

TraBiMap: A Transition-Centric Probabilistic Binary Map of Trajectories for Privacy-Aware Analysis

Paul Walther
Technical University of Munich
School of Engineering and Design
Munich, Bavaria, Germany
paul.walther@tum.de

Florian Theimer
LMU Munich
Munich, Bavaria, Germany
florian.theimer@campus.lmu.de

Xuanshu Luo
Technical University of Munich
School of Engineering and Design
Munich, Bavaria, Germany
xuanshu.luo@tum.de

Matthias Schubert
Munich Center for Machine Learning,
LMU Munich
Munich, Bavaria, Germany
schubert@dbi.fim.lmu.de

Martin Werner
Technical University of Munich
School of Engineering and Design
Munich, Bavaria, Germany
martin.werner@tum.de

Abstract

Analyzing collective mobility data unlocks valuable opportunities in domains such as marketing, traffic, urban planning, disaster management, and public health assessment. However, this data is usually stored per individual, resulting in complex mobility pattern extraction and a high risk of privacy threats. To address this, we propose an alternative discrete trajectory storage in grid cell transitions, which (a) decouples user IDs from temporal patterns and (b) obfuscates single users' travel patterns by not explicitly storing the location-ID relation but instead applying Bloom filters as an approximate set membership data structure. We show that our approximate approach improves performance for analytical queries with sufficient utility for mobility analysis tasks like assessment of crowdedness, co-movement analyses, and travel pattern extraction, while posing privacy gains compared to raw data processing. The practical usefulness of the TraBiMap is evaluated on three trajectory datasets *Character*, *Roma*, and *Tdrive*, along the dimensions privacy protection level, service quality, anonymization success, candidate result set size, and computational performance.

CCS Concepts

• **Information systems** → **Location based services**; • **Security and privacy** → **Data anonymization and sanitization**.

Keywords

Mobility Data, Trajectories, Bloom Filter, TraBiMap, Privacy, Utility

ACM Reference Format:

Paul Walther, Florian Theimer, Xuanshu Luo, Matthias Schubert, and Martin Werner. 2026. TraBiMap: A Transition-Centric Probabilistic Binary Map of Trajectories for Privacy-Aware Analysis. 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, 14 pages. <https://doi.org/10.1145/3841645.3842970>



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/10.1145/3841645.3842970>

1 Introduction

With the proliferation of hardware for tracking movements through Global Navigation Satellite Systems (GNSS), the availability of human trajectories has increased substantially. This has transformed mobility analysis, as the precise movement capture capabilities of GNSS services demonstrate superior performance compared to manual travel analysis based on surveys [39]. Such automated systems provide opportunities for reduced error rates, sustained data capture with increased resolutions, and diminished effort requirements for study participants. This creates new research opportunities for social and economic sciences on a more fine-grained basis, e.g., analyses of mobility behavior [15], shopping [36], health [32], and urban mobility [21, 33]. Additionally, context-aware research becomes feasible when additional data from other sources is incorporated [13, 36, 42]. As a result, trajectories became a central data type in spatial data analysis and management. At the same time, publishing and using this data type implies serious challenges. While in "traditional" data sensitive attributes and (quasi-)identifiers usually can be separated, trajectories strongly couple individuals' locations and spatial context [33]. Therefore, even single locations can disclose sensitive individual properties like behavior, lifestyle, and health status [1, 33]. Consequently, raw or slightly processed trajectory data is vulnerable to adversarial, linkage, and re-identification attacks [17, 20, 26, 33]. Regulations like the General Data Protection Regulation (GDPR)¹ in the European Union further reinforce these concerns and limit the available mobility datasets for research.

This puts data management systems under significant tension: while access to large trajectory datasets is required for various applications, the publishing of detailed individual trajectories implies significant privacy risks and becomes increasingly infeasible. A solution is to only publish an aggregated or approximated representation, which supports sufficient utility for specific use cases while reducing single-trajectory exposure. In contrast, existing trajectory data management systems mostly focus on the efficient indexing and storage of individual trajectories and provide limited support for aggregated or approximative storage [2, 5, 23, 53].

Only limited research was conducted on non-exact storages based on a rasterization of the data and sliced approaches [28] and

¹www.gdpr.eu, last accessed 30.04.2026

a fuzzy trajectory storage [40]. Other existing non-exact probabilistic representations of geospatial objects are Global Binary Maps (GloBiMaps) [46], which primarily encode where observations occurred rather than how a trajectory unfolds in space. For trajectory analysis, this has limited utility as interest often lies not in the static trajectory footprints but in transitions between spatial regions.

Therefore, we propose the Trajectory Binary Map (TraBiMap): A transition-centric probabilistic data structure for large-scale trajectory data, initially presented by the authors in an earlier workshop short paper [43]. The TraBiMap discretizes and aggregates trajectories in transitions between adjacent raster cells and stores these aggregates in probabilistic representations. This organization of trajectory data along shared cell transitions enables an efficient analysis of collective mobility while avoiding the explicit storage of single trajectories. Furthermore, the data structure emphasizes a storage in directional patterns essential for tasks such as co-movement analyses, crowdedness estimation, and travel pattern extraction while reducing risks for privacy-threatening re-identification and linkage attacks compared to the raw data publishing. The main contributions of the paper are:

- a thorough description of TraBiMap, a transition-based discretization of trajectories capturing (directional) movement patterns.
- an adaptive aggregation strategy to reduce spatial sparsity in the TraBiMap.
- the evaluation of the TraBiMap on benchmark datasets in scenarios like co-movement analyses and frequent pattern mining in comparison to baseline approaches.

In summary, these contributions exemplify how transition-based data representations can be used to share aggregated movement information and conduct meaningful spatial analyses while limiting the publication of detailed individual trajectories.

2 Preliminaries: Bloom Filters and GloBiMaps

Bloom filters are a basic data structure to store the set property of elements in an efficient but probabilistic way by trading storage size against error probability [8, 12, 41]. The core of a standard Bloom filter is a binary array B of length m_{BF} and a set of k_{BF} independent and evenly distributed hash functions $H_i, i \in \{1, \dots, k_{BF}\}$ mapping stored elements to locations in the bit array. Storing the set property of an element into this Bloom filter is conducted by hashing it with all H_i and setting the corresponding bits in the array B to true. Querying the Bloom filter for the set property of an element works similarly: The element is hashed, but now the locations denoted by the hash functions are only checked. If they are all true, the element was originally stored in the Bloom filter with a high probability; on the contrary, if one of the denoted locations is false, we can say for sure that the element was not stored before. Thus, the Bloom filter prevents false negatives. Based on further considerations of this data structure, we can also denote that the false positive rate p_{FP} is dependent on the length of the Bloom filter m_{BF} , the amount of used hash functions k_{BF} and the number of elements already stored in the filter n_{BF} . Jang et al. additionally explain that Bloom filters can be used to approximate set similarity measures [25].

Based on Bloom filters Werner developed a new data structure for sparse global raster data, the *GloBiMaps* [45, 46]. GloBiMaps

allow for constant-time access of random spatial pixels in a stateless access pattern with a high compression ratio and low error probability [46]. Storing a spatial object o into this data structure consists of four steps: ① rasterize the object into distinct raster cells, ② obtain unique cell identifiers of these raster cells, ③ hash these unique identifiers using k_{BF} hash functions, and finally ④ turn the respective denoted cells in a Bloom filter to true. Querying whether a spatial object is part of the database follows the same path and checks in the Bloom filter whether the raster cells spanned by the object are stored in the Bloom filter. The disadvantage of this approach is that only binary data can be stored, which introduces errors: The rasterization error may result in two distinct spatial objects having the same raster footprint and, therefore, a similar footprint in the GloBiMaps. Furthermore, the false positive error of the Bloom filter accumulates for large objects that comprise many raster cells.

3 Related Work

3.1 Trajectory Representations and Aggregations

In general, mobility or object movements can be represented by spatiotemporal trajectories, which means paths through space and time, respectively, a record of the evolution of a position [35]. The formal definition of a trajectory T comprises a set of points $\mathbf{p}_i$ in $\mathbb{R}^n$ with corresponding time-stamps t_i which belong to a trajectory with a specific ID identifying the user instance [30]:

$$T = \{ID, (\mathbf{p}_1, t_1), \dots, (\mathbf{p}_i, t_i), \dots, (\mathbf{p}_n, t_n)\}. \quad (1)$$

Existing trajectory storage systems, e.g., MobilityDB [53], SECONDO [23] and Big Data Systems like HadoopTrajectory [5], and Summit [2] usually store trajectories as such ordered sets of points, per ID , meaning either per-user or per-object. This allows for exact analyses as the movement is captured in full fidelity and detail, and supports expressive querying via spatiotemporal indexes. A disadvantage of this ID -centric exact storage is that aggregate queries need to traverse all data objects: While well-suited to workloads that require access to single trajectories or movement histories, a global evaluation on a location level is usually not well supported. In addition, the required storage space and query effort scales linearly with the dataset size and trajectory length. Further, ID -centricity results in potentially high privacy threats for the single user and, therefore, requires heavy anonymization for data publishing.

3.1.1 Aggregation. Alternatively, trajectories can be stored aggregated to improve scalability, reduce data volume, and simplify spatial relations. Therefore, they can be discretized by reducing the cardinality of trajectory points. Given a continuous space D to be rasterized into a hyper-grid G using cells g of n dimensions, i.e., $G = \bigcup_j \{g_j\}$, a trajectory can be represented by the cells g in which there is a point $\mathbf{p}_i \in T$. Then the rasterized trajectory $\mathcal{T}$ is a subset of G :

$$\mathcal{T} = \{g \in G \mid \exists \mathbf{p}_i \in T \wedge \mathbf{p}_i \in g\}. \quad (2)$$

The cells g can be regular (e.g., squares, triangles, and hexagons in 2D; cubes in 3D) or irregular (Voronoi cells, quadtree cells). More sophisticated approximations smooth out trajectories, making rasterization independent of the resolution of captured trajectory points.

Therefore, depending on the discretization procedure, the selection of these cells may vary (e.g., Bresenham Algorithm [11] and Double Step Incremental Generation [48]). To account for the time dimension of the trajectory, time stamps can either be stored as a cell property in a tuple (g_i, t_i) or the trajectory can be rasterized temporally as well, introducing one more grid dimension.

Anyhow, these discretized representations typically encode *cell occupancies* of trajectories, meaning they capture where trajectories or users were present and are therefore location-centric, treating each cell mostly independently. Consequently, directional movement information is not stored as a high-level element of the data model, but is only implicit in the locations. This limits the capabilities for movement-centric analyses.

3.1.2 Probabilistic Spatial Representations. To further summarize and aggregate spatiotemporal trajectories, methods like spatial count matrices, grid-based summaries [38], and approximate representations [31] have been proposed, which reduce the storage requirement while allowing for (probabilistic) query response. As they trade storage and computation for exactness, they are especially suited to large-scale geospatial analyses. A specific solution for this use case is the previously described GloBiMaps (compare Section 2). They allow constant-time access with high compression ratios by using Bloom filter-based storage [46]. Nevertheless, all these approaches focus on location storage, not on movement or spatial relations between the elements in the data representation.

3.2 Movement and Co-Movement Analysis

One primary goal of trajectory analysis is the collective movement analysis, meaning extracting spatial and temporal patterns apparent in datasets [33]. Traditionally, this is conducted either through frequent pattern extraction, origin-destination (OD) analyses, or trajectory clustering.

For *frequent pattern analysis*, a basic element of many of these methods is a pairwise comparison of the full trajectory or trajectory segments based on similarity measures. The typically used measures like Fréchet [19] and Hausdorff distance [24] or Dynamic Time Warping (DTW) [14] are thereby expensive to compute [44] and require the full trajectory traces. Alternatively, task-specific aggregations such as *origin-destination matrices* or flow graphs are comparably coarse and need to be explicitly generated for specific analyses requiring a non-flexible recalculation or a new query for every task. For general data publishing, they are therefore not well-suited [7]. Other coarse methods rely on *trajectory clustering* [6]. The clustering objective can thereby be manifold and therefore does not necessarily require the trajectories at full spatial resolution. For example, only metadata such as trajectory lengths, start points, and total needed times might be sufficient to cluster trajectories into task-specific, meaningful groups.

3.3 Privacy-Aware Trajectory Publishing

While the exact trajectory storage allows for the most possibilities to analyze the trajectory datasets, this representation is generally not privacy-aware and allows for an easy extraction of sensitive attributes or a re-identification of the users, and the social relationship inference, e.g., with adversarial attacks [20, 34].

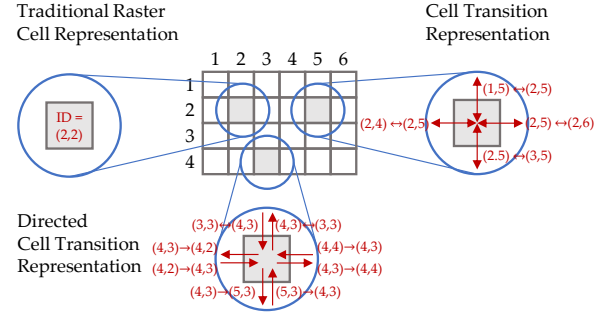


Figure 1: The comparison between the traditional cell-based and the proposed transition-based raster representation.

Therefore, three groups of trajectory publishing techniques have been proposed, which allow for certain privacy guarantees. These include *differential privacy* frameworks for trajectories, which add noise to obfuscate the original data [16, 22], *statistical privacy guarantees* such as k-anonymity [1] or *generative and synthetic approaches*, not publishing the real dataset but instead generating a synthetic dataset, which mimics the original data's distribution without publishing single user's actual movements [3, 7, 27, 52]. Those techniques cover approaches for generalization, inhibition, and perturbation [20].

Still, in practice, many published datasets do not provide explicit privacy guarantees; instead, they generalize and aggregate the data to a level where practical privacy is ensured. For example, the largest widely used human mobility datasets are published at very low spatial and temporal resolution [49] or are limited to only a few users who disclose their privacy [51]. This also shows that there are currently no practical ways to publish high-resolution trajectory datasets in a fully privacy-preserving manner.

4 TraBiMap: A Transition-Centric Representation for Mobility Data

In this paper, we propose the Trajectory Binary Map (TraBiMap) to store mobility data transition-centric, still allowing for a fast, high utility, and privacy-aware evaluation. We use a (directed) cell transition-based discretization approach for trajectory data (compare Section 4.1) and apply a Bloom filter-based storage of these transitions for a simplified movement analysis and added obfuscation (compare Section 4.2 and following).

4.1 Trajectory Discretization by Transition

To get the transition representation of a trajectory T , we first choose an appropriate grid G consisting of grid cells g_k , as with every classical rasterization-based discretization. Depending on the chosen grid, cells have different numbers of neighbors, denoted by the connectivity of a cell. For rectangular raster cells with 4-connectivity, each cell g_k has four adjacent cells and therefore four associated cell transitions. These cell transitions can be undirected or directional, distinguishing the transition from g_k to g_l from that from g_l to g_k (compare Figure 1). Based on these general considerations, we can then get a directed transition-based representation $\mathcal{T}^{\rightarrow}$ of a given

trajectory T :

$$\mathcal{T}^{\rightarrow} = \{(g_k, g_l) \mid \exists p_i, p_j \in T, \text{ s.t. } p_i \in g_k \wedge p_j \in g_l; \\ g_k \text{ and } g_l \text{ adjacent}; \\ j = i + 1\}. \quad (3)$$

and its undirected version $\mathcal{T}^{\leftrightarrow}$:

$$\mathcal{T}^{\leftrightarrow} = \{(g_k, g_l) \mid (g_k, g_l) \in \mathcal{T}^{\rightarrow} \vee (g_l, g_k) \in \mathcal{T}^{\rightarrow}\}. \quad (4)$$

Additionally each of these transitions (undirected $\{g_k, g_l\}$ or directed (g_k, g_l)) might include an ordering criterion t making them tuples: $(\{g_k, g_l\}, t)$ or $((g_k, g_l), t)$. Thus, a continuous trajectory T being traditionally discretized with n grid cells, that means partitions of space, and time tuples is now instead discretized by a set of $n - 1$ intersections with lower-dimensional spatial boundaries: hyperplanes representing cell borders (i.e., planes in 3D and lines in 2D). This shifts the discretization paradigm from volumetric spatial partitioning to a sequence of boundary-crossing events.

4.2 The TraBiMap Architecture

Although the intersection-based discretization of trajectories reduces spatial cardinality, trajectories are still stored as individual data objects. Consequently, the temporal dimension and the user ID remain unchanged compared to the continuous case. With our approach, we aim to further aggregate these trajectory properties, to enable a more efficient analysis and querying of the trajectories, and additionally reduce privacy threats by avoiding the publishing of continuous individual movement patterns.

Therefore, we store trajectory information collectively rather than individually. In traditional storage, trajectory points are aggregated and indexed per user or trajectory ID . In contrast, we reorganize the data along spatial dimensions: Each user or trajectory still retains an ID , but the ID is stored in the spatial locations or regions the trajectory passes through, effectively inverting the traditional data structure. This structural pivot is illustrated in Figure 2. Formally, this leads to a set of tuples of user ID s and times t

Trajectory Storage

Traditional		
$uuid_1$		
$uuid_2$		
Spatial Raster		
(1,2)		
(1,3)		
(2,1)		
(2,2)		
(2,3)		
Generalized Spatial		
(1,2) = {	$uuid_2$	
(1,3) = {	$uuid_2$	
(2,1) = {	$uuid_1$	
(2,3) = {	$uuid_1$	
(2,2) = {	$uuid_1, uuid_2$	
Raster Transition		
(1,2) ↔ (2,2) = {	$uuid_2$	
(1,3) ↔ (2,3) = {	$uuid_2$	
(2,1) ↔ (2,2) = {	$uuid_1$	
(2,1) ↔ (3,1) = {	$uuid_1$	
(2,2) ↔ (2,3) = {	$uuid_1, uuid_2$	

Figure 2: Four ways of storing a trajectory: points per ID , points per raster cell, ID s passing spatial regions, and ID s transitioning between spatial regions.

per undirected transition $S_{kl}^{\leftrightarrow}$ between grid cells g_k and g_l :

$$S_{kl}^{\leftrightarrow} = \{(ID, t) \mid \text{trajectory } ID \text{ crosses boundary } \{g_k, g_l\} \\ \text{at time } t, g_k \text{ and } g_l \text{ adjacent}\} \quad (5)$$

and its directed counterpart $S_{kl}^{\rightarrow}$:

$$S_{kl}^{\rightarrow} = \{(ID, t) \mid \text{trajectory } ID \text{ moves from } g_k \text{ to } g_l \\ \text{at time } t, g_k \text{ and } g_l \text{ adjacent}\} \quad (6)$$

So far, this only increases dataset size, as the ID is now not only stored once per trajectory, but instead once per transition. Additionally, all performed data reorganizations are lossless and can be reversed. Therefore, although the new order can be more suited for a direct comparison of spatial movements across some space boundaries, it neither provides obfuscation nor introduces any privacy enhancement.

The relation between trajectory ID and time t in a tuple does not need to be preserved, resulting in independent sets of ID s and timestamps t per transition. We are therefore looking for a one-way map of the sets of ID s and times t to a smaller space. Ideally, this map would allow a comparison between different transition sets to support spatial analysis while obfuscating the single trajectory or user ID s and timestamps to reduce privacy risks.

Bloom filters are naturally such a one-way map, as the bit patterns of items stored in this approximate membership data structure can collide, and therefore, reversing the mapping is not possible without additional information. Further, these collisions obfuscate the members of the stored sets. Thus, it is always possible to deny the presence of a single member in a set stored in a Bloom filter (denoting it a false positive). Anyway, Bloom filters keep the general membership information and allow for a direct comparison with other Bloom filters to obtain the similarity of the stored sets without revealing single set members: For an approximate comparison of sets, simple bitwise operations are available, and also union and intersection operators exist.

To integrate these Bloom filters into the transition-based discretization of trajectories and create the Trajectory Binary Map (TraBiMap), each cell transition gets assigned its own Bloom filter. We can then define the mappings

$$S_{kl}^{\leftrightarrow} \mapsto B_{kl}^{\leftrightarrow} \text{ and } S_{kl}^{\rightarrow} \mapsto B_{kl}^{\rightarrow} \quad (7)$$

where $B_{kl}^{\leftrightarrow}, B_{kl}^{\rightarrow} \in \{0, 1\}^m$ are bit vector Bloom filter representations of the respective transition. The Bloom filter then holds either the set of ID s of each transition or each tuple of ID and time t .

The directed or undirected TraBiMap $\mathcal{B}$ is then the set of all Bloom filter representations of all directed or undirected transitions, respectively:

$$\mathcal{B}^{\leftrightarrow} = \{B_{kl}^{\leftrightarrow}\} \text{ and } \mathcal{B}^{\rightarrow} = \{B_{kl}^{\rightarrow}\}. \quad (8)$$

For the subsequent theoretical considerations, we do not further differentiate between directed and undirected TraBiMap and generalize a TraBiMap $\mathcal{B} = \{B_{kl}\}$ as assumptions hold for both. Similarly, the evaluation is only conducted for the undirected version.

While this, on the one hand, removes the explicit storage of every ID in every transition as described before, it has the additional advantage that the Bloom filter has a fixed length, resulting in a fixed storage footprint, independent of the amount of stored elements. Our approach is thereby fundamentally different from other spatial Bloom filter data structures, e.g., the GloBiMap [46], which use one big filter to store a set of all relevant locations, while we store a set of entities in each location. Moreover, while the transition sets are defined over (ID, t) tuples for generality, the

remainder of this paper and all experiments consider the purely spatial variant that stores only the *IDs*. The temporal extension additionally encoding the timestamp *t* is left to future work.

4.3 Similarity in the TraBiMap Representation

Meanwhile, these bit vectors still contain information about the set similarity. As such, it is expected that the similarity of two transitions can be obtained from their Bloom filter representations. Therefore, we denote for two bit vectors B_{kl} and B_{op} with following primitives $|B| = \sum_{i=1}^{m_{BF}} B[i]$ (number of set bits), $|B_{kl} \wedge B_{op}|$ (number of shared bits, bitwise AND), and $|B_{kl} \vee B_{op}|$ (number of bits set in at least one filter, bitwise OR). These primitives induce the distance and similarity measures used throughout this work: the Hamming distance d_H , and bitwise Jaccard similarity $\bar{J}_B$. On *explicit* trajectory-ID sets, we use the Dice coefficient sim_{Dice} as the ground-truth similarity and as the traditional, non-private baseline an analyst would compute on the raw data: it is the conventional, interpretable set-overlap measure and is well suited to co-movement queries.

$$d_H(B_{kl}, B_{op}) = |B_{kl} \oplus B_{op}| = \sum_{i=1}^{m_{BF}} \mathbf{1}[B_{kl}[i] \neq B_{op}[i]],$$

$$\bar{J}_B(B_{kl}, B_{op}) = \frac{|B_{kl} \wedge B_{op}|}{|B_{kl} \vee B_{op}|}, \quad \text{sim}_{\text{Dice}}(S_{kl}, S_{op}) = \frac{2|S_{kl} \cap S_{op}|}{|S_{kl}| + |S_{op}|} \quad (9)$$

As Dice and Jaccard are strictly monotone transformations of one another ($\text{sim}_{\text{Dice}} = \frac{2J}{1+J}$), they induce identical rankings, so the choice between them affects only absolute thresholds, not rank-based utility. On the *Bloom filter* representation, we compare transitions with the bitwise Jaccard similarity $\bar{J}_B$ as we state that it is a good approximation of the true Jaccard similarity.

LEMMA 4.1. *For fixed set sizes, the expected bitwise Jaccard similarity $\bar{J}_B$ of Bloom filters of transitions is monotonically increasing in the intersection size of the explicit sets of trajectory IDs, and asymptotically approximates their Jaccard similarity J for small loads.*

We show the proof for this Lemma in Appendix A. A similar proof can be made for the bitwise Dice similarity. Thus, the bitwise Jaccard similarity of the TraBiMap Bloom filters preserves the ordering of, and asymptotically approximates, the true set Jaccard. Further, its normalization by the union of set bits makes it far less sensitive to filter load than the raw Hamming distance: it does not conflate dissimilarity with the popularity (bit density) of a transition, yielding thresholds that transfer across transitions and datasets. We nonetheless retain the Hamming distance $d_H(B_{kl}, B_{op})$ for large-scale retrieval and clustering, where its single-pass XOR/pop-count is markedly cheaper than the two pop-counts required by $\bar{J}_B$ and where, as it does not influence the privacy properties, the loss of fill-invariance is an acceptable trade-off for speed. Summarized, the similarity comparisons between TraBiMap transition representations remain meaningful even though the original sets are not stored explicitly. Still, the bias of the similarity estimator scales linearly in $\frac{k_{BF}}{m_{BF}}$ due to the increasing amount of bit overlaps, resulting in higher utility for similarity queries for small k_{BF} and large m_{BF} and reducing exactness for small filters.

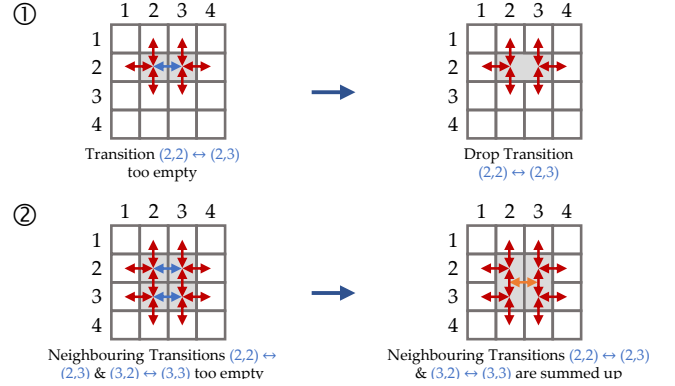


Figure 3: The Adaptive TraBiMap either ① drops transitions or ② merges transitions to achieve fuller Bloom filters.

4.4 Adapting the TraBiMap

In real-world settings, trajectories are not distributed evenly, as mobility is heavily influenced by environmental factors and infrastructure. Therefore, we propose an extension to our previously proposed approach, the Adaptive TraBiMap, which improves usability for these data settings: if some transitions are traversed by only a single trajectory while others are frequently traversed, the resulting Bloom filter fills to different extents. In an extreme case, only a single *ID* is stored in a Bloom filter of one transition, while all *IDs* pass another transition. Consequently, the informativeness of similarity queries is limited for both the crowded and sparse, and the revealed bit-pattern footprints of single users pose a privacy threat in sparse transitions. Therefore, we propose an adaptive conflation of transitions in less crowded regions, including two approaches: ① Dropping of transitions and ② merging of transitions (compare Figure 3). The dropping of transitions is thereby very easy: If a Bloom filter is too sparse to fulfill anonymity constraints, we just drop the respective transition. Alternatively, we might not drop Bloom filters but merge with neighboring ones. This keeps the movement information and just decreases the resolution locally. Merging can thereby happen by a simple binary OR without creating additional artifacts. For a given value of at least *k* users per transition, the adaptive TraBiMap algorithm is given in Algorithm 1.

4.5 Privacy and Limitations

The TraBiMap by itself, provides no formal privacy guarantee. The observed privacy is mainly an obfuscation and aggregation rather than rigorous cryptographic or differential privacy. Still, the TraBiMap removes the explicit user identifiers *ID* as the Bloom filters only store overlapping bit representations (*inhibition*). Further, the *aggregation* of several trajectories and their bit patterns in transitions obfuscates individual contributions, and the false positive rate of the Bloom filters introduce uncertainty, resulting in plausible deniability of a user's attendance at a location (*perturbation*). The adaptive TraBiMap further deletes cell transitions, which can be easily used to infer bit patterns of single users.

With that, the TraBiMap representations combine generalization, inhibition, and perturbation approaches, which are core components of many privacy-improving techniques.

4.5.1 Adversary. A typical adversary to the proposed data structure would have access to the set of published Bloom filter representations $\mathcal{B}$, including the underlying grid. The original dataset, as well as the set of ID values stored in it, is not known to the adversary. Whether the construction algorithm, including the hash functions, is known to the adversary depends on the actual implementation environment, but it is not considered relevant for subsequent discussion, as generally no cryptographic hashes are used; knowledge of the hash functions yields little additional advantage. The goals of the adversary are summarized in Table 1.

Table 1: Goals of an adversary attacking a TraBiMap.

	Goal	Explanation
Ⓐ	Membership inference	Determine whether a candidate element $x = (ID, t)$ is contained in a transition S_{kl} .
Ⓑ	Reconstruction	Recover a subset of the original elements inserted into a transition Bloom filter.
Ⓒ	Linkability	Determine whether two trajectories or users contribute to similar transition filters B_{kl} .
Ⓓ	Cardinality inference	Estimate the number of elements n inserted into a transition filter B_{kl} .
Ⓔ	Similarity inference	Estimate overlap between transition sets from the similarity of their Bloom filters.

4.5.2 Limitations. Regarding the Ⓐ membership inference, Bloom filters always allow a plausible deniability regarding the attendance in a transition. Conversely, the absence of false negatives can still leak information about individuals, as non-membership can always be certified. This means that, for Ⓑ reconstruction, an adversary can also exploit negative information: knowing that a user has not been to a certain location can render many regions infeasible. Generally, the set of candidate relations C for an ID is

$$C(ID) = \{(k, l) : H(ID) \subseteq \text{supp}(B_{kl})\}, \quad (10)$$

with $\text{supp}(B_{kl})$ denoting positions with value one. Observing the same zero bits across multiple filters reduces $C(ID)$, as the constraints aggregate, what can substantially reduce the reachable transitions for an ID even without exact identification, just by knowing of its absence in some transitions.

Combined with additional knowledge, the structure also allows for Ⓒ linkability of trajectories across transitions by matching bit patterns of known trajectory segments together with negative evidence (zero bits) to other transitions to reduce uncertainty in the representations.

Further, the Bloom filter always leaks general structural information such as Ⓓ cardinality. As such, the bit density reveals how many users traveled a certain transition: $n \approx -\frac{m_{BF}}{k_{BF}} \ln(1 - \rho)$ with ρ being the bit density of a filter $\rho \approx \frac{|B|}{m_{BF}}$. Whereas the Ⓔ similarity needs to be leaked to keep utility for the desired tasks (compare Section 4.3 and 4.7).

4.6 Privacy-Utility Trade-Off and Parameter Selection

Generally, the privacy and efficiency of the TraBiMap are strongly related to the choice of parameters. On the one hand, the *spatial grid parameters*, mainly the cell size g_{size} , which, together with the dataset extent, also define the grid size. Deciding on the perfect

grid and cell size is thereby strongly application-dependent. As Seidl et al. explains, for classical mobility patterns in cities, e.g., a resolution of more than 250 m might reserve anonymity best, but reduce spatial patterns in the data to analyze [37]. This holds for TraBiMap representations.

On the other hand, the *Bloom filter parameters* can be defined. Main parameters are the filter's length m_{BF} and the number of hash functions used k_{BF} . Together with the number of elements stored in the Bloom filter n_{BF} , this defines its FP-rate p_{FP} and, therefore, the privacy properties of the filter. A high filter load (large $k_{BF}n_{BF}/m_{BF}$) results in fewer zero bits, less informativeness, and greater filter obfuscation due to overlapping hash footprints.

Meanwhile, as shown in Section 4.3, the exactness of similarity queries on TraBiMap data structures is primarily governed by the ratio of the Bloom filter parameters k_{BF}/m_{BF} , which creates a parameter-dependent trade-off between the utility for similarity queries, the representation size, and potential obfuscations supporting privacy. For simplicity, we assume that all Bloom filters in the data structure share the same basic properties. Therefore, for choosing them, the transitions with the most surpassing trajectory or user IDs s_{max} and the fewest surpasses s_{min} define the choice of parameters. Thereby, s_{max} influences the minimum utility and s_{min} the minimum privacy protection level.

In summary, the TraBiMap architecture provides probabilistic and practical privacy through obfuscation, but does not provide cryptographic privacy: privacy is strengthened but not absolute.

4.7 TraBiMap Construction and Usage

The *construction* of a TraBiMap comprises five steps, as depicted in Figure 4. For a single trajectory $T \in D$, its representation can be obtained by deriving its corresponding set of cell transitions $\mathcal{T}^{\leftrightarrow}$ (compare step ① in Figure 4). Next, the hash footprint of T is obtained by hashing its ID with k_{BF} hash functions and masking these hash functions to the universe $\{0, m_{BF} - 1\}$, where m_{BF} is the length of the Bloom filter (compare step ② and ③ in Figure 4). The hash footprint is then a binary array of length m_{BF} with true values at the locations indicated by the hash functions. Inserting an ID into a transition S is then equivalent to a binary OR with the corresponding Bloom filter B (compare step ④ in Figure 4). Starting from an empty Bloom filter, for the first trajectory, every B for $S \in \mathcal{T}$ gets filled with at most k_{BF} true values. Then, supposing another trajectory T_2 passing through the same transitions S , at most another k_{BF} true values would be added to the same filter.

Using the proposed TraBiMap representation allows for backward-oriented analysis tasks. The *crowdedness* of certain regions can be deduced from the hamming weight of the single Bloom filters in the TraBiMap. For *co-movement relations*, a simple distance measure of the corresponding Bloom filter representations can be sufficient. For that, it can be assumed that statistically, transitions with similar Bloom filters were passed by similar trajectories (compare Section 4.3). *Frequent spatial pattern mining* might be conducted similarly when clustering Bloom filter representations based on a distance measure. The data representation can further be enriched with semantic context information of the adjacent environment without losing its privacy-preserving property. Further, a TraBiMap

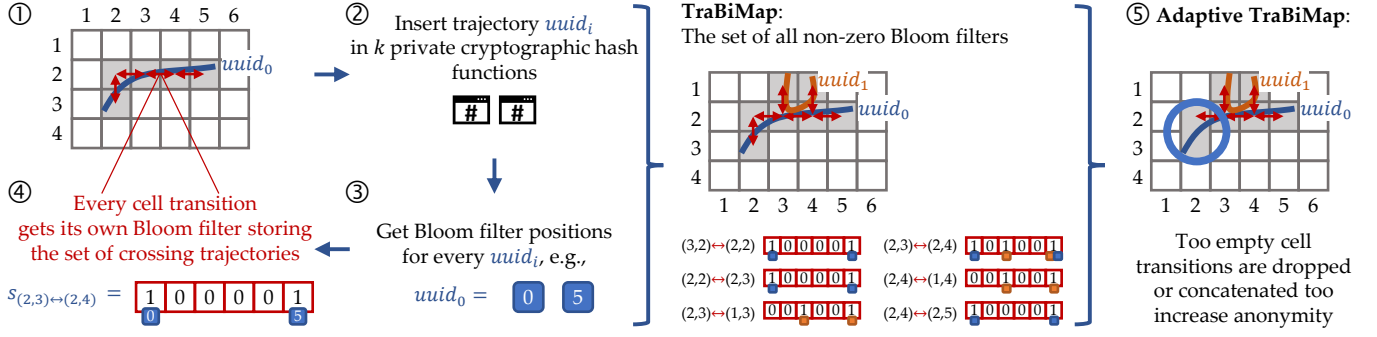


Figure 4: The TraBiMap approach stores IDs in Bloom filters of the corresponding cell transitions (partly adapted from [43])

Table 2: Dataset Overview

Dataset	No. of Trajectories	Length Avg.± Std.	Total Points
Character	2,858	170.5 ± 23.2	487,277
Roma	316	27,740.0 ± 15,602.3	8,765,870
Tdrive	10,141	1,149.3 ± 2,745.6	11,655,104

can be easily *visualized*. Similarities of sets of IDs passing transitions can be shown color-coded and, therefore, give an easy visualization of typical movement patterns. Analysis algorithms for anomaly detection and the semantic classification of single trajectories, as well as forward-oriented tasks such as the prediction of future locations for single users, are not supported by the TraBiMap.

5 Evaluation

We evaluate the TraBiMap along five directions: re-identification risk, service quality, anonymization success, candidate result set size, and computational performance. We conclude with a description of the privacy-utility trade-off of the method.

The study employs three trajectory datasets to assess the TraBiMap (compare Table 2): *Character* [47], a dataset comprised of 2,858 trajectories of a pen tip writing characters, a preprocessed version of the *Roma* trajectory dataset [9, 10] containing GPS traces of taxis in Rome, Italy, and a cleaned version of *Tdrive*, a trajectory dataset from 10,141 taxi trajectories in Beijing, China [50]. For *Character* we use 100×100 grid cells to generate the TraBiMap, and for the two real-world datasets *Roma* and *Tdrive*, resolutions are chosen, such that each grid cell spans approximately 500×500 m in the real-world, resulting in 200×200 (*Roma*) and 340×340 cells (*Tdrive*).

To contextualize the TraBiMap’s privacy-utility tradeoff, we compare against six reference methods using the abbreviations introduced in Table 3. The raw count of IDs in each transition, *TCnt*, and the exact inverted index, *ExIdx*, serve as weak and upper-bound baselines, respectively. *SuppK* implements a standard k -anonymity on the exact ID sets in every transition, which means transitions with fewer than k IDs are removed. *SpatCloak* generalizes SuppK to a coarser grid size (coarsening factor 5). *Geol* [4] perturbs every trajectory point with planar Laplace noise at $\epsilon = 2/\ell$ (where ℓ is the tile edge length), then applies the transition representation on the perturbed trajectories. *DPSyn* [16] builds a count matrix over

all transitions, adds Laplace $(0, 1/\epsilon)$ noise, and samples n synthetic trajectories by random walk on the resulting noisy transition probability matrix before storing those in the transition representation.

5.1 Re-Identification Risk

Our adaptive approach supports transition k -anonymity by pruning and merging transitions that would otherwise violate this privacy constraint (compare Section 4.4). The number of remaining transitions after pruning compared to the baselines is shown in Figure 6. Qualitatively, Figure 5 gives an impression of the TraBiMap representations for varying levels of k , where the transition color corresponds to the filling of the respective transition’s Bloom filter. It is evident that the proportion of removed cell transitions is the lowest for the *Character* dataset, as it is distributed most evenly across all non-empty transitions (compare Figure 6). In contrast, for the real datasets *Roma* and *Tdrive*, some transitions are intersected by individual trajectories only.

Against a single-observation adversary, TraBiMap leaves no trajectories uniquely identifiable on all three datasets, matching the strongest suppression baseline SuppK and clearly improving on the no-privacy reference ExIdx and on SpatCloak’s partial coarsening. Against a full-trajectory adversary that intersects Bloom-filter membership over all visited transitions, however, almost all trajectories can be re-identified for ExIdx, SuppK, and TraBiMap alike, though such an attack is in practice only a confirmation, as it already presupposes knowledge of the full trajectory.

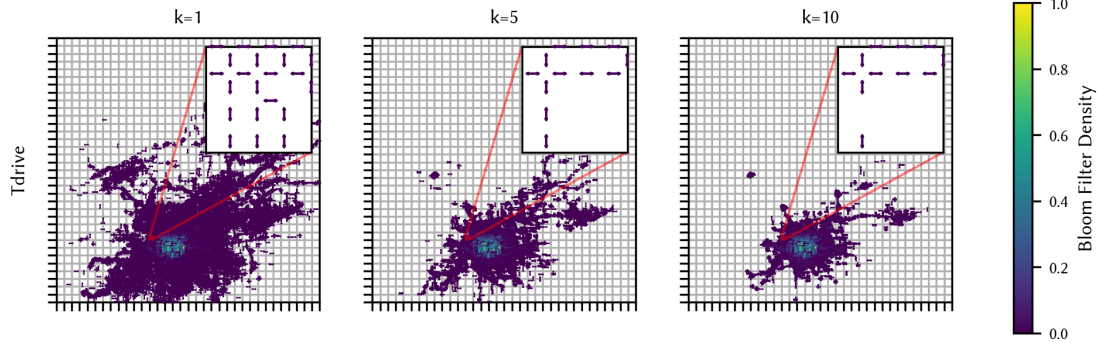
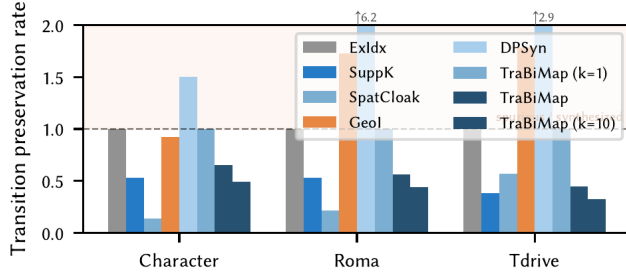
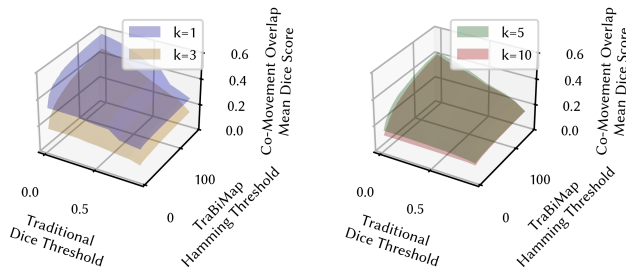
5.2 Service Quality

Figure 7 compares co-movement queries on the undistorted data (thresholded Dice similarity) against the TraBiMap representation (thresholded Hamming distance), reporting for each pairing the best attainable Dice agreement. It is clear that increasing the k -anonymity first decreases the similarity strongly from $k = 1$ to $k = 3$, while a further increase from $k = 5$ to $k = 10$ only slightly reduces the TraBiMap similarity compared to the original data.

To compare all baselines on a common footing, we measure how each method preserves the ranking of neighboring transitions by similarity. For every query transition, we rank all others both by the ground-truth Dice coefficient computed on the exact ID sets (ExIdx) and by the score each method actually publishes, and report the mean Spearman rank correlation ρ between the two orderings

Table 3: Baselines Methods: Transition count, exact inverted index, suppression to k -anonymity, spatial cloaking, geo-indistinguishability, and differential private trajectory synthesis with their abbreviation, privacy method, and procedure.

Name	Abbr.	Privacy	Procedure
Transition Count	TCnt	Inhibition & Generalization	Lower bound: Stores only transition counts. No membership.
Exact Inverted Index	ExIdx	None	Upper bound: Exact ID sets at $k = 1$, ground truth for utility
Suppression k	SuppK	k -anonymity	Classical hard suppression: drops transitions with $k = 5$
Spatial Cloaking	SpatCloak	Generalization	Re-bins trajectories at lower resolution (coarsening factor 5)
Geo-Indistinguishability	GeoI	ϵ -geo-DP	Adds planar Laplace noise to every point [4], then rebuilds index
Differential private trajectory synthesis	DPSyn	ϵ -DP	Noisy first-order Markov synthesizer [16]: samples trajectories from DP transitions

**Figure 5: Visualization of the *Tdrive* dataset in Adaptive TraBiMap representations for different levels of k -anonymity 1, 5 and 10, with zoom-in insets. The false-positive rate of the Bloom filters is fixed at 0.1 for the densest transition.****Figure 6: Transition preservation rate (compared to those in the exact index ExIdx) per method and dataset. TraBiMap is shown at $k \in \{1, 5, 10\}$. The shaded region above 1.0 mark synthesized transitions not in the ground truth.****Figure 7: Service quality of TraBiMap for co-movement queries measured as mean Dice Score between results of traditional thresholded similarity queries and thresholded TraBiMap distance queries pooled from all datasets.**

(compare Figure 8). Two effects must be read together. First, ExIdx and SuppK attain $\rho \approx 1.0$ by construction: their privacy cost is reduced coverage, not ranking distortion. The strawman TCnt, which only exposes transition counts, carries essentially no neighbor information ($\rho \approx 0.00$ on *Character* and slightly negative, -0.17 , on the two real datasets). The fair comparison is therefore among the methods that genuinely perturb their representation. Here TraBiMap clearly dominates on every dataset, reaching $\rho = 0.47$ on *Character*, 0.89 on *Roma*, and 0.69 on *Tdrive*. The non-overlapping bootstrap confidence intervals confirm that TraBiMap's advantage is statistically clear, showing that the Bloom filter representation retains substantially more of the original similarity structure than input perturbation or DP synthesis.

Where Spearman ρ captures the *order* of the retrieved neighbors, retrieval *completeness* is captured by the fraction of the exact ExIdx ground-truth neighbors that each Bloom-filter method recovers (compare Figure 9). All three probabilistic methods recover the majority: on *Character* GeoI reaches full recovery while DPSyn and TraBiMap plateau at 0.91 and 0.87 , on *Roma* the three converge to ≈ 0.8 , and on *Tdrive* DPSyn leads at 0.88 ahead of GeoI and TraBiMap. TraBiMap thus does not maximize raw recall, but instead achieves completeness while keeping its candidate sets smaller than those of the differentially private baselines (compare Figure 10). Combined with its higher ranking fidelity (compare Figure 8), it retrieves the relevant neighbors both more selectively and in a more useful order.

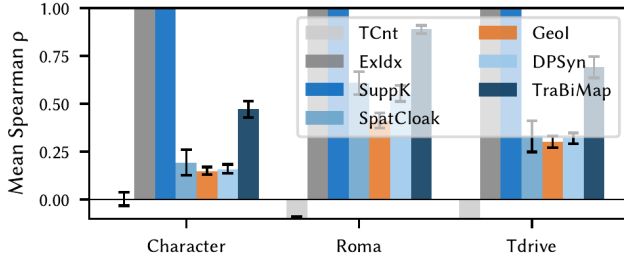


Figure 8: Mean Spearman rank correlation ρ between each method’s per-query neighbor ranking and the ground-truth Dice ranking from the exact inverted index (ExIdx), for all three datasets. Error bars denote bootstrap 95% confidence intervals. ExIdx and SuppK reach $\rho \approx 1$ by construction.

Qualitatively, we can extract travel patterns in the two real-world datasets *Roma* and *Tdrive* by clustering the raw privacy-preserving TraBiMap Filters with Agglomerative Clustering² using the Hamming distance and a preset number of clusters 20 (*Roma*) and 50 (*Tdrive*) (compare Figure 11). Clearly visible are the travel patterns towards the airport (black markers) and along certain road segments (exemplary blue markers), and the segregation of the cities into districts. Especially for the *Tdrive* dataset, a separation in northern and southern parts, as well as in some suburbs, is visible.

5.3 Anonymization Success

The trajectory preservation rate, a widely proposed measure of anonymization success, cannot be directly calculated for a TraBiMap. The data is stored per location rather than by *ID*. Thus, full trajectories can never be retained from a TraBiMap. Probing the unadapted TraBiMap for a single trajectory requires lifting the inhibition measures, i.e., knowing the private *ID* and the hash functions used to build the TraBiMap, probing every Bloom filter with the *ID* to recover all original transitions plus $x = (\text{resolution}^2 - \text{len}(T_{ID})) * p_{FP}$ false positives. This results in a trajectory preservation rate of $\frac{x + \text{len}(T_{ID})}{x}$. With stricter k -anonymity, only entire locations are removed or merged, while counts in surviving transitions remain unchanged (compare Section 4.4 and Figure 6). TraBiMap retains a substantial genuine fraction at $k = 5$ (0.65/0.56/0.45) without exceeding 1.0, and degrades gradually for higher k .

5.4 Candidate Result Set Size

The number of found items for a given bound on the similarity or distance measure is given in Figure 10. On this axis, TraBiMap returns the most compact candidate sets. A more detailed comparison of the real Dice similarity and the Hamming Distance of TraBiMaps is given in the Appendix C.

5.5 Computational Performance

The *construction time* scales with the Bloom filter size and the number of processed data points, but stays within practical bounds for the tested datasets (some seconds to some minutes single threaded).

Figure 12 shows the *query time* between searching for similar transitions with a TraBiMap representation using the Hamming distance and the baseline ExIdx comparing the *ID* sets directly using the Dice Coefficient. The query time is observed to be steady over the number of found items, indicating a CPU-bound operation for both the TraBiMap and real representations. The TraBiMap is generally slightly slower on small datasets, while lying on par with the non-privacy-preserving baseline for *Tdrive*. When extending the comparison to baselines (compare Figure 10), TraBiMap returns the most compact candidate sets.

5.6 Privacy-Utility Trade-Off

To summarise the comparison, Figure 13 positions every method in the privacy–utility plane. A method is preferable the further it lies towards high utility (large ρ) and, simultaneously, low re-identification risk in the top and high coverage in the bottom. Reading the two panels together resolves an apparent paradox. In the top panel, the exact-ID methods ExIdx and SuppK occupy the high- ρ region ($\rho = 1$), and SuppK additionally reaches zero single-observation risk; taken alone, this would suggest they dominate. Their $\rho = 1$, however, is an artifact of publishing exact identifiers rather than a perturbed representation. The right panel shows the corresponding cost in coverage: SuppK retains only 40 to 50% of the ground-truth transitions, while ExIdx attains full coverage only by offering no privacy at all. Among the methods that genuinely perturb their published representation, TraBiMap attains by far the highest ranking quality while holding single-observation risk at 0% and keeping coverage below one (i.e. without fabricating transitions). Geol and DPSyn reach only smaller ρ and inflate their apparent coverage with synthesized transitions, whereas SpatCloak combines the lowest ranking quality with the lowest genuine coverage. TraBiMap thus occupies the favorable corner of the privacy-utility plane among the privacy-preserving representations. Appendix D further explains the dependency privacy and utility based on chosen TraBiMap parameters.

6 Conclusion

The proposed TraBiMap serves as a new data structure for publishing trajectory datasets, enabling a distribution with reduced privacy risks. In this paper, we analyze the approach’s properties theoretically before evaluating it against criteria such as privacy protection and service quality. The evaluation shows that the anonymized TraBiMap representation can be used for tasks such as co-movement queries and travel pattern analyses without a significant processing overhead, while increasing privacy without formal guarantees. Possible future work includes adapting the TraBiMap to non-orthogonal grids (e.g., H3) or to use-case-specific regions, and adapting transition-specific Bloom filter parameter sets. As such, different numbers of hash functions might be used, proportional to the degree of crowding in transitions. While this would reduce the exactness of crowdedness analysis, it would improve anonymity. Furthermore, a full ϵ -differential privacy guarantee might be obtained by applying concepts of Differential Private Bloom filters using the RAPPOR algorithm [18].

²www.scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html, last accessed 12.08.2026

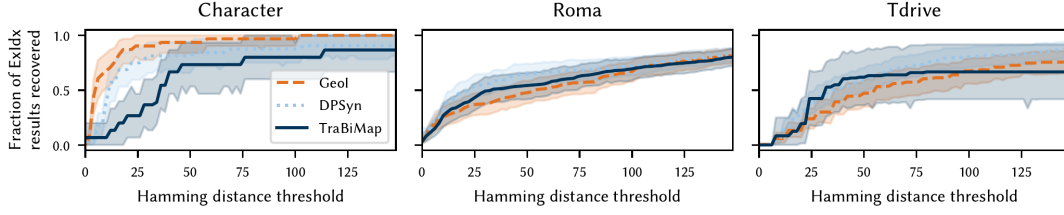


Figure 9: Fraction of the exact ExIdx ground-truth neighbors recovered by each method (GeoI, DPSyn, TraBiMap) as a function of the Hamming-distance threshold, per dataset.

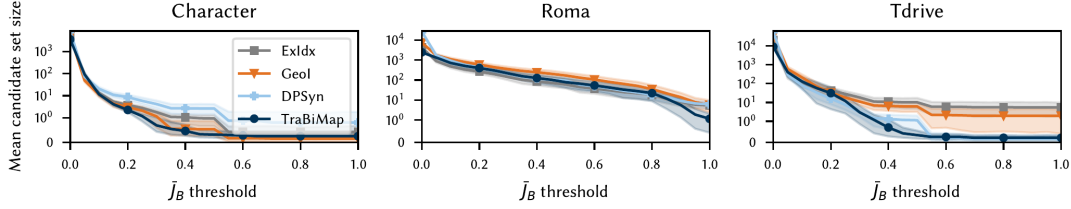


Figure 10: Mean candidate-set size versus bitwise-Jaccard ($\bar{J}_B$) threshold for the Bloom-filter methods (ExIdx, GeoI, DPSyn, TraBiMap), per dataset (symlog y -axis).

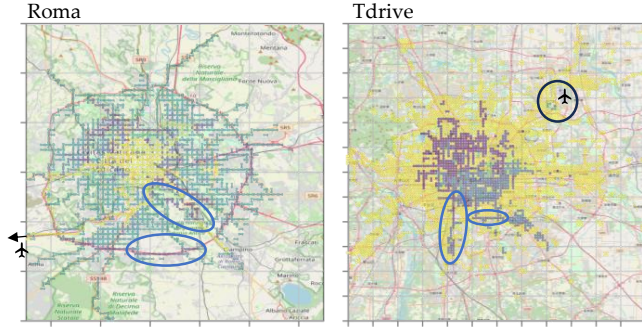


Figure 11: Travel patterns from clustering the TraBiMap Bloom filter representations ($m_{BF} = 8192$, k -anonymity= 10) for Roma and Tdrive. Background map from OSM.

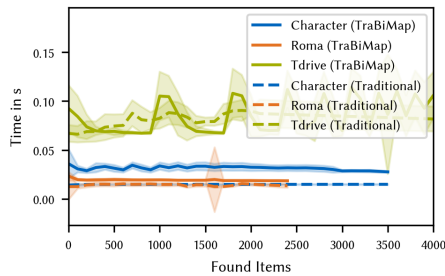


Figure 12: Mean query time for a similarity query in seconds for the TraBiMap vs. ExIdx, depicted with standard deviation intervals. For smoothing the curve, the results are binned over intervals of 50 found items. Conducted single-threaded on AMD EPYC 7713P CPU with max. 3.7GHz.

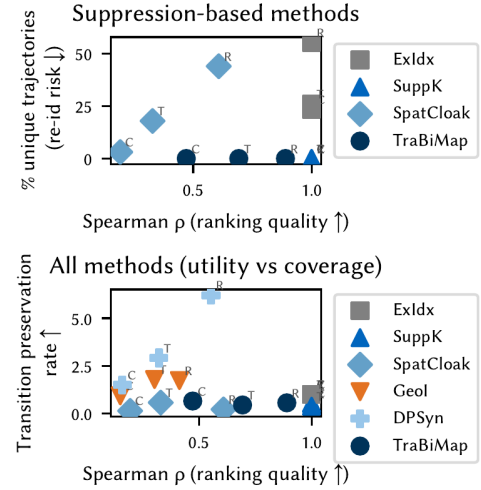


Figure 13: Privacy-utility trade-off across all methods and datasets. Top: ranking quality (Spearman ρ) versus single-observation re-identification risk for the index-based methods. Bottom: ranking quality versus transition preservation rate for all methods. Marker letters denote the dataset (C - Character / R - Roma / T - Tdrive).

Acknowledgments

Generative AI tools (Claude Code, Grammarly) assisted with code development, debugging, and manuscript editing. All AI-generated content was reviewed, verified, and revised by the authors, who take full responsibility.

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 507196470.

References

- [1] Osman Abul, Francesco Bonchi, and Mirco Nanni. 2008. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In *2008 IEEE 24th International Conference on Data Engineering*. IEEE. doi:10.1109/icde.2008.4497446
- [2] Louai Alarabi. 2019. Summit: a scalable system for massive trajectory data management. *SIGSPATIAL Special 10*, 3 (2019), 2–3. doi:10.1145/3307599.3307601
- [3] Hossein Amiri, Will Kohn, Shiyang Ruan, Joon-Seok Kim, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2024. The Patterns of Life Human Mobility Simulation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems (ACM Digital Library)*, Mario A. Nascimento, Li Xiong, Andreas Züfle, Yao-Yi Chiang, Ahmed Eldawy, and Peer Kröger (Eds.). Association for Computing Machinery, New York, NY, USA, 653–656. doi:10.1145/3678717.3691319
- [4] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (ACM Conferences)*, Ahmad-Reza Sadeghi (Ed.). ACM, New York, NY, 901–914. doi:10.1145/2508859.2516735
- [5] Mohamed Bakli, Mahmoud Sakr, and Taysir Hassan A. Soliman. 2019. Hadoop-Trajectory: a Hadoop spatiotemporal data processing extension. *Journal of Geographical Systems 21*, 2 (2019), 211–235. doi:10.1007/s10109-019-00292-4
- [6] Jiang Bian, Dayong Tian, Yuanyan Tang, and Dacheng Tao. 2018. A survey on trajectory clustering analysis. <https://arxiv.org/pdf/1802.06971>
- [7] Vincent Bindschadler and Reza Shokri. 2016. Synthesizing Plausible Privacy-Preserving Location Traces. In *2016 IEEE Symposium on Security and Privacy*. IEEE, Piscataway, NJ, 546–563. doi:10.1109/SP.2016.39
- [8] Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM 13*, 7 (1970), 422–426. doi:10.1145/362686.362692
- [9] Marco Bonola, Lorenzo Bracciale, Pierpaolo Loreti, Raul Amici, Antonello Rabuffi, and Giuseppe Bianchi. 2016. Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers. *Ad Hoc Networks 43* (2016), 43–55. doi:10.1016/j.adhoc.2016.02.002
- [10] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. 2022. CRAWDAD dataset roma/taxi (v. 2014-07-17). doi:10.15783/C7QC7M
- [11] J. E. Bresenham. 1965. Algorithm for computer control of a digital plotter. *IBM Systems Journal 4*, 1 (1965), 25–30. <https://ohiostate.pressbooks.pub/app/uploads/sites/45/2017/09/bresenham.pdf>
- [12] Andrei Broder and Michael Mitzenmacher. 2004. Network Applications of Bloom Filters: A Survey. *Internet Mathematics 1*, 4 (2004), 485–509. doi:10.1080/15427951.2004.10129096
- [13] Vanessa Brum-Bastos, Marcelina Loś, Jed A. Long, Trisalyn Nelson, and Urska Demšar. 2022. Context-aware movement analysis in ecology: a systematic review. *International Journal of Geographical Information Science 36*, 2 (2022), 405–427. doi:10.1080/13658816.2021.1962528
- [14] Kevin Buchin, André Nussner, and Sampson Wong. 2022. Computing Continuous Dynamic Time Warping of Time Series in Polynomial Time. <http://arxiv.org/pdf/2203.04531>
- [15] Pablo Samuel Castro, Daqing Zhang, Chao Chen, Shijian Li, and Gang Pan. 2013. From taxi GPS traces to social and community dynamics. *ACM Comput. Surv. 46*, 2 (2013), 1–34. doi:10.1145/2543581.2543584
- [16] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security (ACM Conferences)*, Ting Yu (Ed.). ACM, New York, NY, 638–649. doi:10.1145/2382196.2382263
- [17] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, Wen Hua, and Xiaofang Zhou. 2017. Exploiting Spatio-Temporal User Behaviors for User Linkage. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (ACM Digital Library)*, Ee-Peng Lim (Ed.). ACM, New York, NY, 517–526. doi:10.1145/3132847.3132898
- [18] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (ACM Digital Library)*, Gail-Joon Ahn (Ed.). ACM, New York, NY, 1054–1067. doi:10.1145/2660267.2660348
- [19] M. Maurice Fréchet. 1906. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo 22*, 1 (1906), 1–72. doi:10.1007/BF03018603
- [20] Patricia Guerra-Balboa, Alex Miranda-Pascual, Thorsten Strufe, Javier Parra-Arnau, and Jordi Forné. 2022. Anonymizing Trajectory Data: Limitations and Opportunities. *The Third AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-22)* 2022 (2022). doi:10.5445/IR/1000148633
- [21] Hanqi Guo, Zuchao Wang, Bowen Yu, Huijing Zhao, and Xiaoru Yuan. 2011. TripVista: Triple Perspective Visual Trajectory Analytics and its application on microscopic traffic data at a road intersection. In *2011 IEEE Pacific Visualization Symposium*. 163–170. doi:10.1109/PACIFICVIS.2011.5742386
- [22] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. 2019. Differentially Private and Utility Preserving Publication of Trajectory Data. *IEEE Transactions on Mobile Computing 18*, 10 (2019), 2315–2329. doi:10.1109/TMC.2018.2874008
- [23] Ralf Hartmut Güting, Thomas Behr, and Christian Düntgen. 2010. SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations. *IEEE Data Eng. Bull. 33*, 2 (2010), 56–63. <http://sites.computer.org/debull/A10june/secondo.pdf>
- [24] Felix Hausdorff. 1914. *Grundzüge der Mengenlehre*. Vol. 7. von Veit.
- [25] Jiyong Jang, David Brumley, and Shobha Venkataraman. 2011. BitShred. In *Proceedings of the 18th ACM conference on Computer and communications security (ACM Conferences)*, Yan Chen (Ed.). ACM, New York, NY, 309–320. doi:10.1145/2046707.2046742
- [26] Fengmei Jin, Wen Hua, Thomas Zhou, Jiajie Xu, Matteo Francia, Maria E. Orłowska, and Xiaofang Zhou. 2022. Trajectory-Based Spatiotemporal Entity Linking. *IEEE Transactions on Knowledge and Data Engineering 34*, 9 (2022), 4499–4513. doi:10.1109/TKDE.2020.3036633
- [27] Jong Wook Kim and Beakcheol Jang. 2022. Deep learning-based privacy-preserving framework for synthetic trajectory generation. *Journal of Network and Computer Applications 206* (2022), 103459. doi:10.1016/j.jnca.2022.103459
- [28] Ioannis Kontopoulos, Iraklis Varlamis, Antonios Makris, and Konstantinos Tserpes. 2024. A spatio-temporal matrix representation for trajectory classification. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems (ACM Digital Library)*, Mario A. Nascimento, Li Xiong, Andreas Züfle, Yao-Yi Chiang, Ahmed Eldawy, and Peer Kröger (Eds.). Association for Computing Machinery, New York, NY, USA, 481–484. doi:10.1145/3678717.3691209
- [29] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86. doi:10.1214/aoms/117729694
- [30] Songyuan Li, Hong Shen, and Yingpeng Sang. 2020. A Survey of Privacy-Preserving Techniques on Trajectory Data. In *Parallel Architectures, Algorithms and Programming*, Hong Shen and Yingpeng Sang (Eds.). Springer eBook Collection, Vol. 1163. Springer Singapore and Imprint Springer, Singapore, 461–476. doi:10.1007/978-981-15-2767-8_41
- [31] Tianyi Li, Ruikai Huang, Lu Chen, Christian S. Jensen, and Torben Bach Pedersen. 2020. Compression of uncertain trajectories in road networks. *Proceedings of the VLDB Endowment 13*, 7 (2020), 1050–1063. doi:10.14778/3384345.3384353
- [32] Zongyu Lin, Shiqing Lyu, Hancheng Cao, Fengli Xu, Yuqiong Wei, Hanan Samet, and Yong Li. 2020. HealthWalks: Sensing Fine-grained Individual Health Condition via Mobility Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4*, 4 (2020), 1–26. doi:10.1145/3432229
- [33] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Taylor Anderson, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, Sanjay Chawla, Reynold Cheng, Panos Chrysanthos, Xiqi Fei, Gabriel Ghinita, Anita Graser, Dimitrios Gunopoulos, Christian S. Jensen, Joon-Seok Kim, Kyoung-Sook Kim, Peer Kröger, John Krumm, Johannes Lauer, Amir Magdy, Mario Nascimento, Siva Ravada, Matthias Renz, Dimitris Sacharidis, Flora Salim, Mohamed Sarwat, Maxime Schoemans, Cyrus Shahabi, Bettina Speckmann, Egemen Tanin, Xu Teng, Yannis Theodoridis, Kristian Torp, Goce Trajcevski, Marc van Kreveld, Carola Wenk, Martin Werner, Raymond Wong, Song Wu, Jianqiu Xu, Moustafa Youssef, Demetris Zeinalipour, Mengxuan Zhang, and Esteban Zimányi. 2024. Mobility Data Science: Perspectives and Challenges. *ACM Transactions on Spatial Algorithms and Systems 10*, 2 (2024), 1–35. doi:10.1145/3652158
- [34] Anna Monreale and Roberto Pellungrini. 2023. A Survey on Privacy in Human Mobility. *Transaction on Data Privacy 16* (2023), 51–82. <https://www.tdp.cat/issues21/tdp.a464a22.pdf>
- [35] Wided Oueslati, Sonia Tahiri, Hela Limam, and Jalel Akaichi. 2023. A systematic review on moving objects' trajectory data and trajectory data warehouse modeling. *Computer Science Review 47* (2023), 100516. doi:10.1016/j.cosrev.2022.100516
- [36] M. C. Popa, L.J.M. Rothkrantz, C. Shan, T. Gritti, and P. Wiggers. 2013. Semantic assessment of shopping behavior using trajectories, shopping related actions, and context information. *Pattern Recognition Letters 34*, 7 (2013), 809–819. doi:10.1016/j.patrec.2012.04.015
- [37] Dara E. Seidl, Piotr Jankowski, and Ming-Hsiang Tsou. 2016. Privacy and spatial pattern preservation in masked GPS trajectory data. *International Journal of Geographical Information Science 30*, 4 (2016), 785–800. doi:10.1080/13658816.2015.1101767
- [38] Giannis Spiliopoulos, Marios Votas, Georgios Grigoropoulos, Konstantina Bereta, and Dimitris Zissis. 2024. Patterns of Life : Global Inventory for maritime mobility patterns. In *Proceedings 27th International Conference on Extending Database Technology (EDBT 2024)*. OpenProceedings.org. doi:10.48786/EDBT.2024.67
- [39] Peter R. Stopher and Stephen P. Greaves. 2007. Household travel surveys: Where are we going? *Transportation Research Part A: Policy and Practice 41*, 5 (2007), 367–381. doi:10.1016/j.tra.2006.09.005
- [40] Goce Trajcevski, Ouri Wolfson, Klaus Hinrichs, and Sam Chamberlain. 2004. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems 29*, 3 (2004), 463–507. doi:10.1145/1016028.1016030
- [41] Paul Walther. 2024. Advancements of Randomized Data Structures for Geospatial Data. In *Proceedings of the Workshops of the EDBT/ICDT 2024 Joint Conference co-located with the EDBT/ICDT 2024 Joint Conference*, Themis Palpanas and H V Jagadish (Eds.). CEUR-WS.org. <https://ceur-ws.org/Vol-3651/PhDW-1.pdf>

- [42] Paul Walther, Fabian Deuser, and Martin Werner. 2024. Multi-Modal Contextualization of Trajectory Data for Advanced Analysis. *Datenbank-Spektrum* 24, 3 (2024), 223–231. doi:10.1007/s13222-024-00484-3
- [43] Paul Walther, Xuanshu Luo, and Martin Werner. 2024. TraBiMap: Reducing Privacy Concerns in Trajectory Analysis with Randomized Data Representations. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Geo-Privacy and Data Utility for Smart Societies (ACM Digital Library)*. Association for Computing Machinery, 31–34. doi:10.1145/3681768.3698496
- [44] Paul Walther, Carla Rieger, and Martin Werner. 2026. Efficient Pruning of Range Queries under Fréchet Distance. In *New Trends in Database and Information Systems. ADBIS 2026*. Springer Nature Switzerland and Imprint Springer, Cham.
- [45] Martin Werner. 2019. GloBiMaps - A Probabilistic Data Structure for In-Memory Processing of Global Raster Datasets. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19)*. Association for Computing Machinery, New York, NY, USA, 3–12. doi:10.1145/3347146.3359086
- [46] Martin Werner. 2021. GloBiMapsAI: An AI-Enhanced Probabilistic Data Structure for Global Raster Datasets. *ACM Transactions on Spatial Algorithms and Systems* 7, 4 (2021), 1–24. doi:10.1145/3453184
- [47] Ben Williams. 2008. Character Trajectories. doi:10.24432/C58G7V
- [48] Xiaolin Wu and Jon G. Rokne. 1987. Double-step incremental generation of lines and circles. *Computer Vision, Graphics, and Image Processing* 37, 3 (1987), 331–344. doi:10.1016/0734-189X(87)90041-7
- [49] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. 2024. YJMob100K: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific data* 11, 1 (2024), 397. doi:10.1038/s41597-024-03237-9
- [50] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM Conferences)*, Divyakant Agrawal (Ed.). ACM, New York, NY, 99–108. doi:10.1145/1869790.1869807
- [51] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.
- [52] Fan Zhou, Xin Liu, Kunpeng Zhang, and Goce Trajcevski. 2021. Toward Discriminating and Synthesizing Motion Traces Using Deep Probabilistic Generative Models. *IEEE transactions on neural networks and learning systems* 32, 6 (2021), 2401–2414. doi:10.1109/TNNLS.2020.3005325
- [53] Esteban Zimányi, Mahmoud Sakr, and Arthur Lesuisse. 2020. MobilityDB: A Mobility Database based on PostgreSQL and PostGIS. *ACM Transactions on Database Systems* 45, 4 (2020), 1–42. doi:10.1145/3406534

A Derivation of the Expected Bloom Filter Similarity

PROOF. We define two sets S_{kl} and S_{op} , e.g. ID s or (ID, t) tuples, associated to different transitions and denote

$$a = |S_{kl}|, \quad b = |S_{op}|, \quad c = |S_{kl} \cap S_{op}| \quad (11)$$

and therefore

$$|S_{kl} \cup S_{op}| = a + b - c. \quad (12)$$

The Jaccard similarity of these two sets is then defined as:

$$J(S_{kl}, S_{op}) = \frac{|S_{kl} \cap S_{op}|}{|S_{kl} \cup S_{op}|} = \frac{c}{a + b - c} \quad (13)$$

The bitwise Jaccard similarity of the associated Bloom filter representations B_{kl} and B_{op} then is according to [25]

$$J_B(B_{kl}, B_{op}) = \frac{|B_{kl} \wedge B_{op}|}{|B_{kl} \vee B_{op}|} \quad (14)$$

If these Bloom filters are of large length m_{BF} and are constructed with k_{BF} sufficiently independent and uniform hash functions, we can describe this by the expected bit-level Jaccard similarity as the ratio concentrates around the ratio of expectations, which we denote $\bar{J}_B$.

$$\bar{J}_B(B_{kl}, B_{op}) = \frac{\mathbb{E}[|B_{kl} \wedge B_{op}|]}{\mathbb{E}[|B_{kl} \vee B_{op}|]} \quad (15)$$

With a random variable $Z_i = \mathbf{1}[B_{kl}[i] = 1 \wedge B_{op}[i] = 1]$ we can describe the bit overlap $\mathbb{E}[|B_{kl} \wedge B_{op}|]$ as

$$\mathbb{E}[|B_{kl} \wedge B_{op}|] = \sum_{i=1}^m \mathbb{E}[Z_i] \quad (16)$$

Further, the probability for a position i in the filter being zero is the probability that a fixed bit remains zero after inserting n elements with k_{BF} hash functions is

$$p_0(n) = \Pr[B[i] = 0] = \left(1 - \frac{1}{m}\right)^{k_{BF}n} \approx e^{-k_{BF}n/m}. \quad (17)$$

Based on the known amount of elements inserted in the Bloom filter, a and b and the inclusion-exclusion principle:

$$\begin{aligned} \Pr[B_{kl}[i] = 1 \wedge B_{op}[i] = 1] &= 1 - p_0(a) - p_0(b) + p_0(a + b - c) \\ &= 1 - e^{-k_{BF}a/m} - e^{-k_{BF}b/m} + e^{-k_{BF}(a+b-c)/m} \end{aligned} \quad (18)$$

And finally, the expected bit overlap:

$$\mathbb{E}[|B_{kl} \wedge B_{op}|] = m \cdot \left(1 - e^{-k_{BF}a/m} - e^{-k_{BF}b/m} + e^{-k_{BF}(a+b-c)/m}\right) \quad (19)$$

Similarly, the probability that a bit is set in at least one of the two filters is

$$\Pr[B_{kl}[r] = 1 \vee B_{op}[r] = 1] = 1 - p_0(a + b - c). \quad (20)$$

and the expectation for this is

$$\mathbb{E}[|B_{kl} \vee B_{op}|] = m(1 - p_0(a + b - c)). \quad (21)$$

This yields the expected bit-level similarity

$$\begin{aligned} \bar{J}_B(B_{kl}, B_{op}) &= \frac{1 - q(a) - q(b) + q(a + b - c)}{1 - q(a + b - c)} \\ &\approx \frac{1 - e^{-k_{BF}a/m} - e^{-k_{BF}b/m} + e^{-k_{BF}(a+b-c)/m}}{1 - e^{-k_{BF}(a+b-c)/m}}. \end{aligned} \quad (22)$$

For low load with $\lambda = \frac{k_{BF}}{m}$ and $\lambda(a + b - c) \ll 1$ we can show with expansion $e^{-x} \approx 1 - x + \frac{x^2}{2}$ that the expected similarity overestimates the true set Jaccard similarity $J(X, Y) = \frac{c}{a+b-c}$ via a non-negative first-order correction:

$$\bar{J}_B(B_{kl}, B_{op}) = J + \frac{k}{m} \frac{2ab - ac - bc}{2(a + b - c)} + O\left(\left(\frac{k_{BF}(a + b - c)}{m}\right)^2\right) \quad (23)$$

It is obvious that the Bloom filter similarity systematically overestimates the true Jaccard similarity in expectation due to the positive bias. Further, for fixed set sizes a and b , the expected Bloom-filter similarity is monotonically increasing in the true intersection size c . Indeed, increasing c decreases the union size $a + b - c$, which increases the dependency between the two Bloom filters and increases the expected number of shared set bits relative to the union of set bits. Consequently, Bloom-filter similarity preserves the qualitative ordering of overlaps in expectation: pairs of sets with larger intersections are expected to obtain larger Bloom-filter similarities. $\square$

B Algorithm for Adapting TraBiMap to k-Anonymity

This outlines the adaptation steps in TraBiMap 1. For every under-

Algorithm 1: Adapt TraBiMap to k -Anonymity

Input : TraBiMap $\mathcal{B} = \{B_s\}$ with Bloom filter length m_{BF} , number of hash functions k_{BF} , desired k for k -anonymity
Output : Adapted TraBiMap $\mathcal{B}$ (under-full transitions merged or dropped)
 Mark all $B_s \in \mathcal{B}$ as *unprocessed*;
foreach $B_s \in \mathcal{B}$ **do**
 if B_s is already processed **then**
 continue;
 $w \leftarrow$ Hamming weight of B_s ;
 $k_s \leftarrow -\frac{m_{BF}}{k_{BF}} \ln\left(1 - \frac{w}{m_{BF}}\right)$;
 // est. # elements in B_s
 if $k_s \geq k$ **then**
 Mark B_s as processed;
 // full enough: keep
 else
 $s' \leftarrow$ next parallel neighbour of s ;
 if s' does not exist **then**
 $s' \leftarrow$ previous parallel neighbour of s ;
 if s' exists **and** $B_{s'}$ is *unprocessed* **then**
 // Lossless merge, performed even if $B_{s'}$ is already k -anon
 $B_{s'} \leftarrow B_s \vee B_{s'}$;
 $B_s \leftarrow B_{s'}$;
 // both s and s' now hold the union
 $k_{merge} \leftarrow -\frac{m_{BF}}{k_{BF}} \ln\left(1 - \frac{w(B_{s'})}{m_{BF}}\right)$;
 if $k_{merge} \geq k$ **then**
 Mark $B_{s'}$ as processed;
 else
 Remove B_s from $\mathcal{B}$;
 // still $< k$: drop s , revisit s' later
 else
 Remove B_s from $\mathcal{B}$;
 // no usable neighbour: drop
 Mark B_s as processed;
return $\mathcal{B}$;

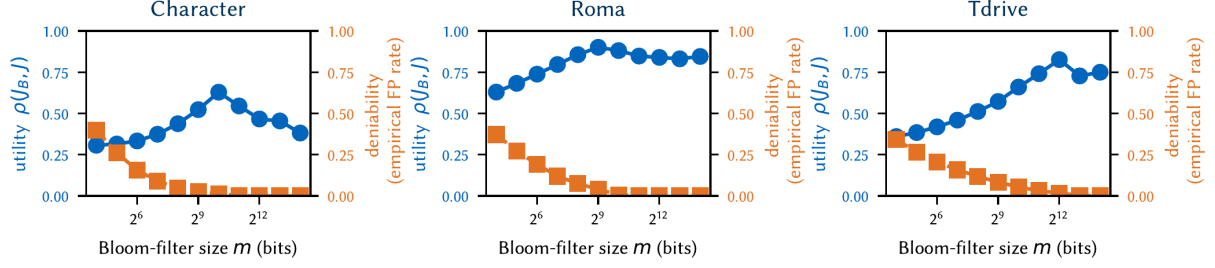


Figure 14: Utility and Deniability in a TraBiMap representation depending on the Bloom filter size.

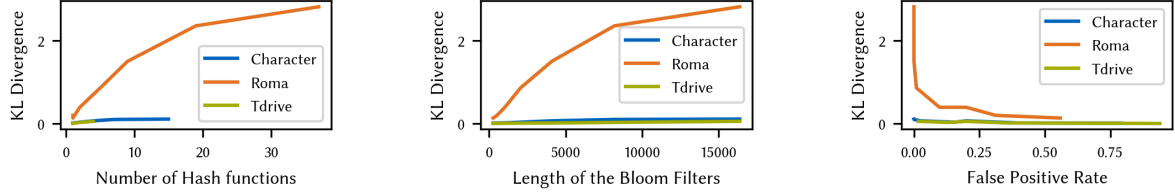


Figure 15: Added information entropy of TraBiMap representations measured with the KL-divergence of the approximate trajectory representations against different Bloom filter parameters, $k = 1$.

full transition, the algorithm searches for a parallel neighbor, first in the forward direction, then in the backward direction. Whenever an unprocessed neighbor exists, the algorithm merges the transition into it with a lossless binary OR. This merge is performed even when the neighbor already satisfies the constraint, because the union can only raise the neighbor’s cardinality; omitting it would leave the sparse transition below k and violate the guarantee. The transition is dropped only when no unprocessed parallel neighbor exists, or when even the merged filter still falls below k (in which case the partial union is retained in the neighbor for a possible later merge).

C Threshold-Based Result Set Size

The result set size based on the chosen threshold is shown in Figure 16: The traditional Dice similarity query is concave, whereas the Hamming distance-based query on the TraBiMap is convex, and with a reasonable threshold, both return a reasonable number of similar transitions. The TraBiMap is thereby less selective.

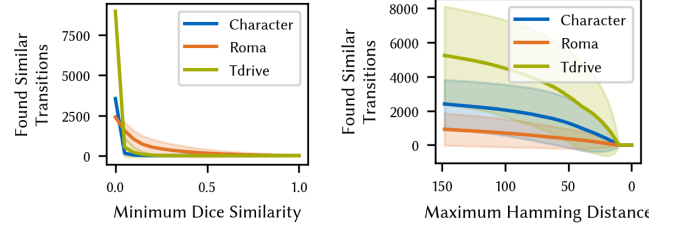
D Parameter Sensitivity: Utility for Different Bloom Filter Parameters

In Figure 14, we analyze the utility of the Bloom filter representation against different Bloom filter sizes.

Additionally, we use the information entropy as a proxy for utility when evaluating against the other BF parameters. This is measured through the Kullback-Leibler (KL) divergence D_{KL} , the difference between two probability distributions P and Q [29]:

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \quad (24)$$

In our case, the information entropy added in one Bloom filter can be described by the difference between the probability of an ID being a valid trajectory and an ID being a false positive. Therefore,



(a) Results set size of similar transitions for minimum Dice similarities in an ExIdx approach without privacy guarantees. (b) Results set size of similar transitions for different maximum Hamming distances in TraBiMap representation.

Figure 16: Result set size of traditional Dice similarity-based approach in comparison to the Hamming Distance of TraBiMap with fixed k -anonymity= 5, depicted with standard deviation intervals.

we calculate the KL divergence by comparing the true membership distribution, expressed as binary indicators of trajectory presence, with the approximate membership probabilities derived from Bloom filter false-positive rates. The KL divergence is computed over the union of all trajectory IDs across transitions, measuring how well the probabilistic Bloom filter representation preserves the original explicit data distribution. A larger divergence indicates that the Bloom filter representation departs more strongly from the explicit membership distribution, i.e., introduces more obfuscating uncertainty. Results depending on the chosen BF parameters are shown in Figure 15.

Received 05 June 2026; accepted 04 August 2026