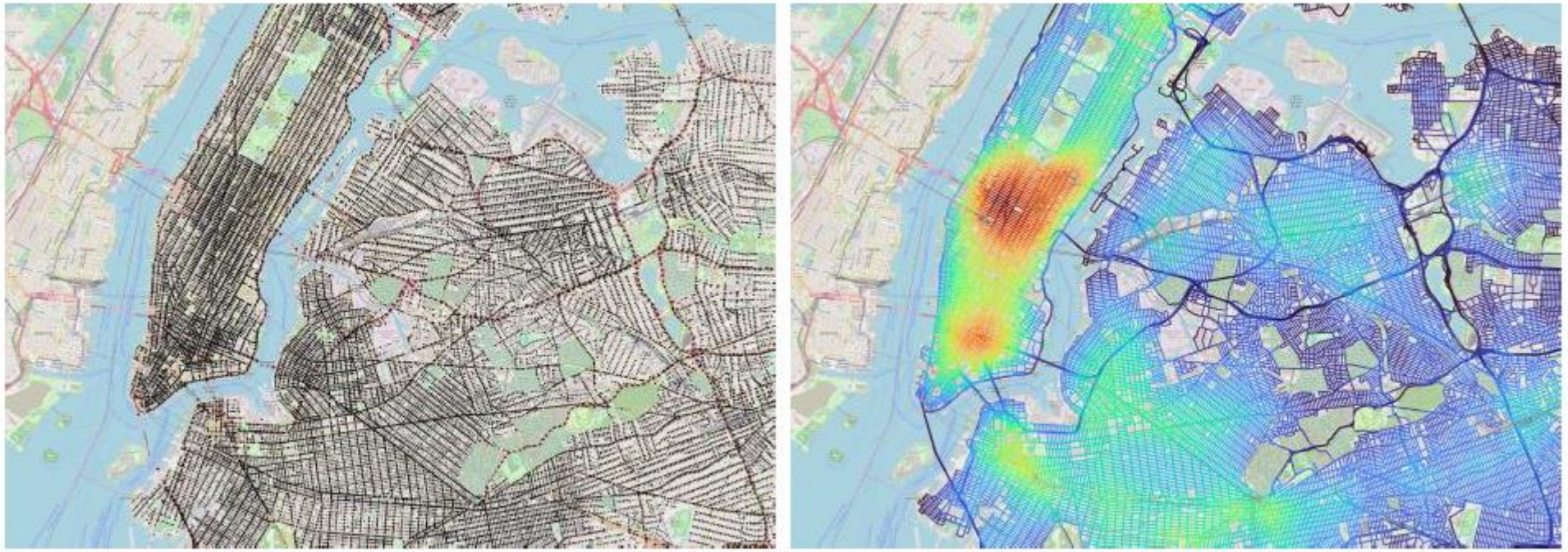
PLAN: Fast and Approximate Gaussian Kernel Density Visualization in Road Networks

(Edison) Tsz Nam Chan[1], Hongwei Ye[1], Bojian Zhu[2], Leong Hou U[3],
Dingming Wu[1], Ruisheng Wang[1], Joshua Zhexue Huang[1]
[1]Shenzhen University [2]Hong Kong Baptist University [3]University of Macau

香港浸會大學
HONG KONG BAPTIST UNIVERSITY

Why Network Kernel Density Visualization (NKDV) ?



(a) New York traffic accident dataset (b) NKDV

- Some representative applications:
  - Traffic/Traffic accident hotspot detection
  - Crime hotspot detection
  - Urban planning

What is NKDV?

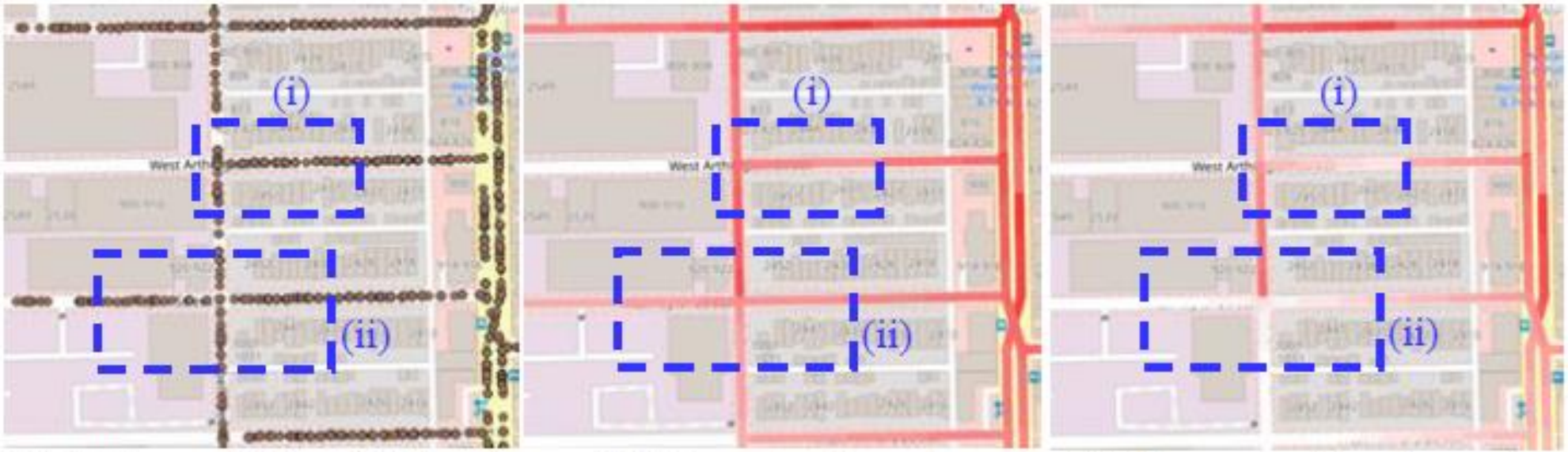
- Given a location dataset $P = \{p_1, p_2, \ldots, p_n\}$ and a road network $G = (V, E)$, we need to color each lixel q based on the network kernel density function $\mathcal{F}_P(q)$.

$$\mathcal{F}_P(q) = \frac{1}{|P|} \sum_{p \in P} K_G(q, p)$$

Kernel	Function
Epanechnikov	$\begin{cases} 1 - \frac{1}{b_G^2} d_G(q,p)^2 & \text{if } d_G(q,p) \leq b_G \\ 0 & \text{otherwise} \end{cases}$
Gaussian	$\exp\left(-\frac{1}{b_G^2} d_G(q,p)^2\right)$

(a) Location dataset (b) NKDV

NKDV: Epanechnikov v.s. Gaussian



(a) Chicago 311-call dataset (b) Gaussian kernel (c) Epanechnikov kernel

- Gaussian kernel can provide smoother visualization. ☺

- Gaussian kernel is the default one in many software packages. ☺

Weakness of Gaussian-based NKDV

- Gaussian-based KDV is computationally expensive. ☹
- Example (Chicago 311-call dataset):
  - Number of data points: 10.893 million
  - Number of lixels (with 10m): 0.67 million
  - Number of operations: > 7.298 trillion
- Not feasible to generate multiple NKDVs. ☹

Weakness of Gaussian-based NKDV

- Cannot directly extend existing fast algorithms for Gaussian-based NKDV. ☹

$$\begin{cases} 1 - \frac{1}{b_G^2} d_G(q,p)^2 & \text{if } d_G(q,p) \leq b_G \\ 0 & \text{otherwise} \end{cases} \quad \text{v.s.} \quad \exp\left(-\frac{1}{b_G^2} d_G(q,p)^2\right)$$

- Does not have short-tail property in Gaussian kernel.

- Cannot decompose the network kernel density function $\mathcal{F}_P(q)$.

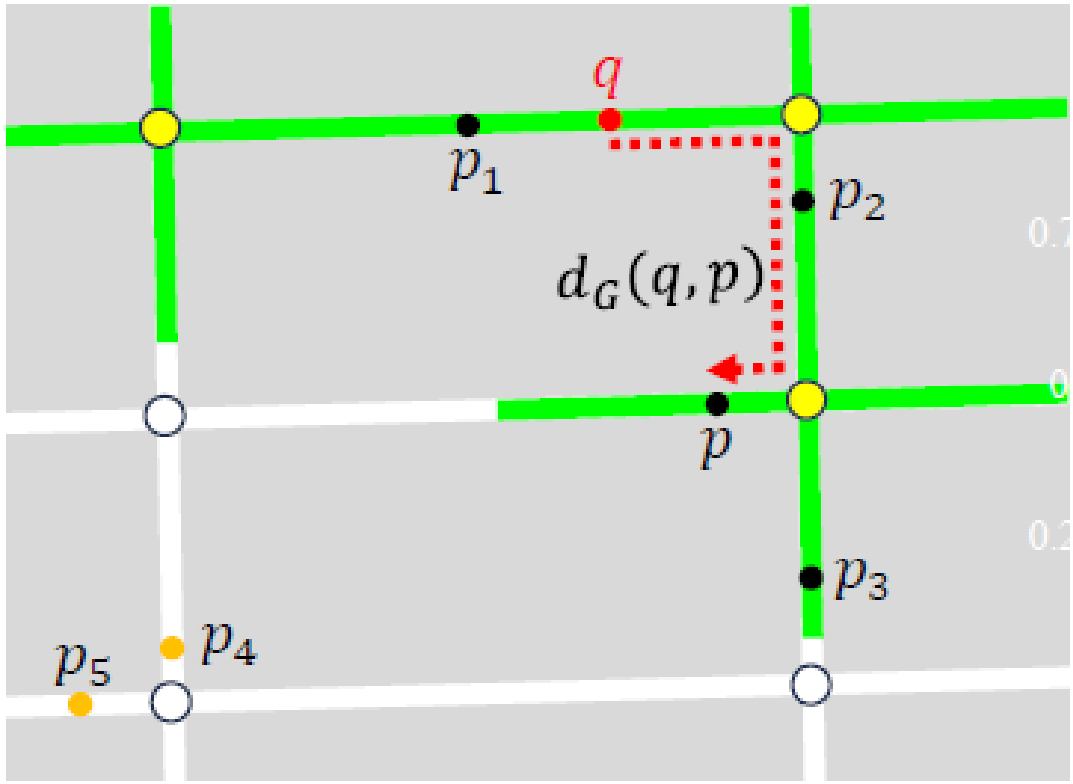
Our Contributions

Method	Time complexity	Space complexity	Accuracy guarantee
SOTA	$O(\|E\|T_{\text{SP}} + \|P\|L)$	$O(\|V\| + \|P\| + L + S_{\text{SP}})$	Exact
PLAN	$O\left(\|E\|T_{\text{SP}} + LM\|E\| \log\left(\frac{\|P\|}{\|E\|}\right)\right)$ (Theorem 1)	$O(\|V\| + \|P\| + L + S_{\text{SP}} + M)$ (Theorem 3)	ϵ-absolute error (Lemma 1)
PLAN+	$O\left(\|E\|T_{\text{SP}} + LM\|E\| \log\left(\frac{\|P\|}{M\|E\|}\right)\right)$ (Theorem 2)		

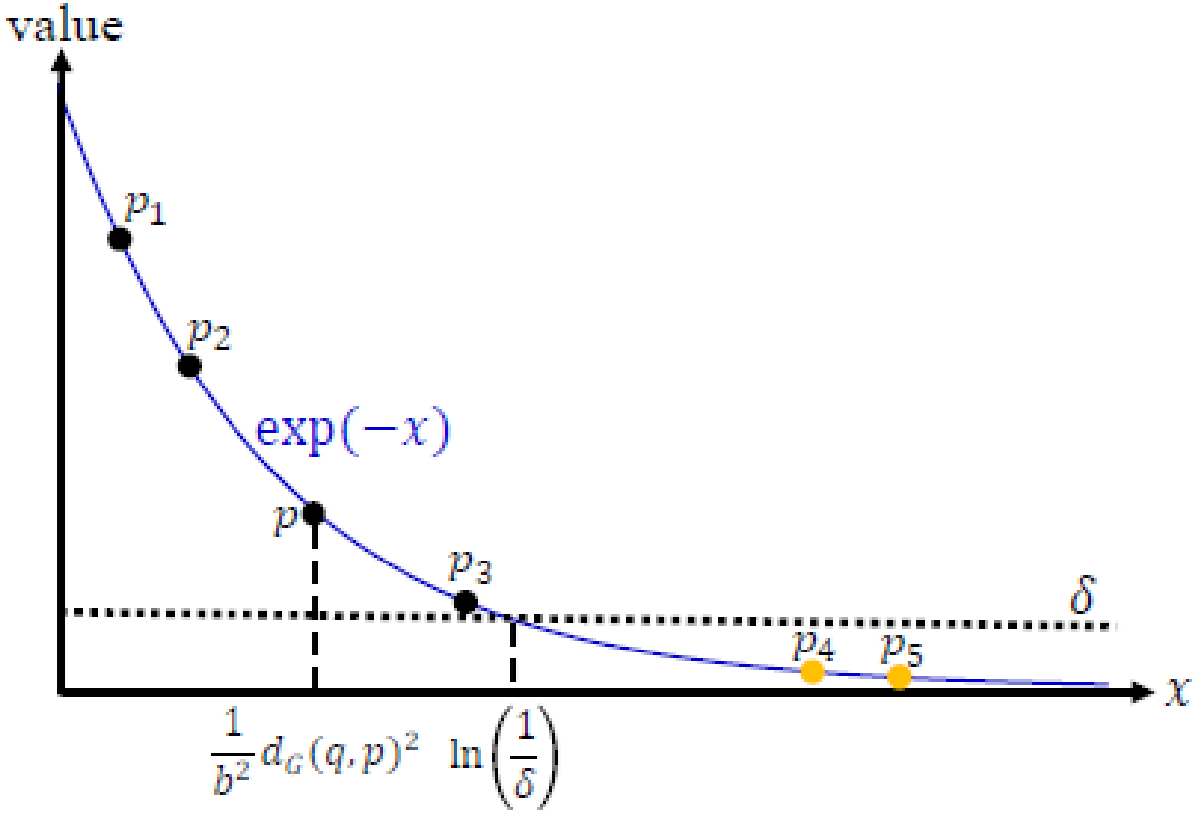
- Reduce the time complexity for generating NKDV with (1) a non-trivial approximation guarantee and (2) a slight overhead in terms of space complexity. ☺

- Achieve 32.47x to 2936.88x speedups compared with SOTA.

Core Idea 1: Enable the Short Tail Property



(a) Data points and a lixel q in a road network



(b) Exponential function

- Data points that are too far away from q can be discarded.

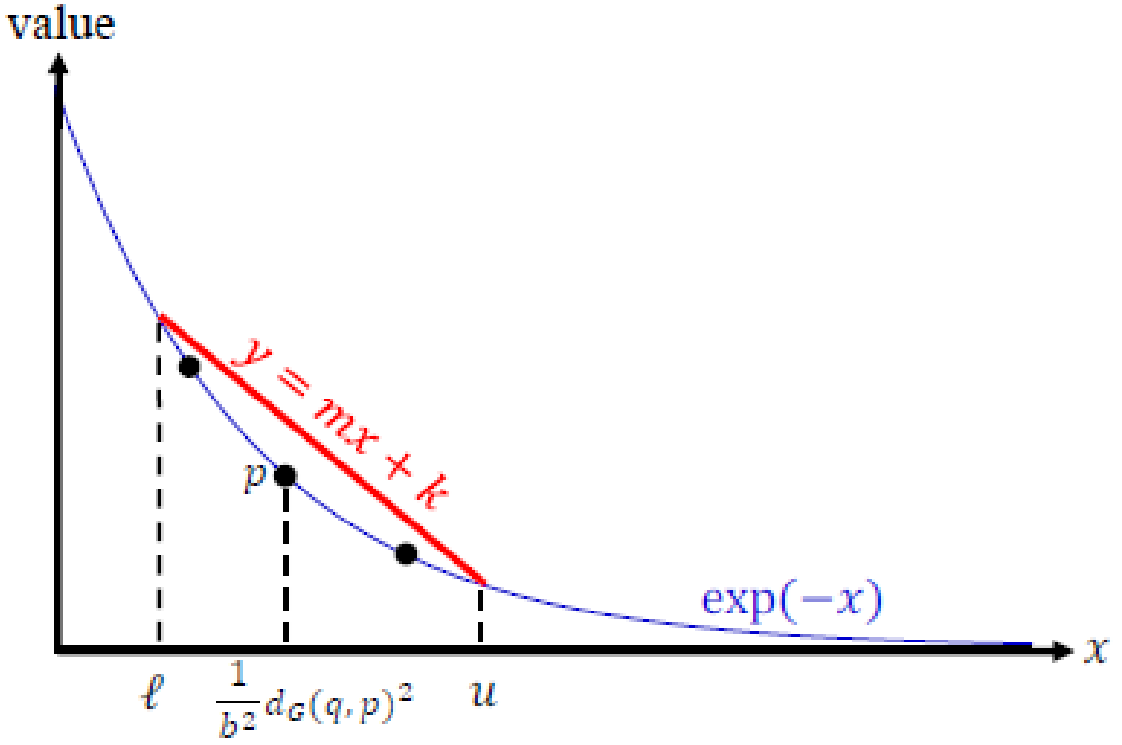
Core Idea 2: Function Approximation

- This upper bound function can approximate $\mathcal{F}_{\mathbb{P}}(q)$.

$$\mathcal{F}_{\mathbb{P}}(q) = \frac{1}{|\mathbb{P}|}\sum_{p\in\mathbb{P}} \exp\left(-\frac{1}{b^2} d_G(q,p)^2\right)$$

$$\leq \frac{1}{|\mathbb{P}|}\sum_{p\in\mathbb{P}} \left(\frac{m}{b^2} d_G(q,p)^2 + k\right)$$

where $\mathbb{P} = \left\{p \in P : \ell \leq \frac{1}{b^2} d_G(q,p)^2 \leq u\right\}$



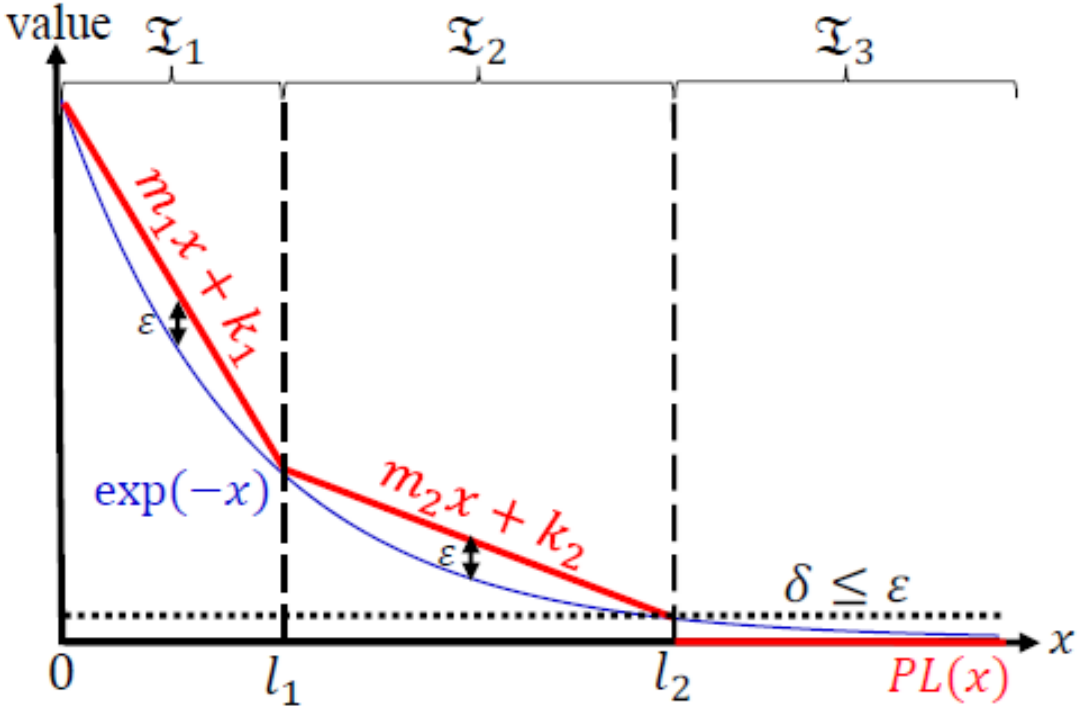
PLAN: Approximate the Network Kernel Density Function

- Use $A_P(q)$ to approximate $\mathcal{F}_P(q)$.

$$A_P(q) = \frac{1}{|P|}\sum_{p\in P} PL\left(\frac{1}{b^2}d_G(q,p)^2\right)$$

where

$$PL(x) = \begin{cases} m_1 x + k_1 & \text{if } x \in \mathcal{I}_1 \\ m_2 x + k_2 & \text{if } x \in \mathcal{I}_2 \\ \vdots & \vdots \\ m_{M-1} x + k_{M-1} & \text{if } x \in \mathcal{I}_{M-1} \\ 0 & \text{if } x \in \mathcal{I}_M \end{cases}$$



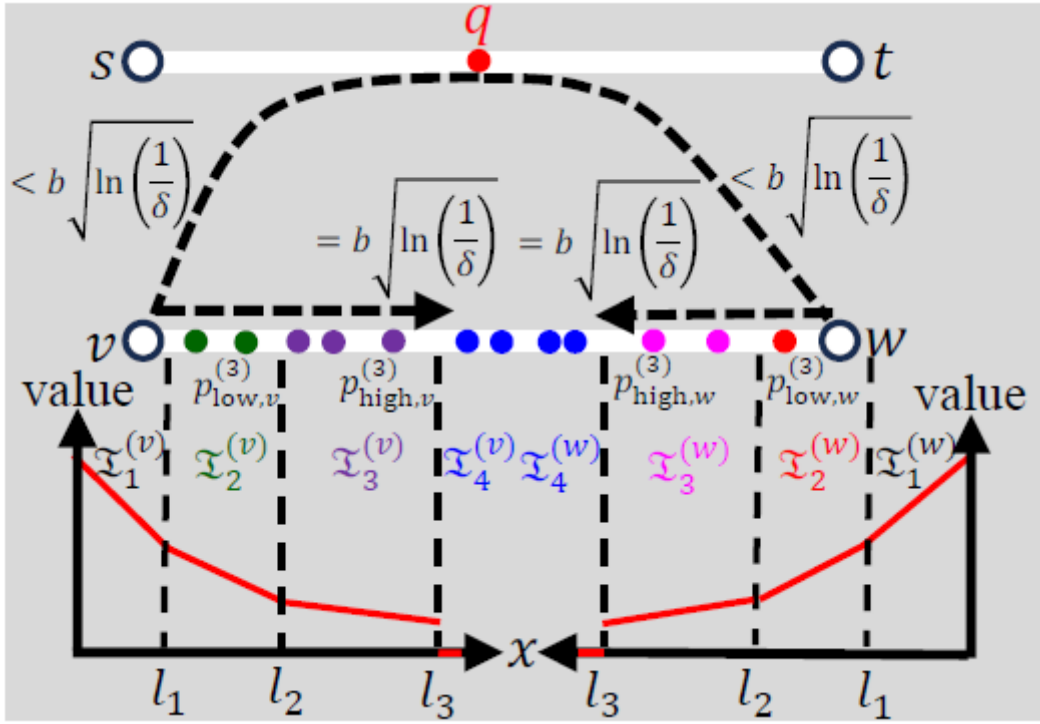
- Can achieve ε-absolute error guarantee. ☺

PLAN: Efficient Computation

- Similar to ADA [a].
  - Perform augmentation for each data point.
  - Adopt the binary search method.

- Main difference
  - Adopt the binary search method M times.

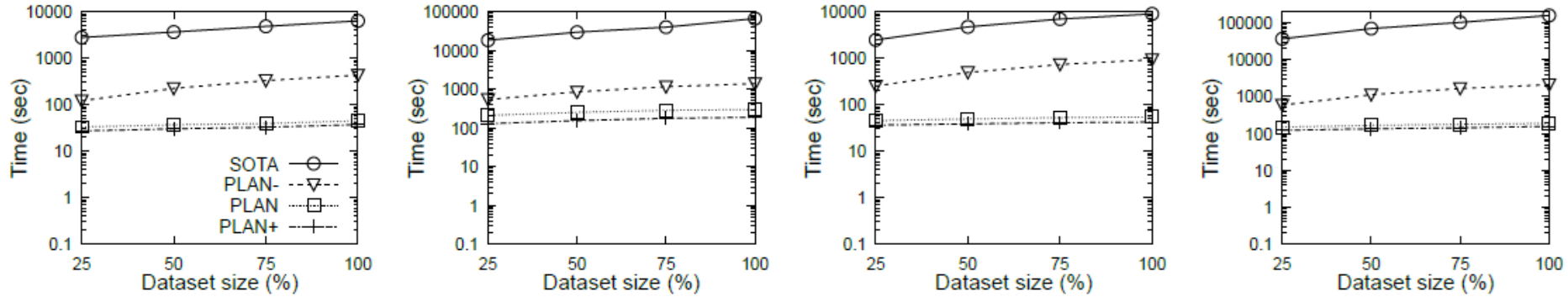
Refer to our paper for details.



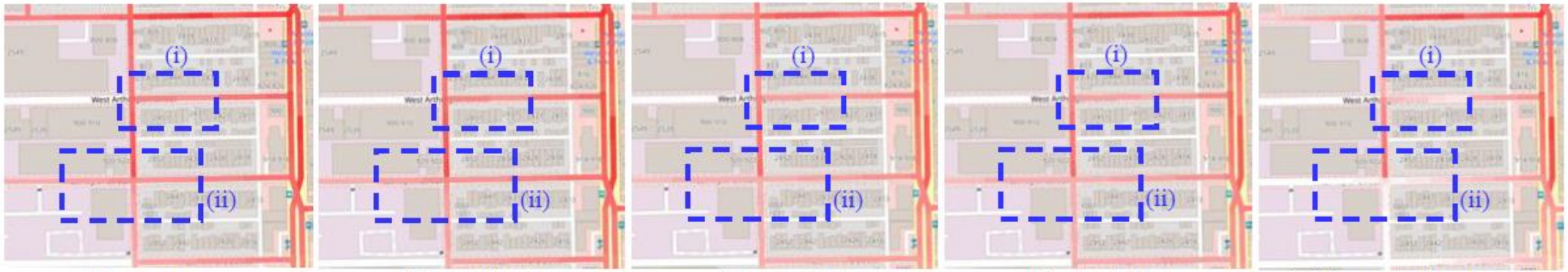
[a] T. N. Chan, Z. Li, L. H. U, J. Xu, and R. Cheng, "Fast augmentation algorithms for network kernel density visualization," *Proc. VLDB Endow.*, vol. 14, no. 9, pp. 1503–1516, 2021.

Experimental Evaluation

Dataset	$\|V\|$	$\|E\|$	$\|P\|$	Category
Minneapolis [36]	7,388	12,913	1,598,877	911 calls
New York [37]	71,019	120,623	1,897,418	Traffic accidents
San Francisco [34]	6,627	11,593	2,985,502	Crime events
Chicago [25]	41,111	70,067	10,893,129	311 calls



(a) Minneapolis (b) New York (c) San Francisco (d) Chicago

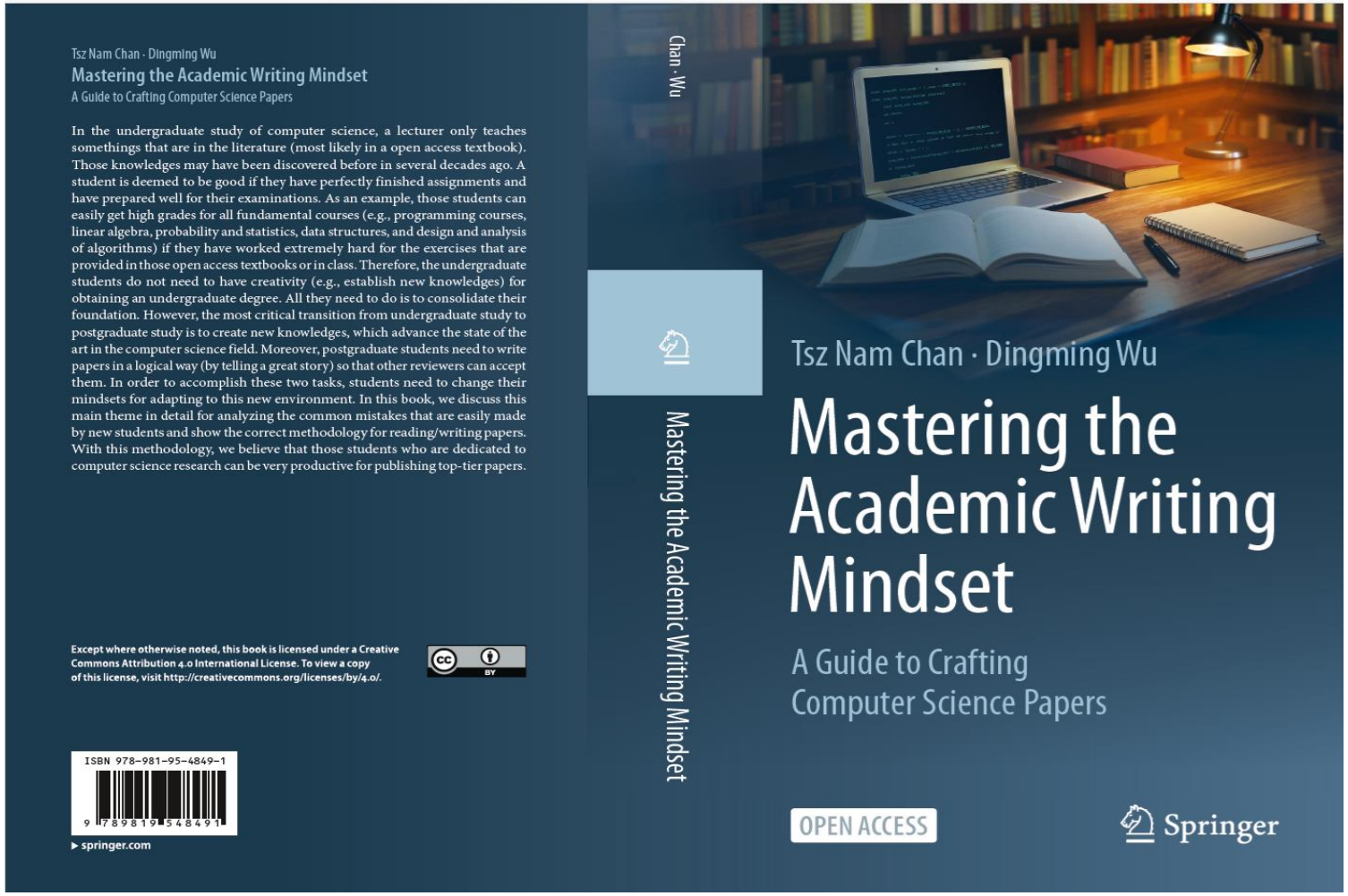


(a) EXACT (b) $\epsilon = 0.01$ (c) $\epsilon = 0.05$ (d) $\epsilon = 0.09$ (e) Epanechnikov kernel

Future Opportunities

- Incorporate PLAN into the QGIS plugin, called Fast Density Analysis.
- Support NKDV with adaptive bandwidth.
- Adopt parallel/distributed/hardware-based approaches to improve the efficiency of NKDV.
- Support other inefficient road network analysis tasks, e.g., network Moran's I, network K-function, and network DBSCAN.

Our New Book



https://link.springer.com/book/10.1007/978-981-95-4850-7