Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

Visual Analysis of Spatio-temporal Phenomena with 1D Projections: Supplemental Material

M. Franke[1], H. Martin[2,3], S. Koch[1], and K. Kurzhals[1,2]

[1]University of Stuttgart, Germany [2]ETH Zurich, Switzerland [3]Institute of Advanced Research in Artificial Intelligence, Vienna, Austria

1. Quality Metrics

1.1. M_1 and M_2 Scores

The scores as defined in [VK01, Eq.s 1–2] are:

$$M_1(k) = 1 - \frac{2}{Nk(2N-3k-1)} \sum_{i=1}^{N} \sum_{x_j \in U_k(x_i)} (r(x_i,x_j) - k) \quad (1)$$

$$M_2(k) = 1 - \frac{2}{Nk(2N-3k-1)} \sum_{i=1}^{N} \sum_{x_j \in V_k(x_i)} (\hat{r}(x_i,x_j) - k), \quad (2)$$

The symbols and variables are the same as in the original work and are reproduced in Table 1. The scores M_1, M_2 lie between 0 and 1, where a value of 1 is desirable; that is, $M_1 = 1$ implies that the low-dimensional k-neighborhoods are free from data vectors that were not part of the high-dimensional k-neighborhood; and $M_2 = 1$ implies that the high-dimensional k-neighborhoods are preserved in the low-dimensional space.

Table 1: *Symbols and variables used in Equations (1) and (2). Reproduced from [VK01, Tab. 1].*

$x_i \in R^2, \ i = 1,\ldots,N$	Data vector
$C_k(x_i)$	The set of those k data vectors closest to x_i in the original space
$\hat{C}_k(x_i)$	The set of those k data vectors closest to x_i after projection
$U_k(x_i)$	The set of data vectors x_j for which $x_j \in \hat{C}_k(x_i) \wedge x_j \notin C_k(x_i)$ holds
$V_k(x_i)$	The set of data vectors x_j for which $x_j \notin \hat{C}_k(x_i) \wedge x_j \in C_k(x_i)$ holds
$r(x_i,x_j), \ i \neq j$	The rank of x_j when the data vectors are ordered based on their Euclidean distance from the data vector x_i, in the original space
$\hat{r}(x_i,x_j), \ i \neq j$	The rank of x_j when the data vectors are ordered based on their Euclidean distance from the data vector x_i, after projection

1.2. Metric and Non-metric Stress

Given n points $\{x_1,x_2,...,x_n\} \in \mathbb{R}^2$ in the original space of our data, and $\{\hat{x}_1,\hat{x}_2,...,\hat{x}_n\} \in \mathbb{R}$ as the corresponding projections of these points into the one-dimensional space. We define d_{ij} as a measure of distance between x_i and x_j, and δ_{ij} as the distances between the points $\hat{x}_i$ and $\hat{x}_j$. The metric stress S_m and the non-metric stress S_{nm} can be defined as

$$S_m = \sqrt{\frac{\sum_{i<j}(d_{ij} - \delta_{ij})^2}{\sum_{i<j} d_{ij}^2}} \quad (3)$$

$$S_{nm} = \sqrt{\frac{\sum_{i<j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} d_{ij}^2}}, \quad (4)$$

where $\hat{d}_{ij}$ are called *disparities*. The *disparities* are chosen such that S_{nm} is minimized under the constraint that all $\hat{d}_{ij}$ have the same rank order as the δ_{ij}. For both measures, a low value indicates a good projection. Metric stress is 0 if all pairwise distances in the projected space are equal to the corresponding pairwise distance in the original space. Non-metric stress operates on the rank order of the distances and is zero if the rank order of the distances in the projected space is the same as the rank order in the projected space. For the calculation of S_m we first normalize the distances matrices to values between 0 and 1 by division with the maximum distance.

2. Projections on Artificial Datasets

We generated small test datasets programmatically and then projected the resulting datasets using our approach. The first two datasets are generated on a regular grid, similar to the wildfire data. The third dataset is generated on more randomly distribution points, similar to the COVID-19 data.

2.1. Spreading Ring

The first test dataset we generated was of a *hotspot* that started as a spot, and then spread outwards as a ring over time, with the values falling off in the middle. We observed such patterns, for example, in the wildfire dataset, where fires would start in one place, spread outward, and run out of fuel in the center. We define an epicenter $\vec{c}$ and parameterize the ring-like value progression over time with a

peak value $\hat{p}(t)$, the median radius of the ring $\mu(t)$, and the standard deviation of the ring's values $\sigma(t)$:

$$\mu(t) = \mu_0 + \mu_a t \tag{5}$$
$$\sigma(t) = \sigma_0 + \sigma_a t \tag{6}$$
$$\hat{p}(t) = \hat{p}_0 + \hat{p}_a t \tag{7}$$

The value at point $\vec{x}$ at time t is then defined as:

$$\text{ring}(\vec{x},t) = \hat{p}(t) \cdot \exp\left[\frac{1}{2} \cdot \left(\frac{\|\vec{x}-\vec{c}\|_2 - \mu(t)}{\sigma(t)}\right)^2\right] \tag{8}$$

The figures in the paper were generated over 40 time steps (0..39), on a 14×14 grid with cells of size 15×15, with the following parameters:

$$\sigma_0 = 12.16 \qquad \sigma_a = 0.05 \tag{9}$$
$$\mu_0 = 0 \qquad \mu_a = 1.94 \tag{10}$$
$$\hat{p}_0 = 1 \qquad \hat{p}_a = 0.3 \tag{11}$$
$$\vec{c} = \begin{pmatrix} 30.4 \\ 40.6 \end{pmatrix} \tag{12}$$

In Tables 2 and 3, we apply the projections to datasets with different parametrizations of the *spreading ring* function. The parametrizations are provided at the top of each column. We can observe that the spreading pattern is clear in all parametrizations, but that it is easier to detect for thicker rings (i.e., larger σ). We conclude that the patterns we observe are stable for different shapes and sizes.

2.2. Wandering Hotspot

A second dataset generates a circular region of higher values, which moves with a constant velocity and direction $\vec{v}$. The circular region is a ring as defined in Equation (8), but with a constant value inside:

$$\text{wander}(\vec{x},t) = \begin{cases} \hat{p}(t) & \|\vec{x}-\vec{c}(t)+t\vec{v}\| \le \mu(t) \\ \text{ring}(\vec{x}-t\vec{v},t) & \text{otherwise} \end{cases} \tag{13}$$

The figures in the paper were generated over 40 time steps (0..39), on a 14×14 grid with cells of size 15×15, with the following parameters:

$$\sigma_0 = 12.3 \qquad \sigma_a = 0.03 \tag{14}$$
$$\mu_0 = 0.95 \qquad \mu_a = 0 \tag{15}$$
$$\hat{p}_0 = 2.3 \qquad \hat{p}_a = 0 \tag{16}$$
$$\vec{c} = \begin{pmatrix} 36 \\ 38 \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} 0.416 \\ 2.63 \end{pmatrix} \tag{17}$$

In Tables 4 and 5, we apply the projections to datasets with different parametrizations of the *wandering hotspot* function. The parametrizations are provided at the top of each column. We can observe how the pattern is visible regardless of movement direction, speed, or size. We conclude that the patterns we observe here are also stable.

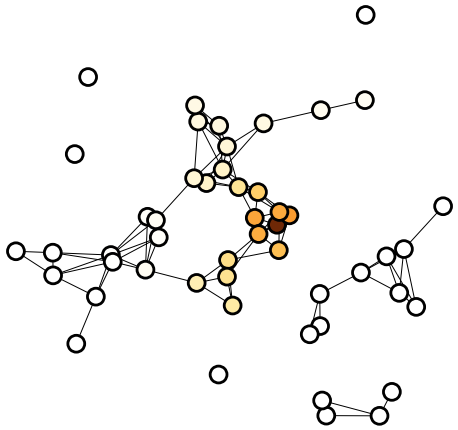


Figure 1: *The graph used for the graph-based spreading dataset in the paper at $t = 13$.*

2.3. Graph-based Spreading

For the third dataset, we generated 50 random points using a normal distribution random number generator with $\vec{\mu} = \binom{0.5}{0.5}$ and $\vec{\sigma} = \binom{0.2}{0.21}$. These points made the basis for the vertices of a k-nearest-neighbors graph $G(V,E)$ with $k = 4$ and a cutoff distance $d_{\text{cut}} = 0.14$. The specific graph is shown for $t = 13$ in Figure 1. Using a cutoff radius meant that the resulting graph would not be connected.

We picked a central vertex c of the graph (dark red in Figure 1), and pinned its value d_c to a normal distribution over time:

$$d_c(t) = \exp\left(-\frac{1}{2}\left[\frac{t-15}{10}\right]^2\right) \tag{18}$$

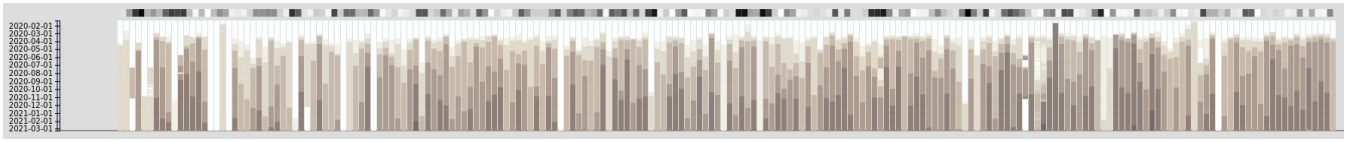
For all remaining vertices i, the value d_i of the next time step was calculated from its own value, as well as the values of all its connected neighbors:

$$d_i(0) = 0 \tag{19}$$
$$d_i(t+1) = d_i(t) + \frac{1}{|\{e \in E \mid i \in e\}|} \sum_{\substack{(u,v)\in E \\ u=i}} \frac{d_v - d_u}{1 + \|\vec{u}-\vec{v}\|_2^2} \tag{20}$$

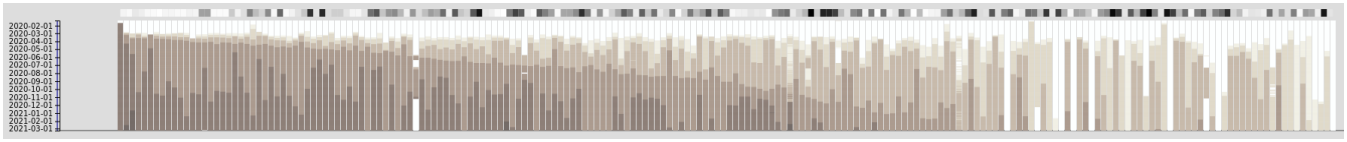
In Tables 6 to 8, we apply the projections to datasets with different graphs of different sizes of the *graph-based spreading* example. As the graphs are constructed with a fixed cutoff radius, the AHC-based projections work really well, because the clustering prioritizes entities that are closer together. The space-filling curve-based projections introduce discontinuities, but the AHC-based ones, especially single linkage, perform really well and reveal a consistent pattern.

3. Other Projections

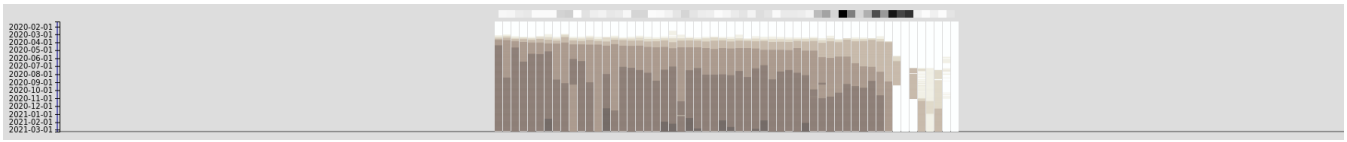
During the evolution of our work, we tried out some other projections. In both cases, we were able to produce a proof of concept.



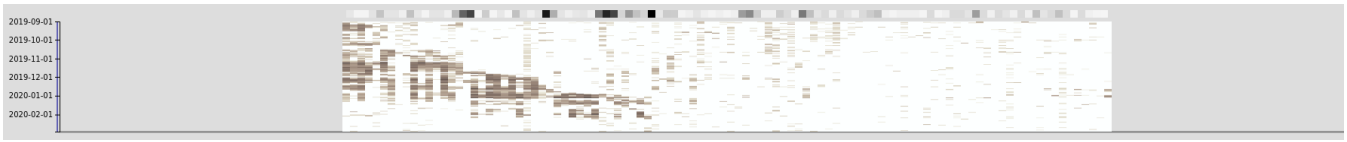
(a) *Projected timeline of countries of the world for the COVID-19 application scenario using air traffic passenger flow as an inverse distance metric between countries.*



(b) First-peak *projection for countries of the world for the COVID-19 application scenario.*



(c) First-peak *projection for the US states for the COVID-19 application scenario.*



(d) First-peak *projection for the top-level entities of the wildfire application scenario.*

Figure 2: *Examples for projected timelines with the two other projections discussed in Section 3.*

We could not produce anything worth discussing in the paper using our rudimentary methods, but got some insights that we wanted to discuss and share in the supplemental material and show some screenshots (Figure 2).

3.1. Passenger Flow

The passenger flow projection was suggested to us by one of the experts. We produced a proof of concept using an air traffic dataset by Huang et al. [HWG*13]. The dataset contains passenger numbers between countries for flights from 2010. It is fairly complete, but there are pairs of countries for which no data exist. We calculate our distance metric between countries c_1, c_2 as follows: If there is a non-zero passenger flow f_{c_1,c_2} between the countries, the distance between them is the inverse of that flow; otherwise, we fall back to Euclidean distance:

$$d_{c_1,c_2} = \begin{cases} f_{c_1,c_2}^{-1} & f_{c_1,c_2} \neq 0 \\ \|\vec{c_2} - \vec{c_1}\|_2 & \text{otherwise} \end{cases} \tag{21}$$

Since the Euclidean distances are all > 1, this means that countries with non-zero passenger flow between them are always closer together. Further, the flight data does not contain any data about sub-country passenger flow, or other means of travel, for example, by car between neighboring countries. Last, the data is not current, and there is no guarantee passenger flows have stayed the same since 2010. For applications concerning, for instance, epidemiology, this type of projection can be very valuable. However, in order to produce good results, a real-world application would require a more complete and up-to-date passenger flow dataset, the likes of which we did not have available for our proof of concept. We have included a screenshot in Figure 2a. We notice that the pairwise distance indicators are very often dark, indicating larger geospatial distances. This is probably due to the fact that air travel is not the main mode of transport between adjacent countries.

3.2. First-peak

We also implemented a projection that orders geospatial entities based on the first time its time series value crosses a threshold. In our implementation, we set that threshold to 1 % of the maximum time series value in all entities of the subtree. The rationale behind this was that this eliminated the effects of slight noise that appear in the data. Figure 2d shows this quite nicely: A clear front of higher values can be seen in the right side of the timeline despite these time series having non-zero values before the peaks.

We can see the usefulness of such a projection for the identification of *trendsetters* in a dataset. However, we see a limitation with the naïve approach in that a global first value is considered. For time series with a continuously increasing nature, such as accumulated COVID-19 cases, this might work very well. We can see this for example with the countries of the world in Figure 2b and the states of the US in Figure 2c. For time series that rise and fall, such as the wildfire scenario, the global first peak might not be what we want to look at, as we can see in Figure 2d. This projection could be further improved to be applicable to more tasks by interactively defining a time frame within which the first peak is considered,

and by being able to interactively tweak the threshold. That would, however, require projections to be calculated on the fly rather than in a preprocessing step. We argue that this projection is useful for very specific tasks and should only serve as an example for what is possible for individual applications of our approach.

References

[FMKK21] FRANKE, MAX, MARTIN, HENRY, KOCH, STEFFEN, and KURZHALS, KUNO. *Visual Analysis of Spatio-temporal Phenomena with 1D Projections: Code Repository*. 2021. URL: https://github.com/UniStuttgart-VISUS/spatiotemporal1d (visited on 04/18/2021).

[HWG*13] HUANG, ZHUOJIE, WU, XIAO, GARCIA, ANDRES J, et al. "An open-access modeled passenger flow matrix for the global air network in 2010". *PloS one* 8.5 (2013), e64317. DOI: 10.1371/journal.pone.0064317 3.

[VK01] VENNA, JARKKO and KASKI, SAMUEL. "Neighborhood preservation in nonlinear projection methods: An experimental study". *Artificial Neural Networks — ICANN 2001*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, 485–491. DOI: 10.1007/3-540-44668-0_68 1.

Table 2: *Different parametrization results for* spreading ring *dataset.*

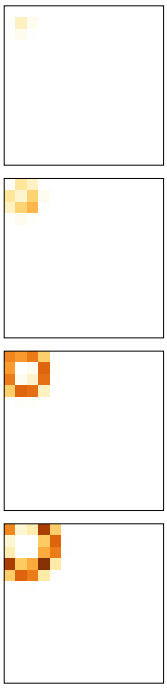
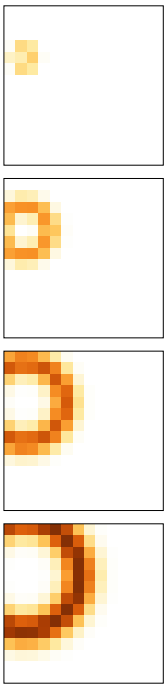
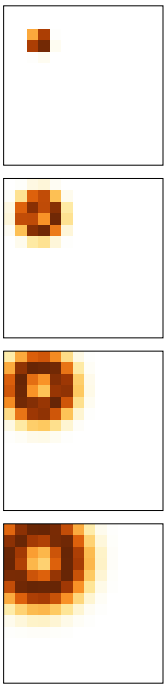
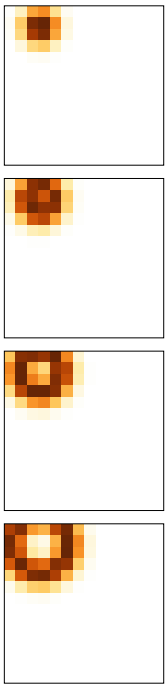
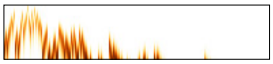
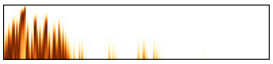
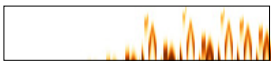
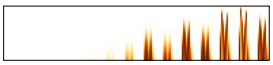
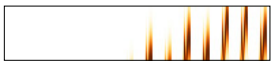
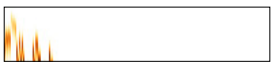
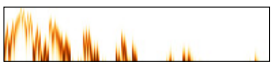
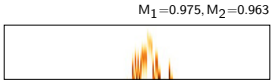
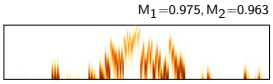
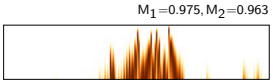
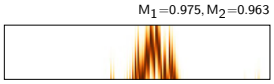
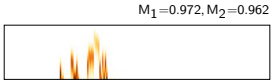
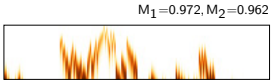
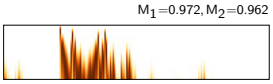
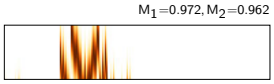
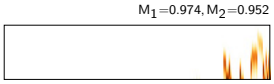
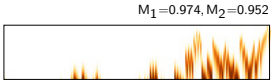
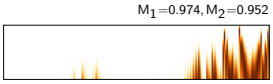
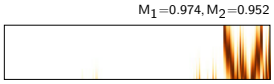
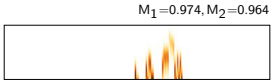
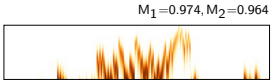
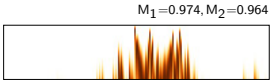
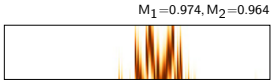
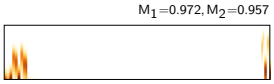
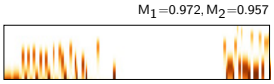
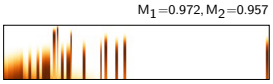
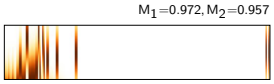
$\mu(t) = 1t + 0$ $\sigma(t) = 0.1t + 3$ $\hat{p}(t) = 0.2t + 0$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}20\\20\end{pmatrix}$	$\mu(t) = 2t + 1.9$ $\sigma(t) = 0.3t + 4$ $\hat{p}(t) = 0.1t + 1$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}20\\60\end{pmatrix}$	$\mu(t) = 1t + 3.3$ $\sigma(t) = 0.5t + 2$ $\hat{p}(t) = 0t + 1$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}40\\40\end{pmatrix}$	$\mu(t) = 1t + 1.9$ $\sigma(t) = 0t + 12$ $\hat{p}(t) = 0t + 2$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}40\\20\end{pmatrix}$
			
			
			
			
			
			
			
			
			
			

Table 3: *Different parametrization results for* spreading ring *dataset.*

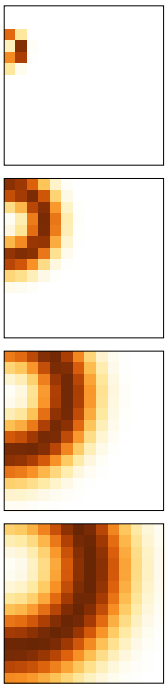
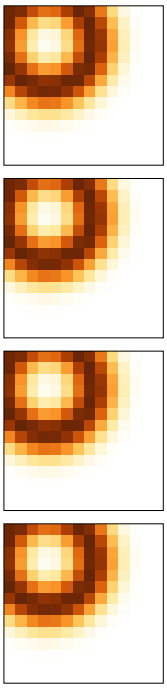
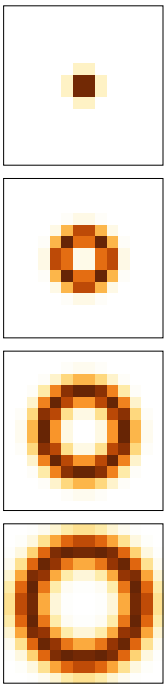
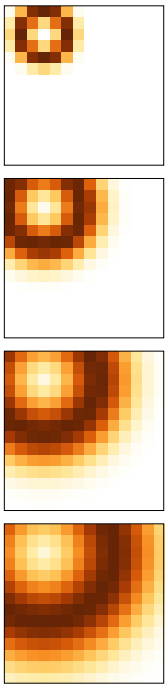
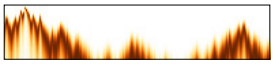
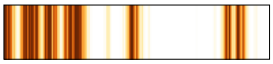
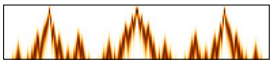
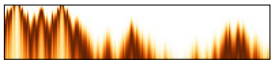
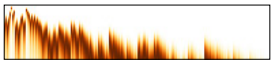
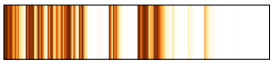
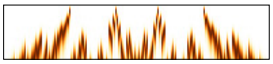
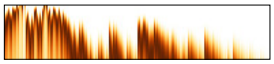
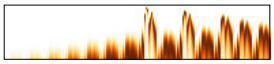
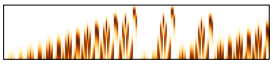
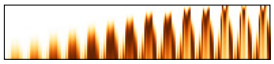
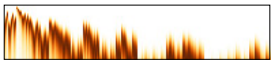
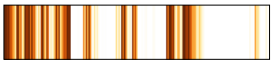
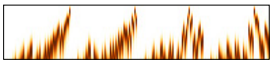
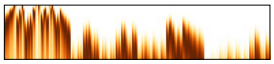
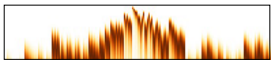
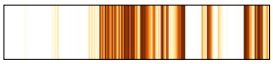
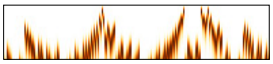
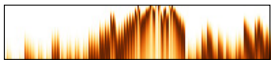
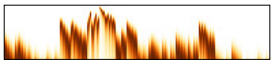
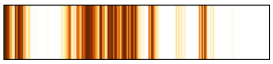
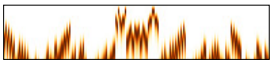
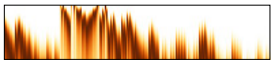
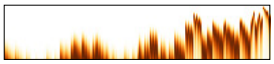
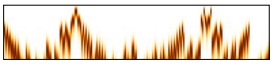
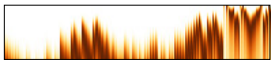
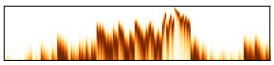
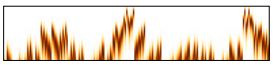
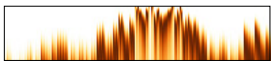
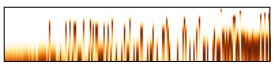
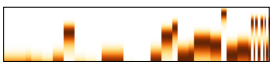
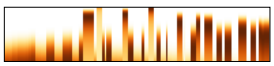
$\mu(t) = 3t + 0.5$ $\sigma(t) = 1t + 0.2$ $\hat{p}(t) = 0.01t + 4$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}0\\50\end{pmatrix}$	$\mu(t) = 0t + 60$ $\sigma(t) = 0t + 20$ $\hat{p}(t) = 0t + 1$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}50\\40\end{pmatrix}$	$\mu(t) = 2t + 2$ $\sigma(t) = 0.3t + 4$ $\hat{p}(t) = 0t + 1$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}97.5\\97.5\end{pmatrix}$	$\mu(t) = 2t + 20$ $\sigma(t) = 1t + 4$ $\hat{p}(t) = 0t + 1$ $\vec{c} = \begin{pmatrix}0\\0\end{pmatrix} t + \begin{pmatrix}45\\30\end{pmatrix}$
			
			
			
			
			
			
			
			
			
			

Table 4: *Different parametrization results for* wandering hotspot *dataset.*

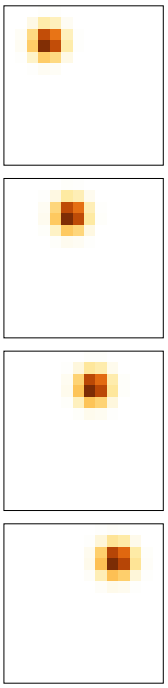
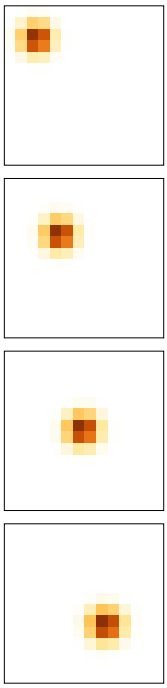
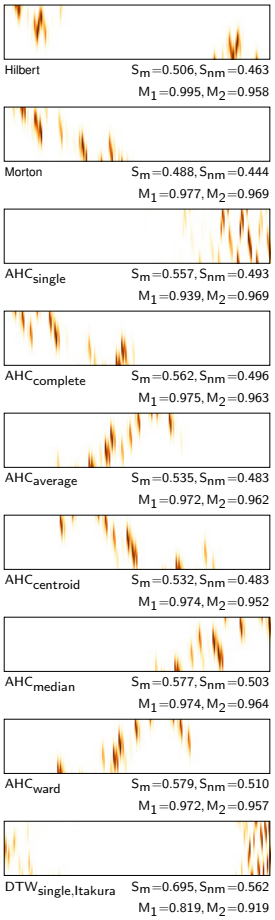
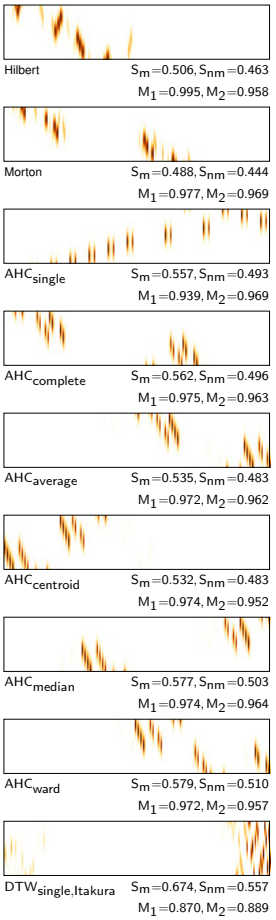
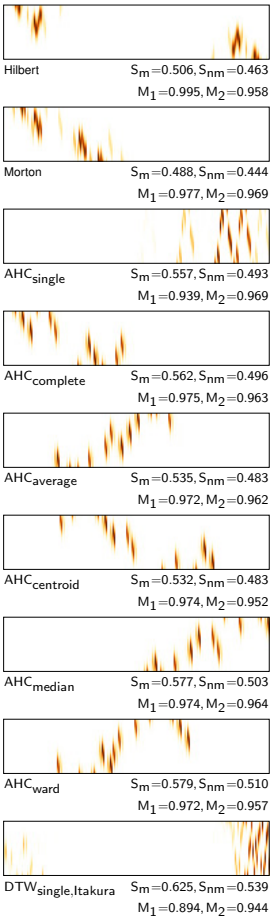
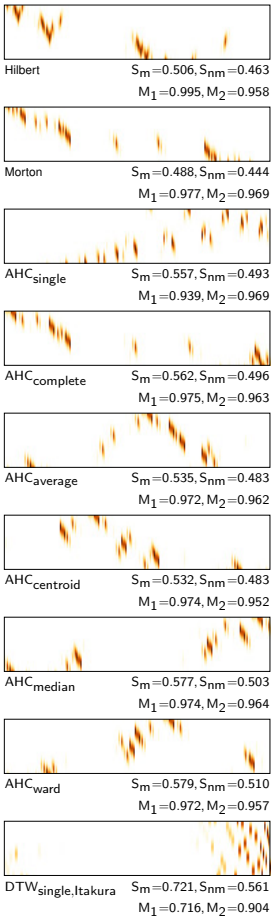
$\mu(t) = 0t + 0.95$ $\sigma(t) = 0.03t + 12.3$ $\hat{p}(t) = 0t + 2.3$ $\vec{c} = \begin{pmatrix} 0.416 \\ 2.63 \end{pmatrix} t + \begin{pmatrix} 36.4 \\ 38 \end{pmatrix}$	$\mu(t) = 0t + 0.95$ $\sigma(t) = 0.03t + 12.3$ $\hat{p}(t) = 0t + 2.3$ $\vec{c} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} t + \begin{pmatrix} 35 \\ 40 \end{pmatrix}$	$\mu(t) = 0t + 0.95$ $\sigma(t) = 0.03t + 12.3$ $\hat{p}(t) = 0t + 2.3$ $\vec{c} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} t + \begin{pmatrix} 35 \\ 40 \end{pmatrix}$	$\mu(t) = 0t + 0.95$ $\sigma(t) = 0.03t + 12.3$ $\hat{p}(t) = 0t + 2.3$ $\vec{c} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} t + \begin{pmatrix} 20 \\ 20 \end{pmatrix}$
			
			

Table 5: *Different parametrization results for* wandering hotspot *dataset.*

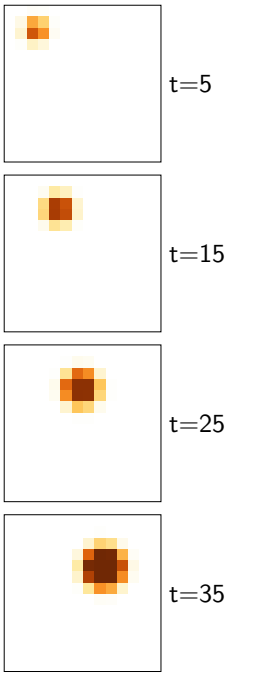
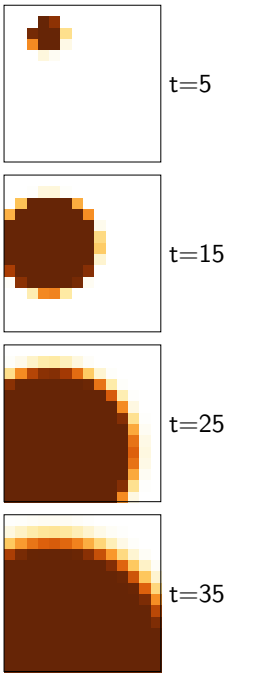
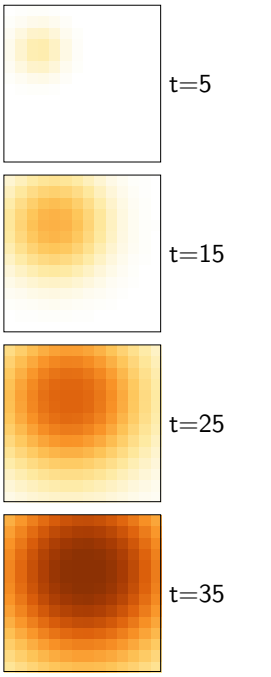
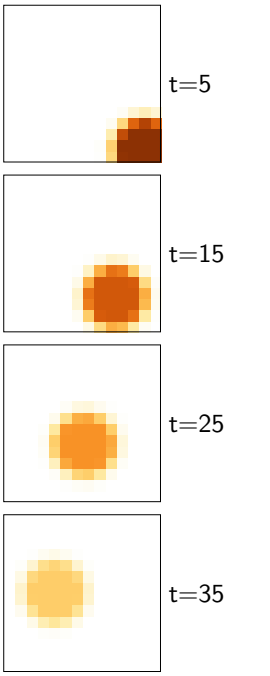
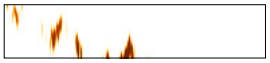
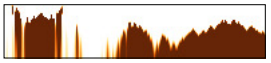
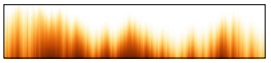
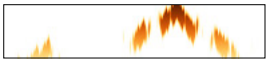
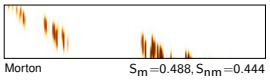
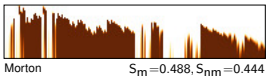
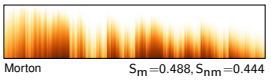
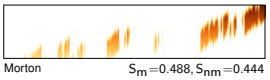
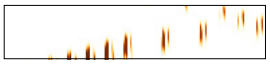
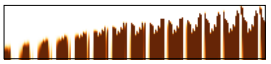
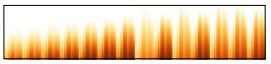
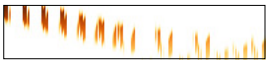
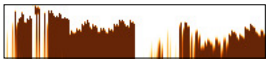
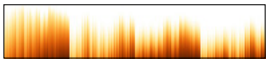
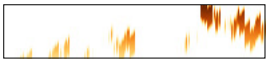
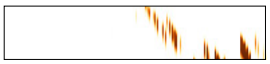
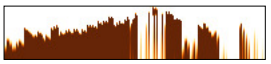
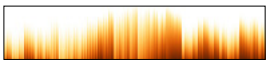
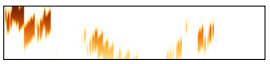
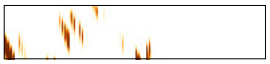
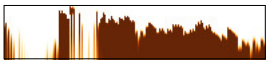
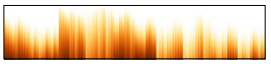
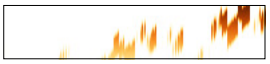
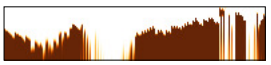
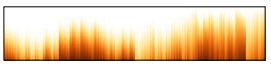
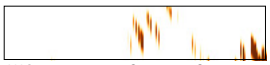
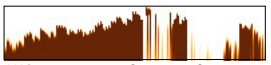
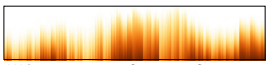
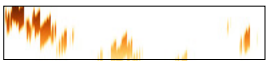
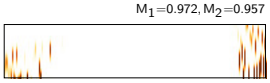
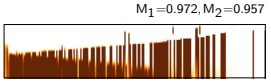
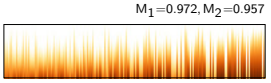
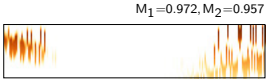
$\mu(t) = 0.5t + 1$ $\sigma(t) = 0.02t + 8$ $\hat{p}(t) = 0.01t + 1$ $\vec{c} = \begin{pmatrix} 3 \\ 1.2 \end{pmatrix} t + \begin{pmatrix} 20 \\ 20 \end{pmatrix}$	$\mu(t) = 4t + 1$ $\sigma(t) = -0.6t + 5$ $\hat{p}(t) = 0t + 1$ $\vec{c} = \begin{pmatrix} 0.2 \\ 5 \end{pmatrix} t + \begin{pmatrix} 50 \\ 10 \end{pmatrix}$	$\mu(t) = 0.5t + 0$ $\sigma(t) = 2t + 12$ $\hat{p}(t) = 1t + 1$ $\vec{c} = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix} t + \begin{pmatrix} 30 \\ 50 \end{pmatrix}$	$\mu(t) = 0t + 30$ $\sigma(t) = 0t + 10$ $\hat{p}(t) = -0.02t + 1$ $\vec{c} = \begin{pmatrix} -4 \\ -3 \end{pmatrix} t + \begin{pmatrix} 200 \\ 200 \end{pmatrix}$
			
 Hilbert S_m=0.506, S_{nm}=0.463 M_1=0.995, M_2=0.958	 Hilbert S_m=0.506, S_{nm}=0.463 M_1=0.995, M_2=0.958	 Hilbert S_m=0.506, S_{nm}=0.463 M_1=0.995, M_2=0.958	 Hilbert S_m=0.506, S_{nm}=0.463 M_1=0.995, M_2=0.958
 Morton S_m=0.488, S_{nm}=0.444 M_1=0.977, M_2=0.969	 Morton S_m=0.488, S_{nm}=0.444 M_1=0.977, M_2=0.969	 Morton S_m=0.488, S_{nm}=0.444 M_1=0.977, M_2=0.969	 Morton S_m=0.488, S_{nm}=0.444 M_1=0.977, M_2=0.969
 AHC_{single} S_m=0.557, S_{nm}=0.493 M_1=0.939, M_2=0.969	 AHC_{single} S_m=0.557, S_{nm}=0.493 M_1=0.939, M_2=0.969	 AHC_{single} S_m=0.557, S_{nm}=0.493 M_1=0.939, M_2=0.969	 AHC_{single} S_m=0.557, S_{nm}=0.493 M_1=0.939, M_2=0.969
 $AHC_{complete}$ S_m=0.562, S_{nm}=0.496 M_1=0.975, M_2=0.963	 $AHC_{complete}$ S_m=0.562, S_{nm}=0.496 M_1=0.975, M_2=0.963	 $AHC_{complete}$ S_m=0.562, S_{nm}=0.496 M_1=0.975, M_2=0.963	 $AHC_{complete}$ S_m=0.562, S_{nm}=0.496 M_1=0.975, M_2=0.963
 $AHC_{average}$ S_m=0.535, S_{nm}=0.483 M_1=0.972, M_2=0.962	 $AHC_{average}$ S_m=0.535, S_{nm}=0.483 M_1=0.972, M_2=0.962	 $AHC_{average}$ S_m=0.535, S_{nm}=0.483 M_1=0.972, M_2=0.962	 $AHC_{average}$ S_m=0.535, S_{nm}=0.483 M_1=0.972, M_2=0.962
 $AHC_{centroid}$ S_m=0.532, S_{nm}=0.483 M_1=0.974, M_2=0.952	 $AHC_{centroid}$ S_m=0.532, S_{nm}=0.483 M_1=0.974, M_2=0.952	 $AHC_{centroid}$ S_m=0.532, S_{nm}=0.483 M_1=0.974, M_2=0.952	 $AHC_{centroid}$ S_m=0.532, S_{nm}=0.483 M_1=0.974, M_2=0.952
 AHC_{median} S_m=0.577, S_{nm}=0.503 M_1=0.974, M_2=0.964	 AHC_{median} S_m=0.577, S_{nm}=0.503 M_1=0.974, M_2=0.964	 AHC_{median} S_m=0.577, S_{nm}=0.503 M_1=0.974, M_2=0.964	 AHC_{median} S_m=0.577, S_{nm}=0.503 M_1=0.974, M_2=0.964
 AHC_{ward} S_m=0.579, S_{nm}=0.510 M_1=0.972, M_2=0.957	 AHC_{ward} S_m=0.579, S_{nm}=0.510 M_1=0.972, M_2=0.957	 AHC_{ward} S_m=0.579, S_{nm}=0.510 M_1=0.972, M_2=0.957	 AHC_{ward} S_m=0.579, S_{nm}=0.510 M_1=0.972, M_2=0.957
 $DTW_{single,Itakura}$ S_m=0.683, S_{nm}=0.559 M_1=0.842, M_2=0.897	 $DTW_{single,Itakura}$ S_m=0.605, S_{nm}=0.531 M_1=0.759, M_2=0.899	 $DTW_{single,Itakura}$ S_m=0.737, S_{nm}=0.560 M_1=0.693, M_2=0.858	 $DTW_{single,Itakura}$ S_m=0.693, S_{nm}=0.547 M_1=0.812, M_2=0.929

Table 6: *Different graphs and starting nodes for the* graph-based spreading *dataset.*

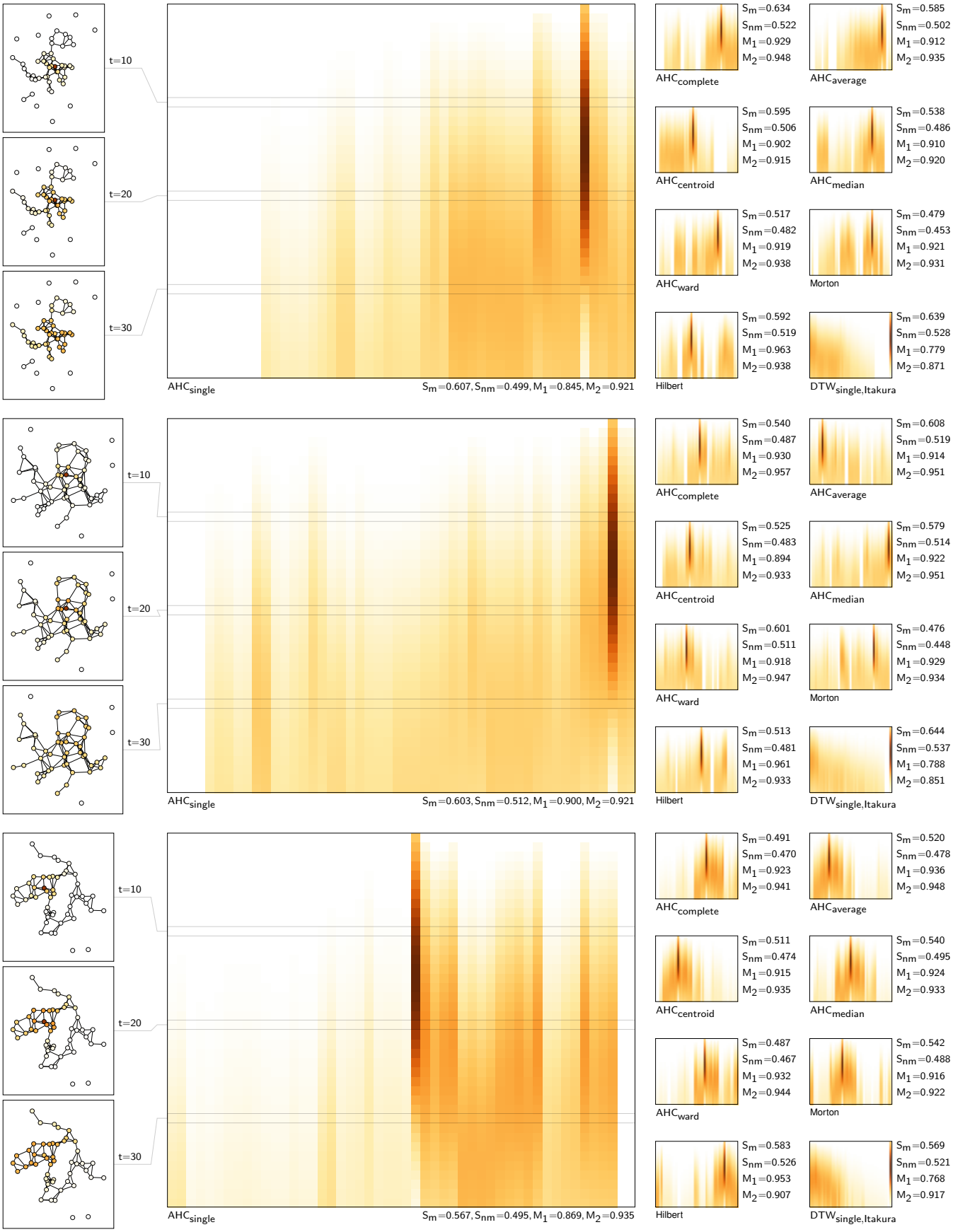


Table 7: *Different graphs and starting nodes for the* graph-based spreading *dataset.*

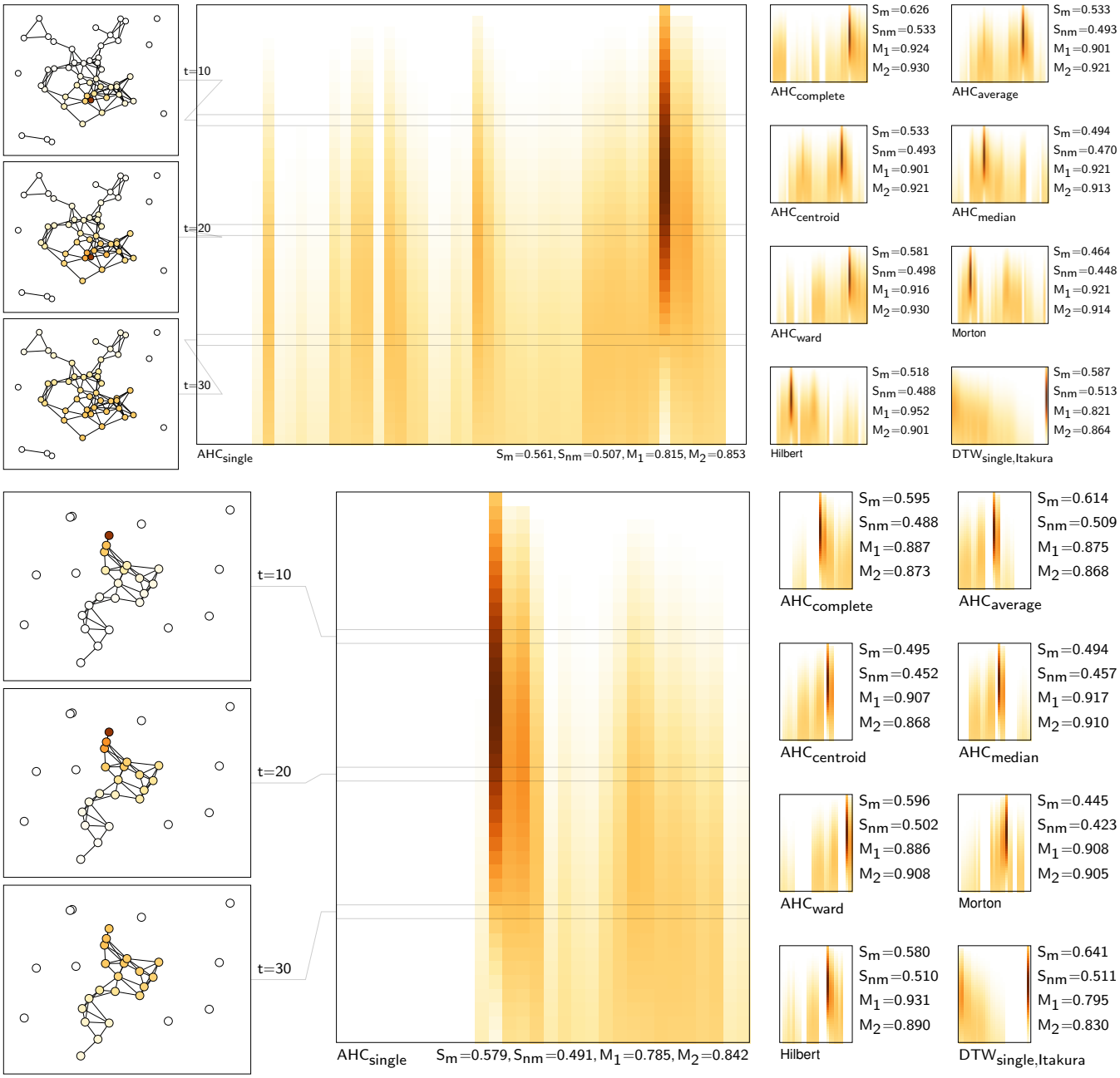


Table 8: *Different graphs and starting nodes for the* graph-based spreading *dataset.*