updates. As illustrated in Figure 3, consider two adjacent timestamps $t^{(1)}$ and $t^{(2)}$ with highly overlapping time windows. Although most data points are shared between consecutive windows, COMP is recomputed from scratch at each timestamp. Consequently, the shared points (highlighted in color in Figure 3) are redundantly re-partitioned into different blocks at $t^{(2)}$, despite having already been processed at $t^{(1)}$. As such, efficiently reusing computations under window transitions is essential for supporting real-time streaming scenarios.

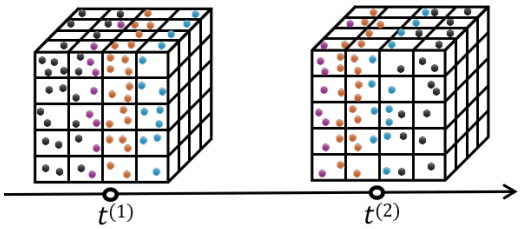


Figure 3: Illustration of redundant computation under sliding window updates.

To address these challenges, we leverage a key observation from the kernel functions listed in Table 2: all supported kernels have finite support (for Triangular, Epanechnikov, Quartic, and Triweight) or decay to near-zero outside a certain range (for Gaussian). In practice, for a given current timestamp t^*, only data points falling within the temporal bandwidth b_τ prior to t^* have non-negligible contributions to the STKDV value at t^*. Therefore, for real-time streaming visualization, we set the temporal window to $[t^* - b_\tau, t^*]$ and only maintain blocks whose representative timestamps remain within this temporal range. Using Figure 4 as an example, the temporal window advances from $[t^{(1)} - b_\tau, t^{(1)}]$ to $[t^{(2)} - b_\tau, t^{(2)}]$ as time evolves from $t^{(1)}$ to $t^{(2)}$. At timestamp $t^{(1)}$, the active compressed set contains two blocks $\mathcal{B}_1$ and $\mathcal{B}_2$ with $\mathbb{C}(\mathcal{B}_1) = 3$ and $\mathbb{C}(\mathcal{B}_2) = 2$. As the temporal window moves forward, blocks whose center timestamps $t_{p_{\mathcal{B}}}$ that are smaller than the updated lower bound $t^{(2)} - b_\tau$ are regarded as expired and removed entirely from the active compressed set (e.g., block $\mathcal{B}_1$ in Figure 4). Meanwhile, as new data points arrive, they are directly mapped to their corresponding blocks without being stored individually. Points falling into existing blocks increment the corresponding block counts (illustrated by the orange spheres in Figure 4, where $\mathbb{C}(\mathcal{B}_2)$ increases from 2 to 4), while points falling into previously empty regions trigger the creation of new blocks (illustrated by the purple spheres in Figure 4, where a new block $\mathcal{B}_3$ with $\mathbb{C}(\mathcal{B}_3) = 2$ is created). Consequently, the compressed representation is updated incrementally at timestamp $t^{(2)}$ without recomputing the entire compressed representation from scratch, enabling responsive real-time hotspot visualization under continuous data streams.

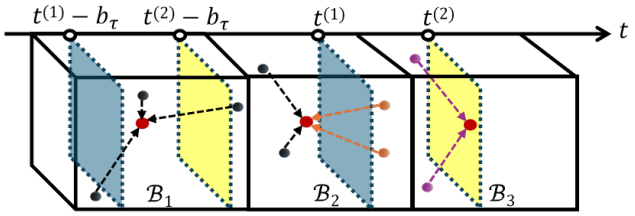


Figure 4: Illustration of streaming block updates under a sliding temporal window.

2.4 Extension to Additional Kernel Functions

To improve the generality of RISE, we further extend COMP to support additional commonly used kernel functions, including Triweight and Gaussian kernels. For each kernel, the corresponding block size $\omega_{\mathcal{B}} \times \omega_{\mathcal{B}} \times \lambda_{\mathcal{B}}$ is determined according to the target error parameter ϵ while preserving deterministic absolute error guarantees (Problem 1). The resulting block size settings are summarized in Theorem 2.1.

Theorem 2.1. *Given a location dataset P and an absolute error ϵ, we can achieve the absolute error guarantee ϵ in Problem 1 if the block size $\omega_{\mathcal{B}} \times \omega_{\mathcal{B}} \times \lambda_{\mathcal{B}}$ of each block $\mathcal{B}$ in $\mathcal{S}$ has the following settings.*

(1) $\omega_{\mathcal{B}} = \frac{16\sqrt{10}\cdot\epsilon\cdot b_\sigma}{147}$ *and* $\lambda_{\mathcal{B}} = \frac{32\sqrt{5}\cdot\epsilon\cdot b_\tau}{147}$ *using the triweight kernels.*

(2) $\omega_{\mathcal{B}} = \sqrt{2e}\pi\cdot\epsilon\cdot b_\sigma$ *and* $\lambda_{\mathcal{B}} = 2\sqrt{e}\pi\cdot\epsilon\cdot b_\tau$ *using the Gaussian kernels.*

3 Illustration of RISE

RISE is a real-time web-based system for STKDV that supports both continuous streaming visualization and interactive analysis of static datasets.

3.1 Real-Time Streaming Visualization

To support online spatiotemporal monitoring, RISE provides a real-time dashboard that continuously ingests streaming data and updates hotspot visualizations without restarting the computation process. As new data arrive, the system incrementally updates block statistics (e.g., removing expired blocks and adding new blocks) within the sliding temporal window introduced in Section 2.3.

Figure 5 illustrates the streaming interface of RISE. The heatmap is refreshed continuously as incoming events are processed, enabling users to observe evolving hotspot patterns in real time. Users can monitor how dense regions emerge, shift, and disappear over



Figure 5: Real-time streaming dashboard of RISE for continuous hotspot monitoring.

time. This design allows RISE to maintain interactive responsiveness even under high-velocity data streams.

To evaluate real-time performance, we measure the frame rate (FPS) of RISE under continuous streaming updates and compare it with Kepler.gl [3], a widely used geospatial visualization system supporting temporal animation. As shown in Table 3, RISE achieves higher FPS across all datasets under identical settings.

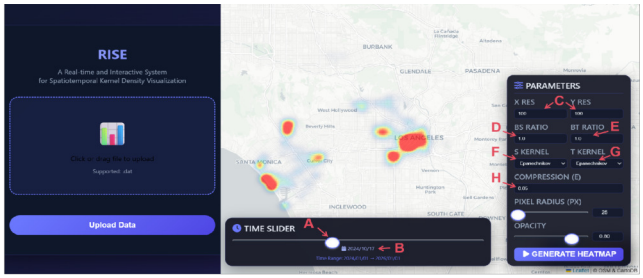
Table 3: Comparison of FPS between Kepler.gl and RISE.

Dataset	Kepler.gl	RISE
India Infectious Disease [2]	11	85
Los Angeles Bike Share Trips [4]	6	68
Chicago Taxi Trips [1]	1	73

3.2 Interactive Exploration of Static Data

Besides streaming visualization, RISE also supports interactive exploration and analysis of static datasets.

Users first upload a dataset through the data upload interface (as shown in Figure 6a). The system caches the data on the server, computes STKDV results with default parameters, and redirects users to the main visualization interface (as shown in Figure 6b), where the heatmap is displayed.



(a) Data upload interface (b) Main visualization interface

Figure 6: Main interface of RISE for interactive spatiotemporal hotspot exploration.

In the main visualization interface, users can interactively explore spatiotemporal hotspot patterns through temporal and spatial navigation. As shown in Figure 6b, temporal exploration is supported via a time slider (A) for continuous navigation or direct timestamp selection (B), while spatial exploration includes zooming and panning to examine hotspot patterns at different scales. The parameter control panel on the right side of the interface allows users to adjust visualization settings, including resolution (C), spatial and temporal bandwidth ratios (D and E), spatial and temporal kernel functions (F and G), and compression error tolerance (H). The bandwidth ratio controls the bandwidth computed by Scott's rule. After each interaction, the heatmap is updated immediately, enabling efficient comparison across different parameter settings and supporting responsive exploratory analysis.

4 Demonstration Plan

In this demonstration, we adopt three spatiotemporal datasets for testing, as summarized in Table 3. Here, we consider two demonstration scenarios.

Investigation of real-time hotspot evolution under streaming data. To demonstrate real-time monitoring in RISE, we simulate streaming input by feeding historical real-world datasets into the system according to their timestamps, with accelerated playback to enable rapid observation of hotspot evolution. We continuously update the heatmap visualization, enabling the emergence, movement, and evolution of hotspot regions over time.

Investigation of interactive parameter-driven hotspot analysis. Since hotspot patterns are sensitive to visualization and analysis settings, users explore different configurations to understand their impact. Therefore, we adjust parameters (b_σ, b_τ, and ϵ), perform spatial zooming and panning, navigate across time, and switch kernel functions. We then update hotspot visualizations after each interaction to compare results under different settings and understand how parameter choices affect hotspot distributions.

References

[1] Chicago taxi trips data. https://catalog.data.gov/dataset/taxi-trips-2024.
[2] Infectious disease cases in india. https://github.com/beoutbreakprepared/nCoV2019/tree/master/latest_data.
[3] Kepler.gl: Open source geospatial analysis tool. https://kepler.gl.
[4] Los angeles shared bike trips data. https://bikeshare.metro.net/about/data/.
[5] T. N. Chan, P. L. Ip, L. H. U, B. Choi, and J. Xu. SWS: A complexity-optimized solution for spatial-temporal kernel density visualization. *Proc. VLDB Endow.*, 15(4):814–827, 2022.
[6] T. N. Chan, P. L. Ip, L. H. U, K. Zhao, B. Choi, and J. Xu. LIBKDV: A versatile kernel density visualization library for geospatial analytics. In *VLDB*, volume 15, pages 3606–3609, 2022.
[7] T. N. Chan, P. L. Ip, B. Zhu, L. H. U, D. Wu, J. Xu, and C. S. Jensen. Large-scale spatiotemporal kernel density visualization. In *ICDE*, pages 99–113, 2025.
[8] T. N. Chan, L. H. U, B. Choi, J. Xu, and R. Cheng. Kernel density visualization for big geospatial data: Algorithms and applications. In *MDM*, pages 231–234, 2023.
[9] T. N. Chan, L. H. U, B. Choi, J. Xu, and R. Cheng. Large-scale geospatial analytics: Problems, challenges, and opportunities. In *SIGMOD Companion*, pages 21–29, 2023.
[10] T. N. Chan, B. Zhu, L. H. U, D. Wu, W. Tu, and J. Xu. A fast, versatile, and user-friendly plugin for kernel density analysis. In *ICDE*, pages 4160–4163, 2026.
[11] T. M. Davies and J. C. Marshall. sparr: Spatial and spatiotemporal relative risk, 2025. R package.
[12] Esri. Space time kernel density (spatial analyst) — ArcGIS Pro. https://pro.arcgis.com/en/pro-app/3.5/tool-reference/spatial-analyst/space-time-kernel-density.htm, 2026.
[13] J. A. González and P. Moraga. kernstadapt: Adaptive kernel estimators for point process intensities on linear networks, 2024. R package.
[14] Y. Hu, F. Wang, C. Guin, and H. Zhu. A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation. *Applied Geography*, 99:89–97, 2018.
[15] Z. Kan, M.-P. Kwan, J. Huang, J. Cai, and D. Liu. A spatial network-based assessment of individual exposure to covid-19. *Annals of the American Association of Geographers*, 114(8):1693–1703, 2024.
[16] Y. Li, M. Abdel-Aty, J. Yuan, Z. Cheng, and J. Lu. Analyzing traffic violation behavior at urban intersections: A spatio-temporal kernel density estimation approach using automated enforcement system data. *Accident Analysis & Prevention*, 141:105509, 2020.
[17] B. Shen, X. Xu, J. Li, A. Plaza, and Q. Huang. Unfolding spatial-temporal patterns of taxi trip based on an improved network kernel density estimation. *ISPRS International Journal of Geo-Information*, 9(11):683, 2020.
[18] Y. Zhong, T. N. Chan, L. H. U, D. Wu, W. Tu, R. Wang, and J. Z. Huang. A fast and accurate block compression solution for spatiotemporal kernel density visualization. In *SIGKDD*, pages 4098–4109, 2025.