Graph Curvature for COVID-19 Network Risk Analytics

SciCAM MS Thesis Presentation

Qingyuan Cui

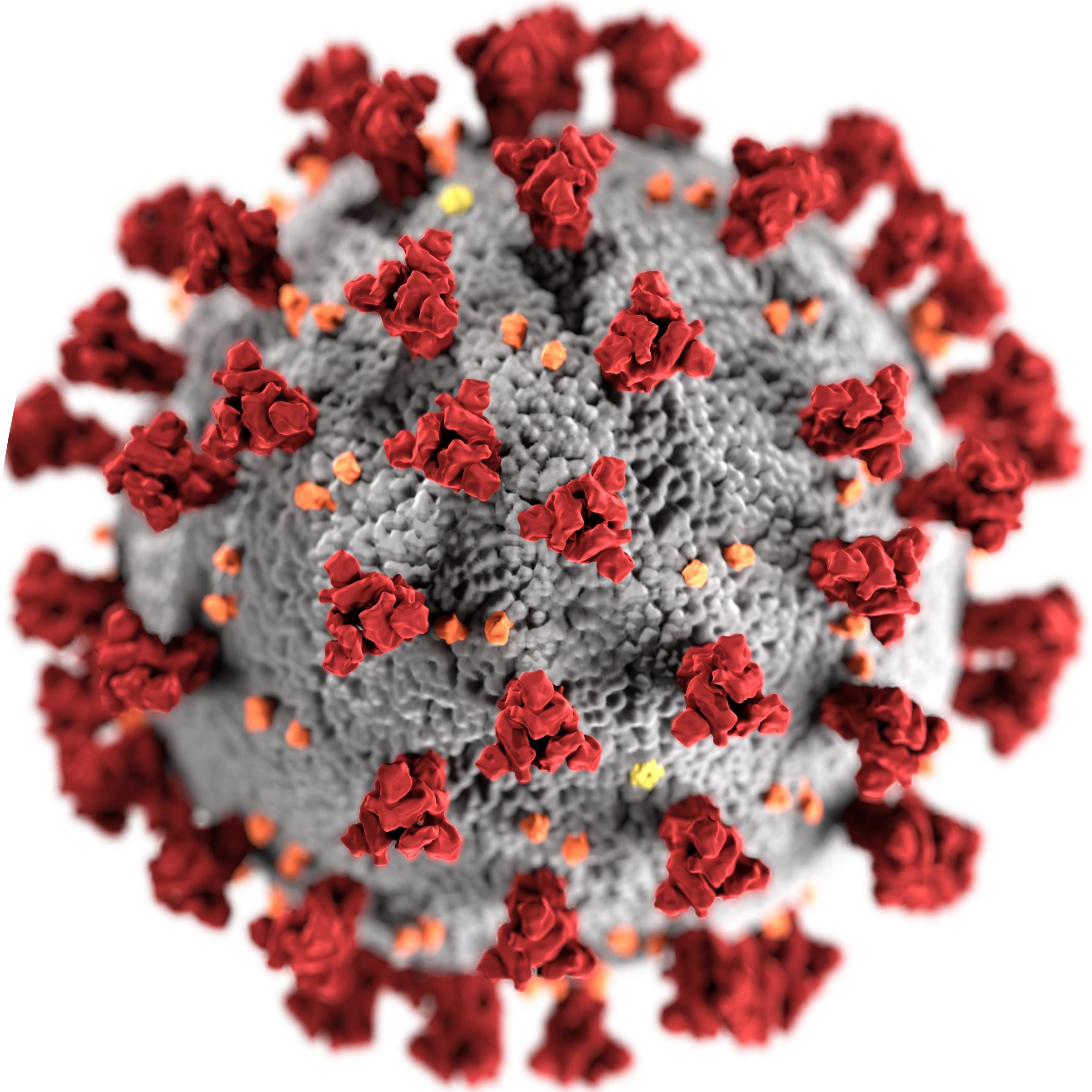
University of California, Santa Cruz

Aug. 11, 2021

COVID-19

203M cases

4.3M deaths



Worldwide total cases

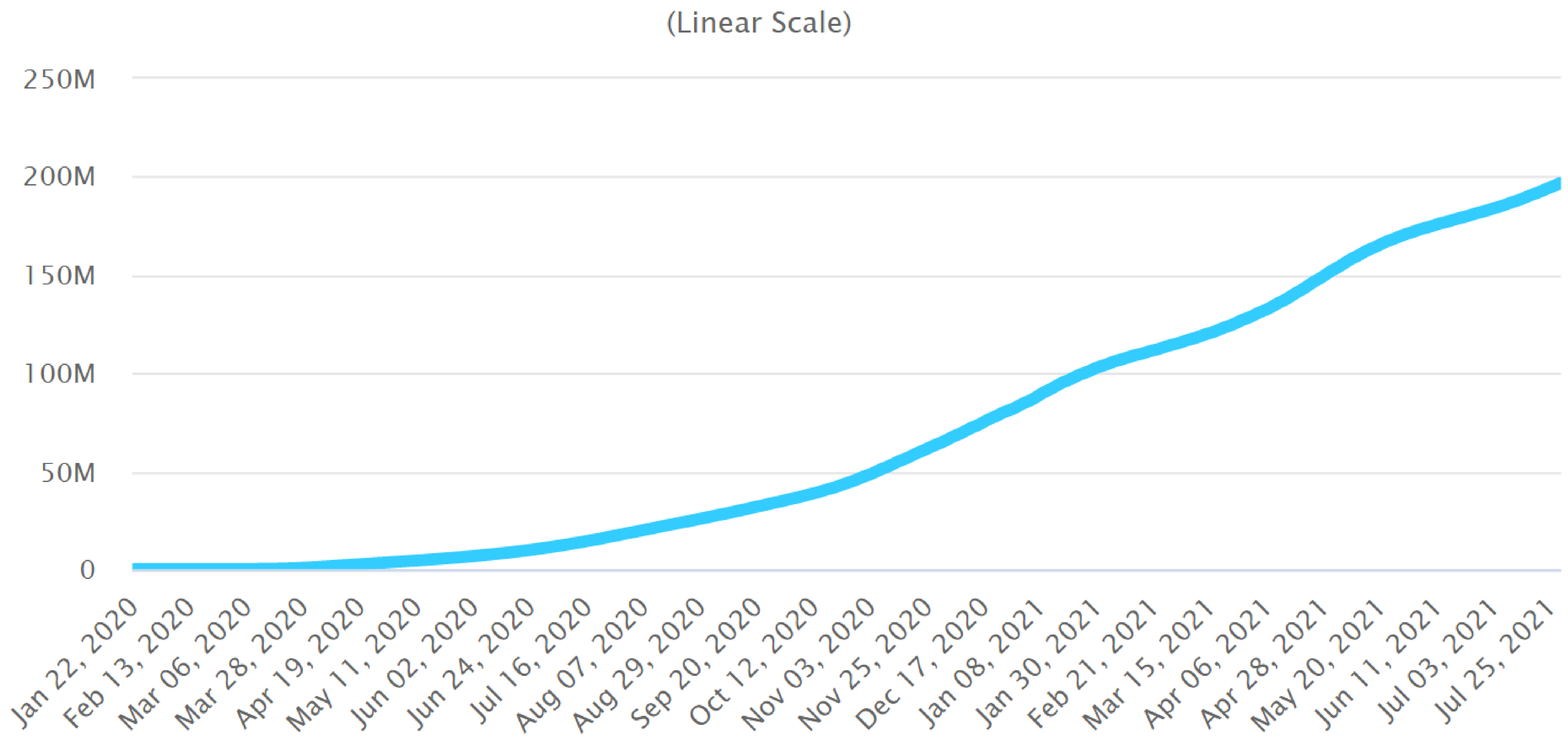


Image Credit: www.worldometers.info

California new reported cases

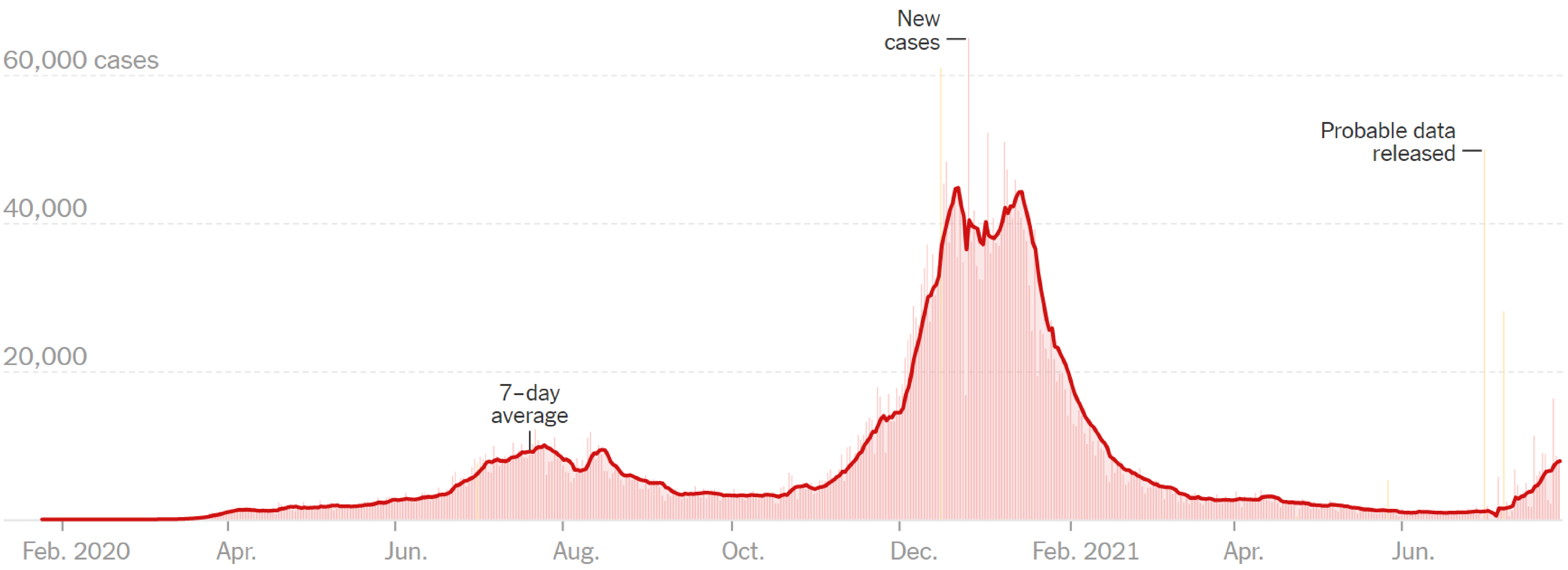


Image Credit: The New York Times

Challenges in COVID-19 risk analytics

Counties need to account time-varying traffic data

When to impose NPI measures? For how long?

Where are the infections spreading from? And spreading to?

This study

Proposes new risk analytics method from **county-level traffic graph**

Uses **graph curvature** to quantify the likelihood of spread

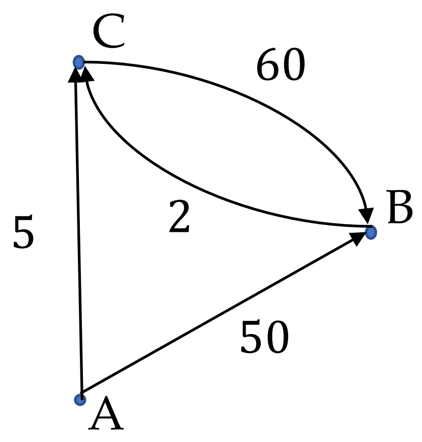
Demonstrates the proposed method on real county-level traffic data for **California during March 1st, 2020-March 31st, 2021**

County-level traffic graph

time-varying weighted directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ from inter-county traffic data

$w(\text{AB})$ = weekly average traffic count from county A to county B

From county	To county	w
A	B	50
A	C	5
B	A	0
B	C	2
C	A	0
C	B	60



Why graph curvature

County-level traffic graph	Graph Ricci and scalar curvatures
Vertex to vertex interaction	Neighborhood to neighborhood interaction
Pairwise information only when an edge exists between the vertices	Pairwise information over all possible pathways in the network

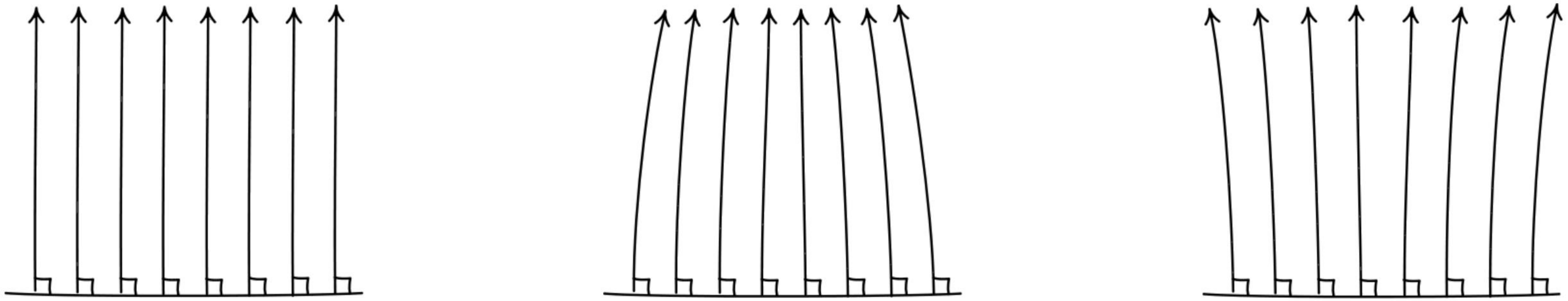
Ricci curvature κ on Riemannian manifold

Measures the deviation of the manifold from being locally flat (here, flat = Euclidean)

Quantifies that deviation in the tangent directions

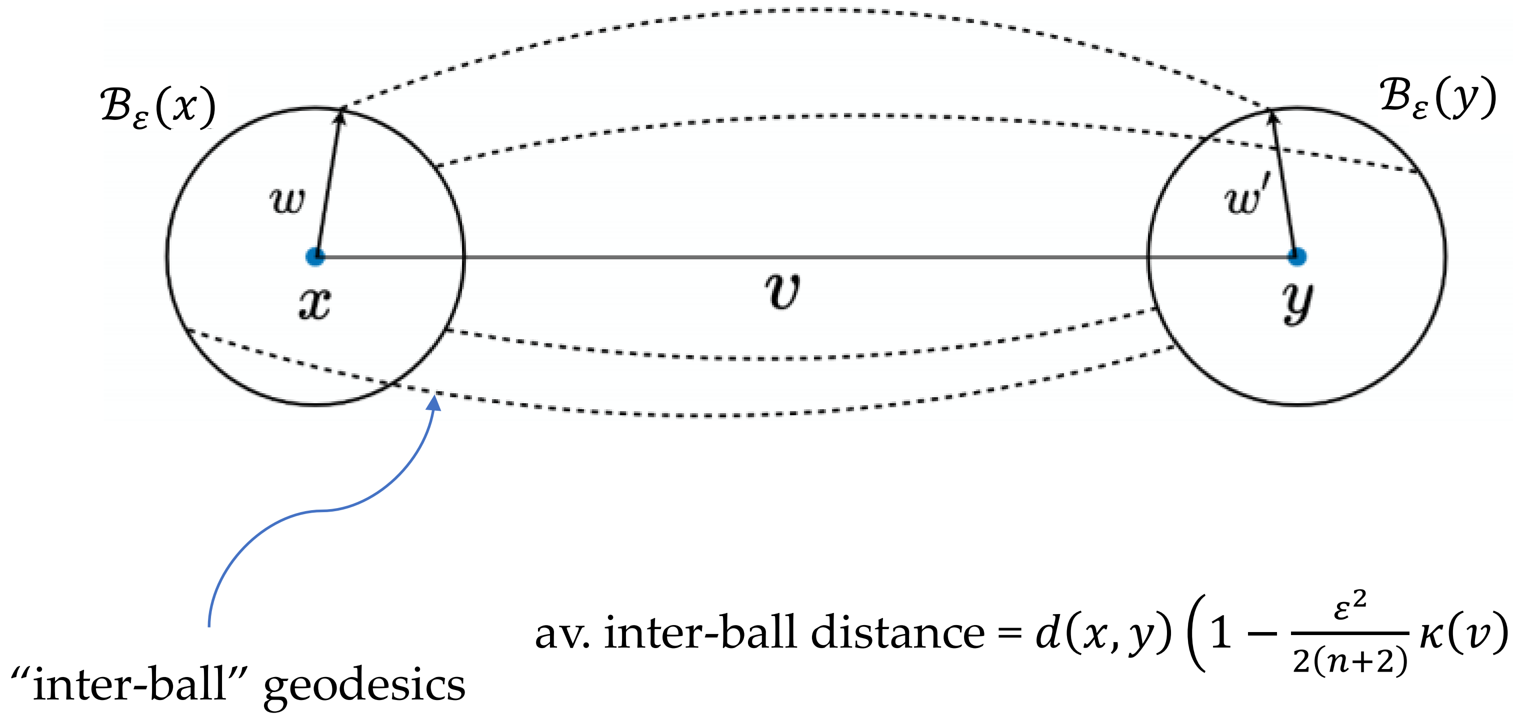
Controls the average dispersion of geodesics around those directions

Curvature and geodesic dispersion



Zero curvature   Positive curvature   Negative curvature

Ricci curvature κ on Riemannian manifold



av. inter-ball distance = $d(x,y)\left(1 - \frac{\varepsilon^2}{2(n+2)}\kappa(v) + O(\varepsilon^3 + \varepsilon^2 d(x,y))\right)$

$\kappa > (<)\ 0 \equiv$ small balls are closer (further) than their centers

Ricci curvature κ on metric measure space

$$\kappa(x, y) = 1 - \frac{W_1(m_x, m_y)}{d(x, y)}$$

Average distance between balls centered at x and y

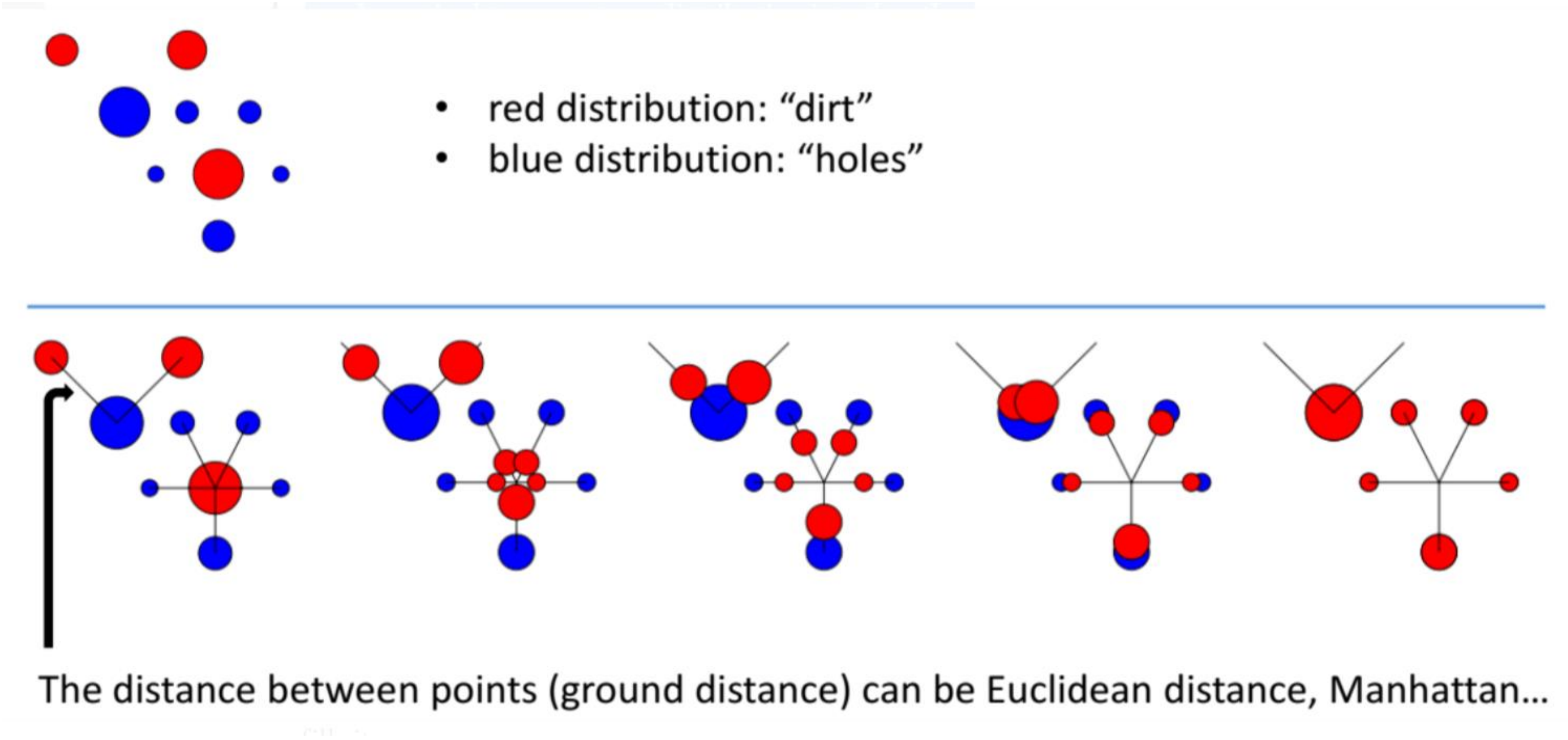
1-Wasserstein distance W_1

Distance between the centers x and y

Minimal geodesic distance d

$\kappa > (<)\ 0 \equiv$ small balls are closer (further) than their centers

1-Wasserstein distance W_1



minimum amount of work to reshape one distribution into other

$W_1(m_x, m_y)$ in metric measure space

Linear programming (LP) formulation:

$$\underset{\boldsymbol{\pi}}{\operatorname{argmin}} \sum_{x \in m_x,\, y \in m_y} d(x,y)\boldsymbol{\pi}(x,y)$$

$$\boldsymbol{\pi}(x,y) \geq 0$$

$$\sum_{y \in m_y} \boldsymbol{\pi}(x,y) = m_x$$

$$\sum_{x \in m_x} \boldsymbol{\pi}(x,y) = m_y$$

optimal transportation plan

Generalize κ to weighted directed graph

For a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$

$$\kappa(x, y) = 1 - \frac{W_1(m_x, m_y)}{d(x, y)} \longrightarrow \kappa(v_i, v_j) = 1 - \frac{W_{ij}}{d_{\text{Hop}}(v_i, v_j)}$$

directed edge ← tangent direction

W_{ij} ← distance between balls centered at v_i and v_j

d_{Hop} ← distance between the “centers” v_i and v_j

Vertex reachability

A **path** from $v_1 \in \mathcal{V}$ to $v_2 \in \mathcal{V}$ is a sequence s of directed edges

$\mathcal{S}$ is the **set of all paths**

$l_s(v_1, v_2)$ is the **length** (hop count) of the path from $v_1 \in \mathcal{V}$ to $v_2 \in \mathcal{V}$ via s

v_2 is **reachable** from v_1 (denote $v_1 \rightarrow v_2$) if $\exists$ a directed path from v_1 to v_2

Hop distance d_{Hop}

$$d(x, y) \longrightarrow d_{\text{Hop}}(v_i, v_j)$$

Geodesic distance between the centers x and y

Hop distance between the vertices v_i and v_j

$$d_{\text{Hop}}(v_1, v_2) := \begin{cases} \min_{s \in \mathcal{S}} l_s(v_1, v_2) & \text{if } v_1 \to v_2, \\ \infty & \text{if } v_1 \nrightarrow v_2, \\ 0 & \text{if } v_1 = v_2, \end{cases}$$

Single hop out-neighborhood measure balls

$W_1(m_x, m_y)$ → W_{ij}

1-Wasserstein distance between the measure balls m_x and m_y centered at x and y

1-Wasserstein distance between the single hop out-neighborhood measure balls centered at vertices v_i and v_j

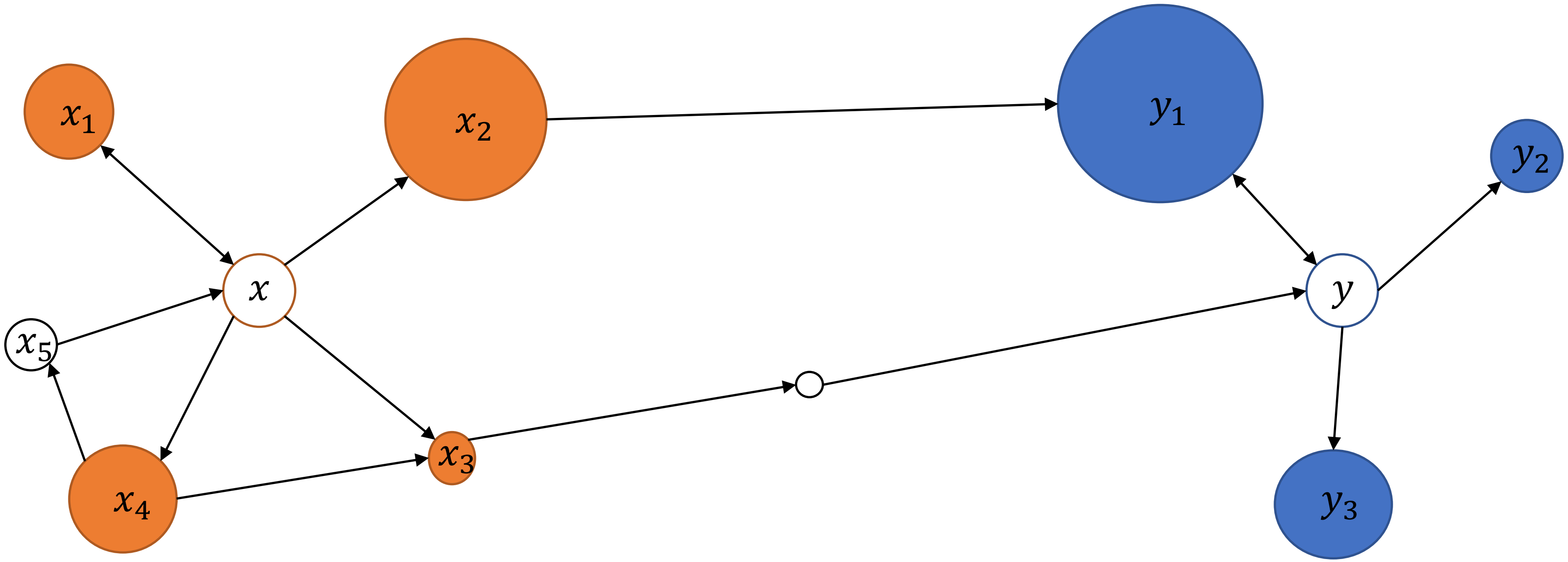
single hop out-neighborhood of x

Single hop out-neighborhood measure balls:

$$m_x = \{\mu_x(x_1), \mu_x(x_2), \dots, \mu_x(x_k)\}$$

$$\mu_x(x_i) := \begin{cases} \frac{w(xx_i)}{\sum_{j=1}^{k} w(xx_j)} & \text{if } x_i \in \mathcal{N}(x), \\ 0 & \text{otherwise.} \end{cases}$$

Single hop out-neighborhood measure balls



$d_{\text{Hop}}(x_2, y_1) = 1, \qquad d_{\text{Hop}}(x_1, y_3) = 5$

1-Wasserstein distance between single hop out-neighborhood measure balls

$W_1(m_x, m_y) \longrightarrow W_{ij}$

1-Wasserstein distance between the measure balls m_x and m_y centered at x and y

1-Wasserstein distance between the single hop out-neighborhood measure balls centered at vertices v_i and v_j

$$W_{ij} := \begin{cases} 0 & \text{if } i = j, \\ \text{undefined} & \text{if } (i \neq j) \wedge \Big((v_i \nrightarrow v_j) \vee (\mathcal{N}(v_i) = \emptyset) \vee \big(\mathcal{N}(v_j) = \emptyset\big) \Big), \\ W_1(m_{v_i}, m_{v_j}) & \text{otherwise} \end{cases}$$

LP for computing $W_1(m_{v_i}, m_{v_j})$

$$\underset{\boldsymbol{\pi}}{\operatorname{argmin}} \langle \boldsymbol{d}_{\text{Hop}}, \boldsymbol{\pi} \rangle$$

$\boldsymbol{\pi} \geq 0$ (elementwise)

$\mathbf{1}_{|m_{v_i}|}^{\top} \boldsymbol{\pi} = m_{v_j}$

$\mathbf{1}_{|m_{v_j}|}^{\top} \boldsymbol{\pi} = m_{v_i}$

can be solved as network flow problem in $\tilde{O}\left(|m_{v_i}| \times |m_{v_j}| \sqrt{|m_{v_i}| + |m_{v_j}|}\right)$ time

Simulation setup

Inter-county daily traffic data for **California during March 1st, 2020-March 31st, 2021**

Source: SafeGraph dataset “Social Distancing Metrics”

URL: https://www.safegraph.com

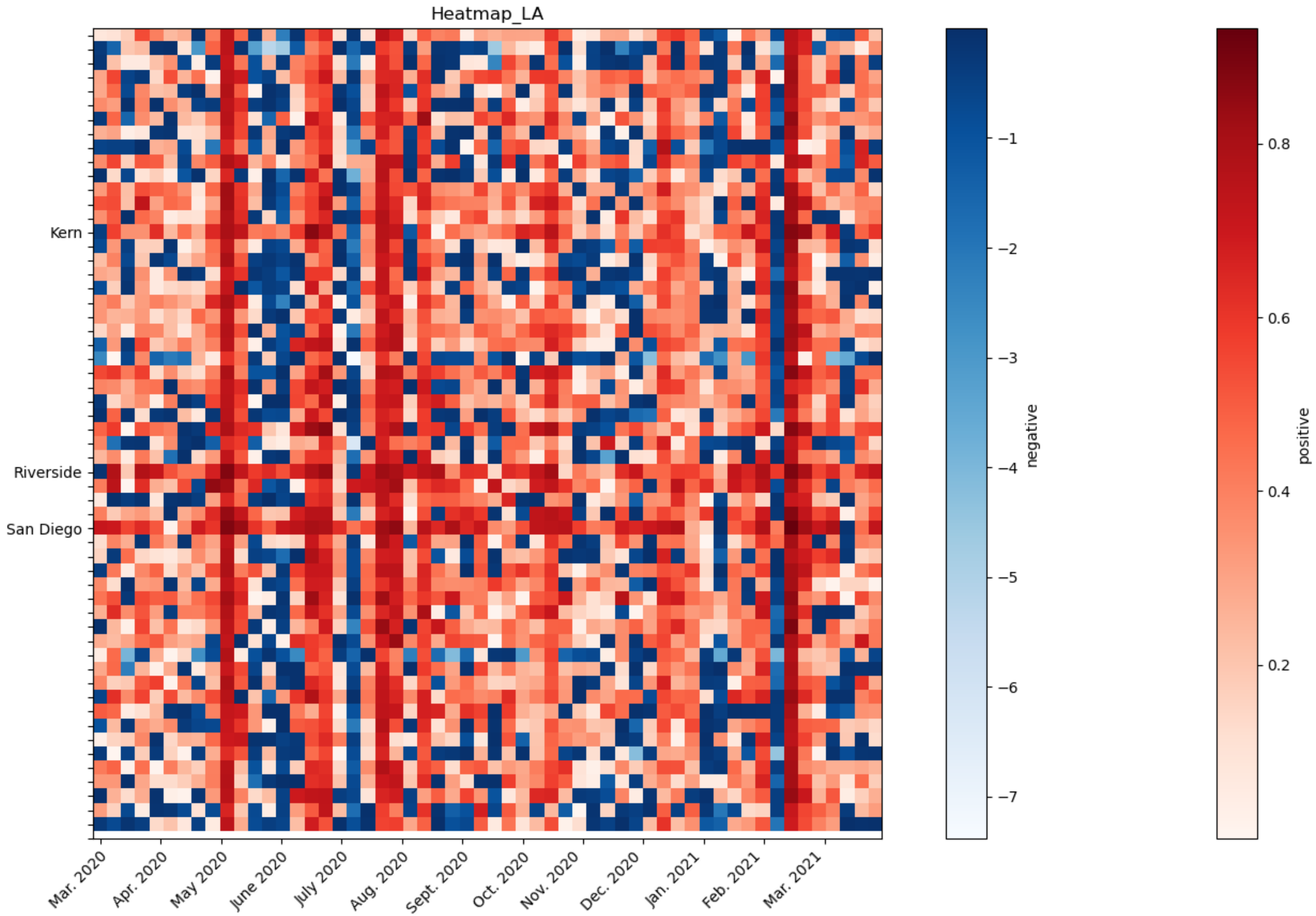
Anonymous commute data based on cellular pings and social network usage

Simulation setup

Constructed county level traffic graphs $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ and corresponding weighted adjacency matrices

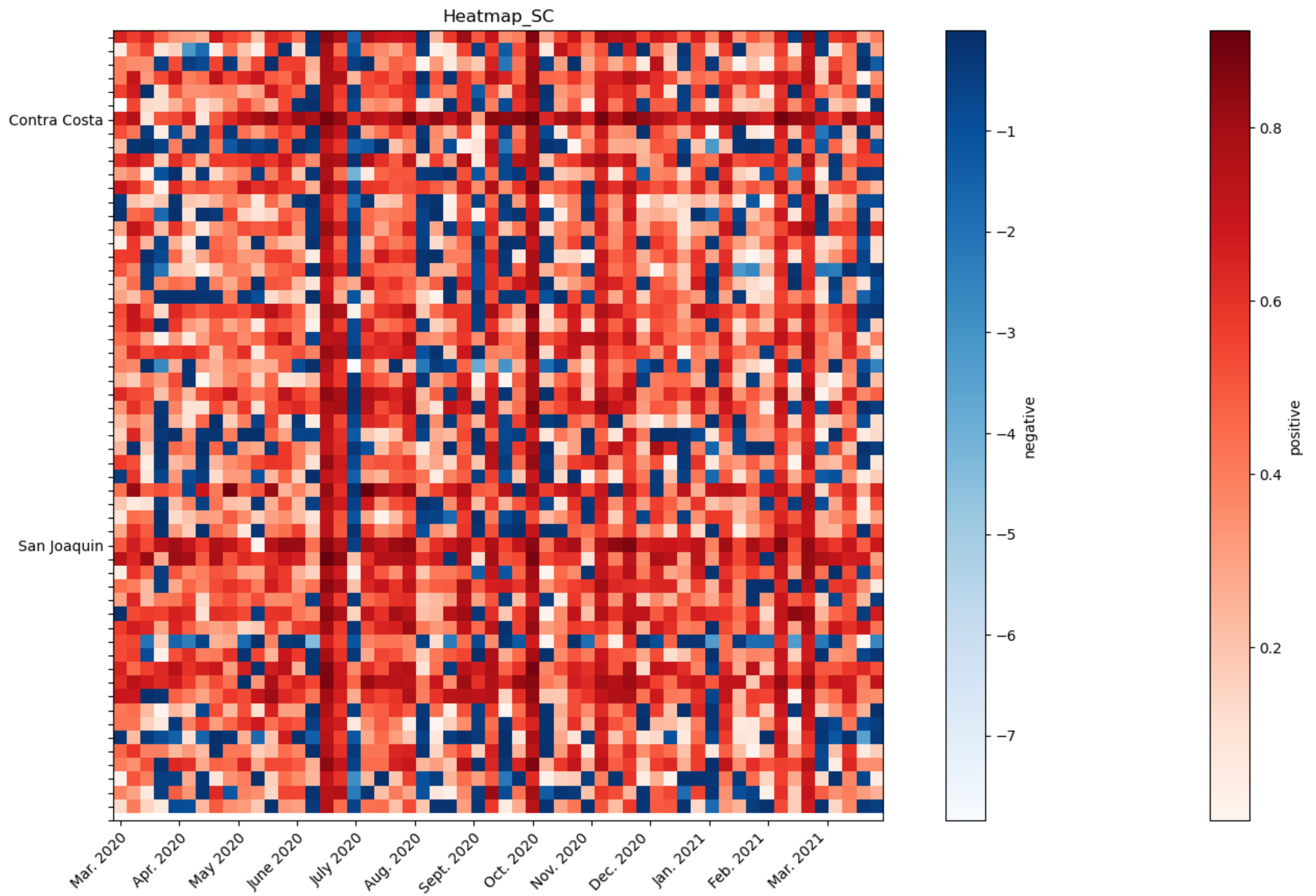
Computed outward and inward Ricci curvatures

Outward Ricci curvature: Los Angeles county

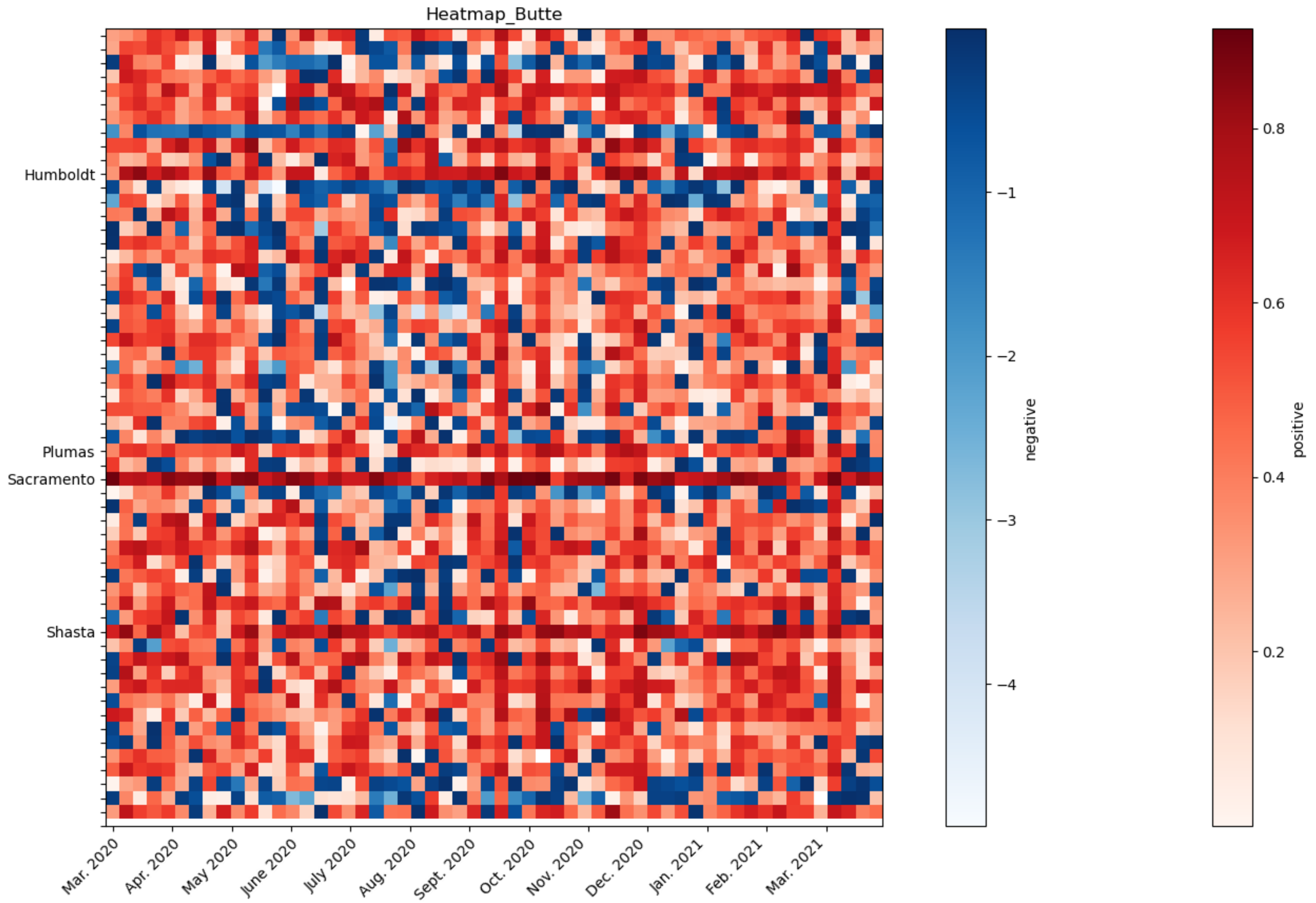



San Luis Obispo
Kern
San Bernardino
Santa Barbara
Ventura
Los Angeles
Orange
Riverside
San Diego
Imperial

Outward Ricci curvature: Santa Clara county

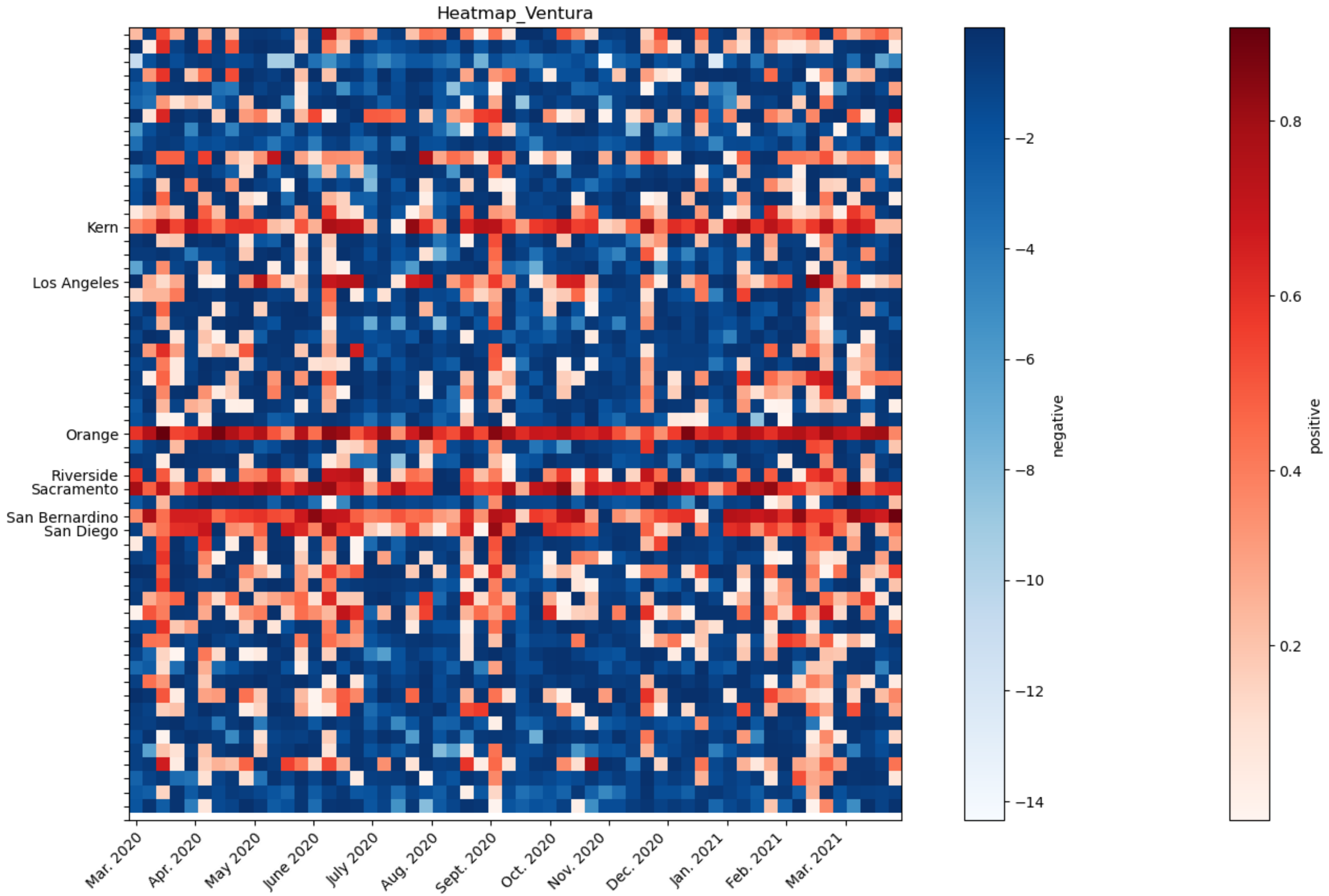


Inward Ricci curvature: Butte county




Del Norte
Siskiyou
Modoc
Shasta
Lassen
Trinity
Humboldt
Tehama
Plumas
Mendocino
Glenn
Butte
Sierra
Nevada
Yuba
Placer
Lake
Colusa
Sutter
El Dorado
Yolo
Sonoma
Napa
Alpine
Sacramento
Amador
Solano
Calaveras
Mann
San Joaquin
Tuolumne
Contra Costa

Inward Ricci curvature: Ventura county

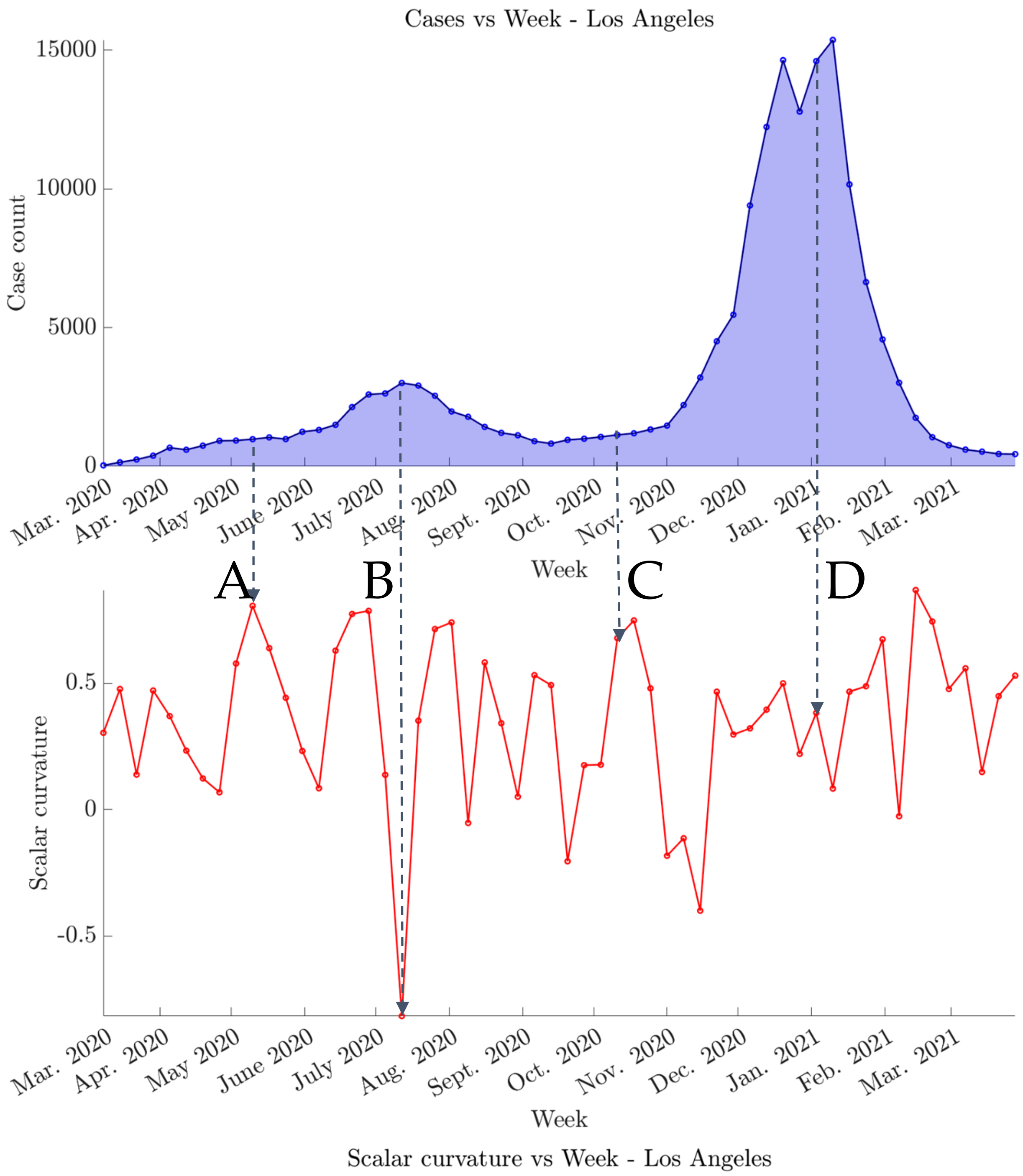


Scalar curvature

Weighted average of Ricci curvature

Defined on vertices:

$$s(v_i) := \sum_{j \in \mathcal{N}_{v_i}} \kappa(v_i, v_j)\mu_{v_i}(v_j),$$

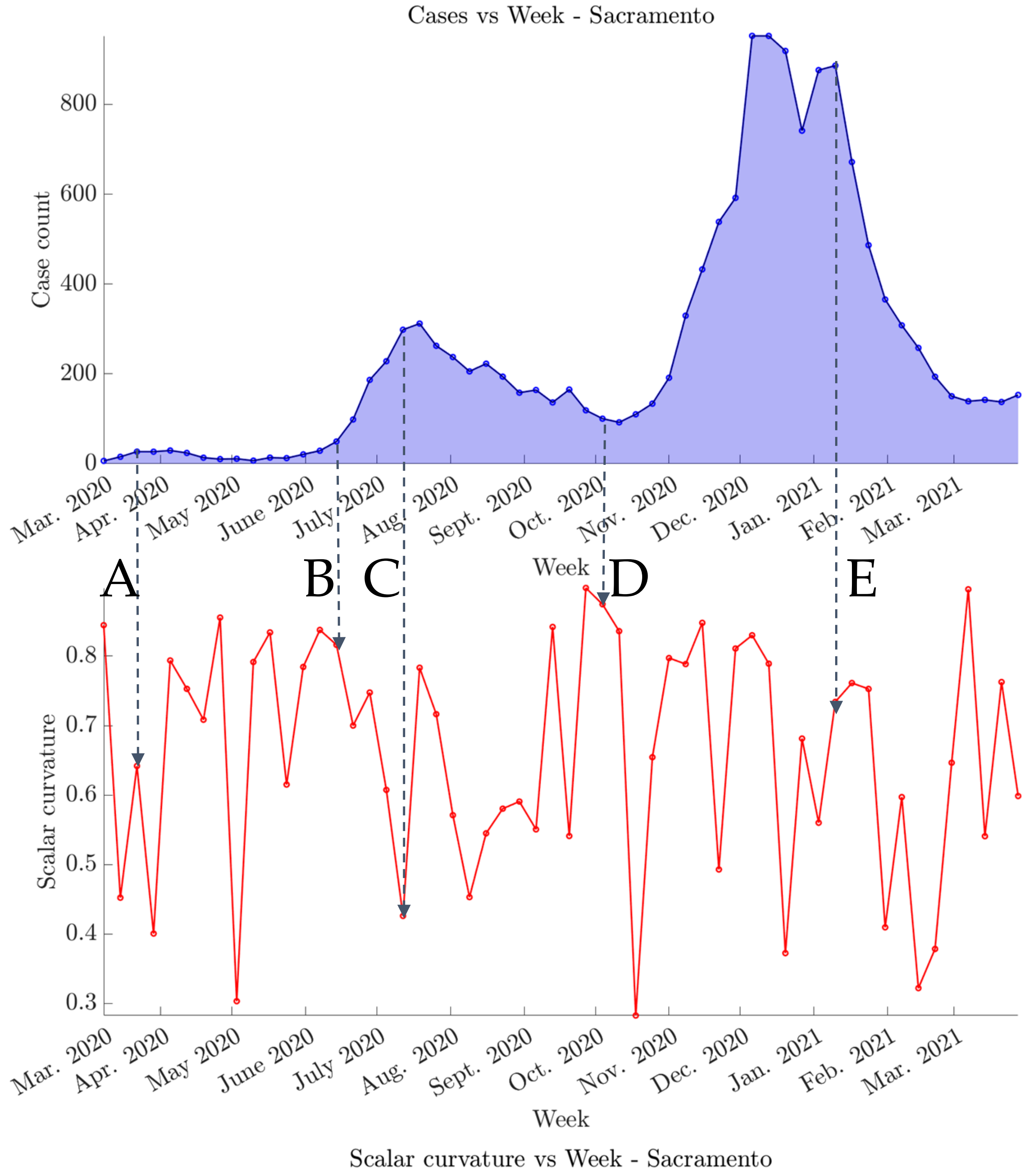


A. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up.

B. Low Scalar Curvature, harder for COVID-19 to spread, and the number of new cases has a negative acceleration. Cases vs Week graph concave down.

C. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up again.

D. COVID-19 vaccine comes out, number of new cases drops significantly as more people were vaccinated.



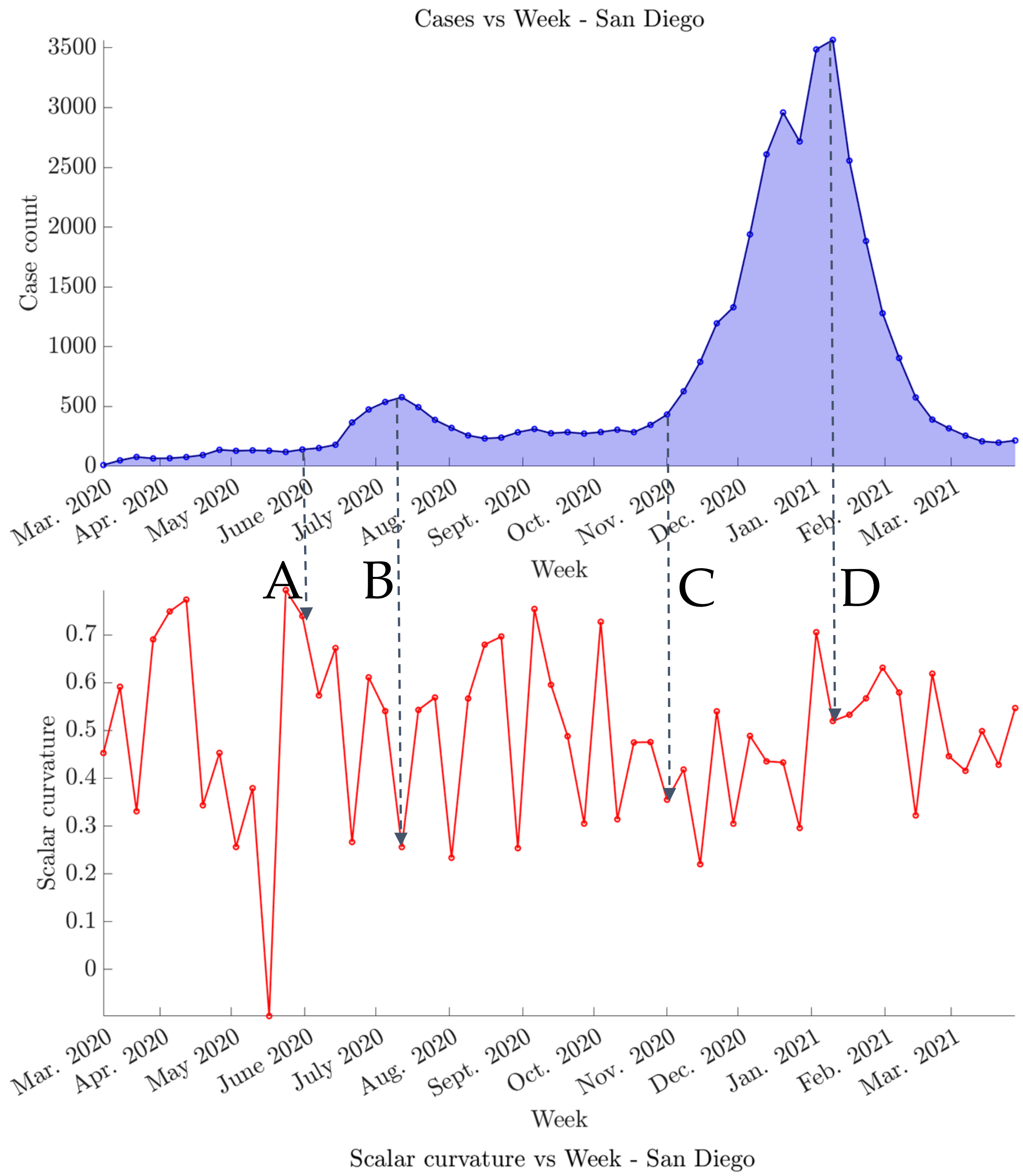
A. Low Scalar Curvature, harder for COVID-19 to spread, and the number of new cases has a negative acceleration. Cases vs Week graph concave down.

B. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up again.

C. Low Scalar Curvature, harder for COVID-19 to spread, and the number of new cases has a negative acceleration. Cases vs Week graph concave down.

D. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up again.

E. COVID-19 vaccine comes out, number of new cases drops significantly as more people were vaccinated.

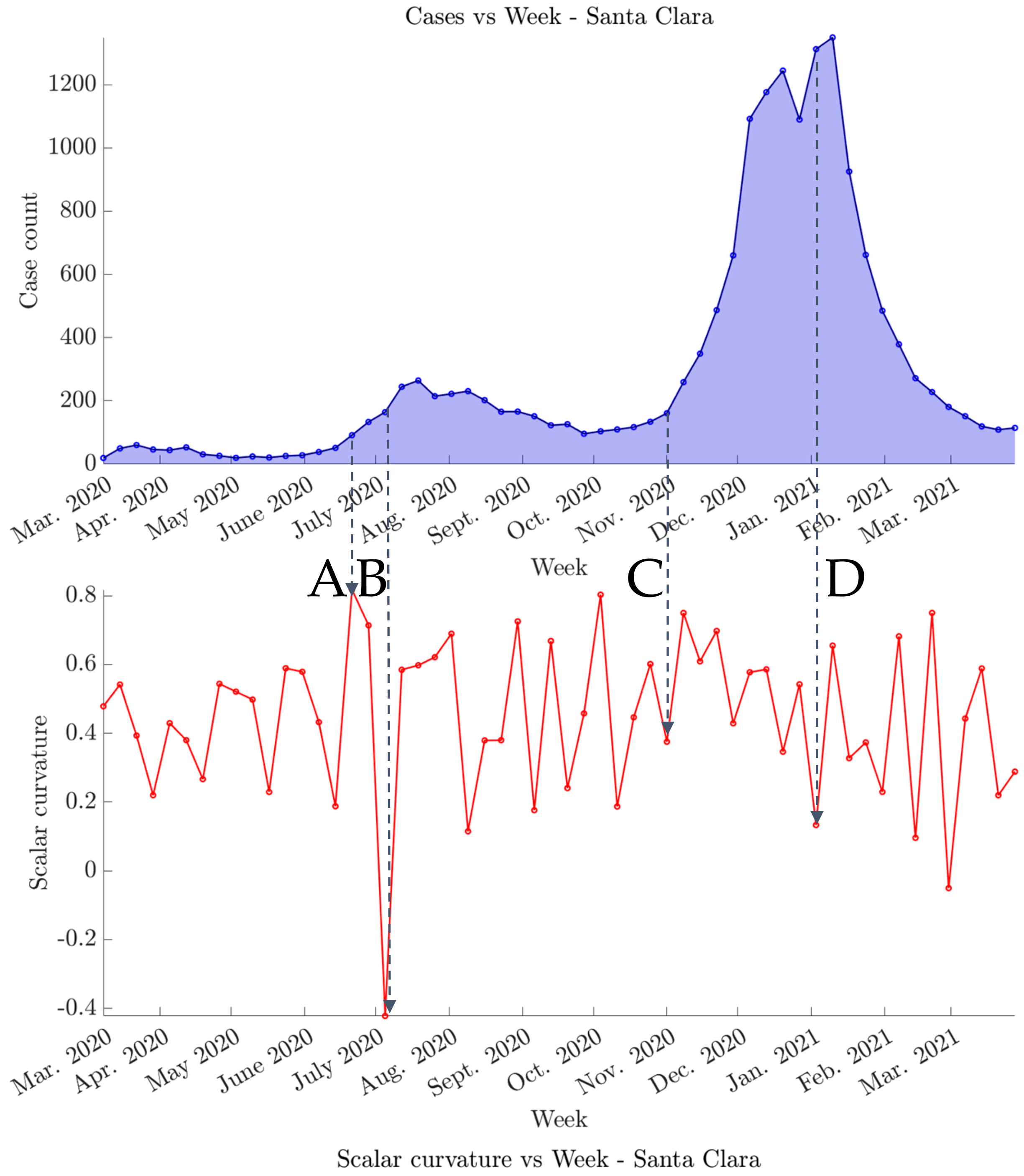


A. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up.

B. Low Scalar Curvature, harder for COVID-19 to spread, and the number of new cases has a negative acceleration. Cases vs Week graph concave down.

C. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up again.

D. COVID-19 vaccine comes out, number of new cases drops significantly as more people were vaccinated.



A. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up.

B. Low Scalar Curvature, harder for COVID-19 to spread, and the number of new cases has a negative acceleration. Cases vs Week graph concave down.

C. High Scalar Curvature, easier for COVID-19 to spread, and the number of new cases has a positive acceleration. Cases vs Week graph concave up again.

D. COVID-19 vaccine comes out, number of new cases drops significantly as more people were vaccinated.

Conclusion

Network Ricci curvature does not “average out” the pairwise interaction information

Reveals which interactions in the network are robust and which interactions are fragile

Scalar curvature is approx. predictor of case counts

Graph curvature analytics can help county officials to plan NPIs

Future research

Generalize curvatures to directed simplicial complexes to capture more than two-way interactions

Design control mechanisms to optimally (e.g., minimum effort) steer the spatial distribution of the graph Ricci curvatures over time

Thank you