SoK: Differentially Private Publication of Trajectory Data

Àlex Miranda-Pascual
Universitat Politècnica de Catalunya
Karlsruhe Institute of Technology
Karlsruhe, Germany
alex.miranda.pascual@upc.edu

Patricia Guerra-Balboa
Karlsruhe Institute of Technology
Karlsruhe, Germany
patricia.balboa@kit.edu

Javier Parra-Arnau
Universitat Politècnica de Catalunya
Karlsruhe Institute of Technology
Barcelona, Spain
javier.parra-arnau@kit.edu

Jordi Forné
Universitat Politècnica de Catalunya
Barcelona, Spain
jordi.forne@upc.edu

Thorsten Strufe
Karlsruhe Institute of Technology
Karlsruhe, Germany
thorsten.strufe@kit.edu

ABSTRACT

Trajectory analysis holds many promises, from improvements in traffic management to routing advice or infrastructure development. However, learning users' paths is extremely privacy-invasive. Therefore, there is a necessity to protect trajectories such that we preserve the global properties, useful for analysis, while specific and private information of individuals remains inaccessible. Trajectories, however, are difficult to protect, since they are sequential, highly dimensional, correlated, bound to geophysical restrictions, and easily mapped to semantic points of interest.

This paper aims to establish a systematic framework on protective masking measures for trajectory databases with differentially private (DP) guarantees, including also utility properties, derived from ideas and limitations of existing proposals. To reach this goal, we systematize the utility metrics used throughout the literature, deeply analyze the DP granularity notions, explore and elaborate on the state of the art on privacy-enhancing mechanisms and their problems, and expose the main limitations of DP notions in the context of trajectories.

KEYWORDS

Systematization of knowledge, privacy-preserving data publishing, trajectory privacy, differential privacy, utility metrics

1 INTRODUCTION

Trajectory data mining and analysis have become a relevant branch of study due to their numerous applications [70]. Not only can their processing improve our daily lives, for instance, through navigation and route recommendation, but it also has various institutional data-analytics applications in both the public and private sectors. The ability of personal devices (e.g., wearables, smartphones [7]) and navigation systems to accurately collect, process, and analyze these data, and their ubiquitous availability, amplify this development, which is growing at an unprecedented rate thanks to recent technological advances. Traffic management, urban planning, transportation-system design, routing advice, or homeland security are just a few of the many applications that benefit from trajectory analyses [47].

Although data analyses bear economic and societal good, tensions regarding privacy risks are growing [88, 105]. Protecting data subjects and reducing possible harm inflicted upon them hence gains importance. Consequently, legal frameworks in the European Union and other regions explicitly limit personal data collection, processing, and sharing. The European General Data Protection Regulation (GDPR) indeed requires personal data anonymization as one way to circumvent processing restrictions [39]. Therefore, assuring tight privacy preservation when analyzing location trajectories is also a legal requirement.

Generally speaking, trajectories are sequences of timestamped locations (such as GPS coordinates). These, at first sight, may appear innocuous to users' privacy, but trajectories can reveal exact home locations and even accurate behavioral patterns [98]. They readily give away when and how long a particular individual does what. Exploiting this, one can infer circumstances and trends affecting sensitive aspects of an individual's life, including health status, religious beliefs, social relationships, or sexual preferences [22].

We investigate the possibility of publishing entire trajectory databases with privacy guarantees towards this end. *Statistical disclosure control* (SDC) addresses the attempt to prevent confidential information from being linked to specific individuals when releasing data [59]. Given a raw database, the goal is to publish a sanitized version that reduces disclosure risk while retaining *utility*: the property that statistical analyses yield similar results in both databases. Adapting SDC techniques to protect trajectories of human mobility is no easy task, as we shall describe in the following sections. Well-known metrics from the field, such as k-anonymity [95] or ε-differential privacy (ε-DP) [35], are not immediately applicable to sequential and high-dimensional data sets.

The uniqueness of human traces implies that, with little background knowledge about data subjects (such as their place of residence or work), adversaries can attack seemingly protected data with ease [27, 122]. In this context, research shows that knowing only four spatio-temporal points at low resolution is enough to uniquely identify 95% of the individuals in a given database of large scale [29]. Furthermore, we can recover an original, seemingly sanitized trajectory within an obfuscated area using auxiliary public information, like road maps, speed limits, or simple spatio-temporal correlation models [9, 117]. All this ultimately leads to poor privacy. Even though there exist numerous proposals in the literature,

Proceedings on Privacy Enhancing Technologies 2023(2), 496–516

https://doi.org/10.56553/popets-2023-0065

most come with apparent deficiencies, ranging from vulnerabilities to simple attacks to spoiled utility due to wasting information contained in trajectory data or publishing impossible trajectories.

Finally, numerous applications of trajectory-data analyses involve repeated computations, since they typically monitor certain conditions (e.g., traffic). However, regularly publishing updated versions of a database in a privacy-preserving way makes the challenge even harder. The main reason is that each publication leaks some information about the individuals contained in the database, and it is not simple to ensure that combinations of published private data will not compromise privacy at any moment.

All the mentioned problems raise serious concerns about the current state of the art for trajectory privacy. They create doubts about whether existing technology can effectively guarantee individuals' privacy and if they can strike an acceptable balance between privacy and the utility of data. This situation demands a comprehensive systematization of use cases, limitations, and misconceptions in the field, as well as a standardized classification that facilitates the way for researchers to choose a good privacy metric, develop a mechanism, and measure its utility adequately.

Contributions and Related Works. In this paper, we conduct a comprehensive, systematic state-of-the-art analysis on *privacy-preserving trajectory publication*, the goal of which is to publish a sanitized database of real-world personal trajectories with actual guarantees in both privacy and utility. This area is formally known as *masking* [59] (i.e., creating a modified version D^* of the original database D), in opposition to the generation of synthetic databases, which generates new data that preserves some statistic properties of the original database [59], or the publication of aggregated statistics. This area has yet to be fully explored and surveyed in the literature, especially in contrast to the other scenarios. We specifically focus on DP for the publication of sanitized trajectories. Our review and analysis of privacy technologies includes *an introduction to trajectories* (Section 2), *utility metrics* (Section 3), *risks and attacks* (Section 4), *DP notions* (Section 5), *DP masking mechanisms* (Section 6), and *DP challenges and limitations for trajectory data* (Section 7). More specifically, our contributions towards a systematization of knowledge are the following.

- We systematically analyze how the utility of the sanitized trajectory data can be measured, providing a novel classification of utility metrics, and exploring similarity measures for the scenario at hand.
- We discuss DP adaptations to trajectory data and the implications these may entail by analyzing the most relevant DP granularity notions proposed for trajectory data, including conclusions and use cases for each.
- We propose a novel taxonomy of privacy-protecting technologies. We systematically survey the state of the art and recent advances in the literature, and we discuss and prove mathematically which algorithms are not formally DP.
- We finally discuss and recompile the challenges and limitations of DP (and its different granularity adaptations) as a privacy notion in the context of trajectories.

We succinctly describe the main differences between our work and prior surveys in the field. Primault et al. [92] provide a deep analysis of location-privacy protection mechanisms, including a division of the protection mechanisms into online and offline methods. However, the authors do not cover trajectory privacy extensively since their main focus is on the more general field of location privacy. Note that trajectory data is inherently more complex than simple location data: trajectories are not only comprised of visited locations but also include correlations and connections between them. In consequence, attacks, privacy-protection mechanisms, and limitations are notably different, even though these data types share a close relationship. Fiore et al. [41] offer a thorough overview and classification of attacks on trajectory databases and discuss privacy-preserving mechanisms. However, they cover mostly mechanisms to generate synthetic data and do not study the various privacy and utility metrics available in the literature for trajectory protection, nor the limitations of DP for trajectory data. More recently, Jin et al. [62] conduct a survey with an analysis and empirical evaluation of trajectory-privacy models to quantify their privacy and utility, but do not consider DP mechanisms in depth.

Our work entirely focuses on DP masking mechanisms for private database publication, which the aforementioned surveys do not fully explore. Other works focus on orthogonal topics, such as trajectory anonymization under syntactic notions [90] and location privacy (not comprising trajectories) [61].

2 TRAJECTORIES AND THEIR DATA SETS

Trajectories correspond to a path or trace generated or drawn by a *moving object*, usually referred to as an *individual* or *user* (we will refer as such independently on what they are, e.g., a person walking, or a car carrying various people).

Different types of trajectories exist. *Raw trajectories* consist of an ordered sequence of spatio-temporal points $T = \langle p_1, \ldots, p_m \rangle$ where $|T| := m$ denotes the *length* of T and $p_i = (x_i, y_i, t_i)$ corresponds to the location (x_i, y_i) at timestamp t_i. Trajectories respect the temporal order (i.e., t_{i+1} must happen strictly after t_i), which ensures there are no movements back in time, and no one is in two different locations at once. The term *subtrajectory* usually refers to a subset of a trajectory, including those formed by non-necessarily consecutive locations, while *n-grams* (also called *subsequences*) are subtrajectories formed by n consecutive spatio-temporal points. The *prefixes* of a trajectory $T = \langle p_1, \ldots, p_m \rangle$ are the n-grams ($n \leq m$) starting at p_1, i.e., $\langle p_1, \ldots, p_n \rangle$.

Semantic trajectories are alternative representations where every spatio-temporal point contains additional *semantic meaning*, such as a name and description (e.g., "coffee shop" or "work"), possibly augmented with additional information such as the number of visitors or opening hours. In this latter case, locations are called *point of interest* (POI). More complex trajectories, called *multiple aspect trajectories* [83], additionally consider any possible type of recordable information, like weather variations, transportation mode, or the current heart rate or emotions of individuals. Simplified trajectories have been suggested, such as $T = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$, where time is omitted and only the order of locations is retained [17, 18, 53, 56].

We will refer to the *spatial* and *temporal* aspects as *dimensions* of a trajectory, which are both commonly represented as numerical data. Semantic locations additionally have a *categorical dimension*.

Trajectory databases consist of one or multiple trajectories from individuals, usually over a shared region. We can represent them as collections of rows, where each row contains the data of a single individual:

$$D = \begin{Bmatrix} T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_{m_1}^{(1)} \\ T_2: & p_1^{(2)} & p_2^{(2)} & \cdots & p_{m_2}^{(2)} \\ \vdots & \vdots & \vdots & & \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_{m_r}^{(r)} \end{Bmatrix},$$

where T_i denotes a trajectory belonging to user i. The length of each trajectory is denoted here by m_i and depends on each user. In some contexts, the same user can contribute multiple trajectories to the database. In this latter case, i is just a label of the trajectory and does not necessarily relate to a user.

Differences in structure between such databases exist. Some consist only of trajectories of equal length, and others assume that trajectories are *periodically recorded* (i.e., every trajectory has a spatio-temporal point for every time interval) [8, 37]. Further types include those with irregular recordings, with spatio-temporal points only included when the user is at a relevant location [12].

A particular scenario in trajectory publishing is the *data-stream scenario*, where a flow of information is received and published periodically. Therefore, a *streaming database* can be viewed as a sequence $D = \{S_1, \ldots, S_t, \ldots\}$, where each *update* S_i represents the information corresponding to time i:

$$D = \begin{Bmatrix} & S_1 & S_2 & \cdots & \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_{m_1}^{(1)} \\ T_2: & p_1^{(2)} & p_2^{(2)} & \cdots & p_{m_2}^{(2)} \\ \vdots & \vdots & \vdots & & \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_{m_r}^{(r)} \end{Bmatrix}.$$

The database at time t is denoted $D_t = \{S_1, \ldots, S_t\}$ and called a *stream prefix*. Note that since some databases consist of non-periodically recorded trajectories, "gaps" in this representation are possible, as shown in Figure 1. Hence, T_i may not have a location for time t, and remain empty in row i of S_t.

The structure of trajectory data and databases makes its protection exceptionally difficult. Long trajectories cause problems due to the *curse of dimensionality* [3, 32], and the sparseness and uniqueness of trajectories can aid in re-identification. Another risk factor is the semantic meaning of points since this information can be enough to expose individuals.

A notorious statistical property of trajectory databases is the presence of correlation. Two conceptually different correlations are present in trajectory data: *Correlations between trajectories* refers to the case when multiple user's records are correlated. In families' trajectories, for instance, we are bound to observe high correlations between their corresponding records as they engage in shared activities. Furthermore, an extreme case is regular repetitions of trajectories contributed by the same individual. *Correlations between attributes* refers to the correlation in the data a single user contributes to the database. In the case of trajectories, it refers to the correlations within the spatio-temporal and semantic dimensions. A high-correlation level exists between close timestamps due to the laws of physics, route distribution, or social patterns. It is also termed *autocorrelation* for time series data. We present in Section 7 the implications of correlation in privacy.

3 UTILITY METRICS

Privacy mechanisms aim to balance two conflicting goals: strong privacy and high utility. Typically, these mechanisms introduce obfuscation and remove detail in the data, so improving privacy usually comes with a reduction in utility. Measuring the utility and privacy provided by an algorithm is laborious since trajectory data are very complex, and the presence of semantic values can complicate its study.

In this section, we explore utility metrics to assess protected trajectories and provide a new classification. We identify two major goals a sanitization mechanism can aim to preserve: data and statistics. Here, we understand *data preservation* as how much of the output data correspond to the original one (i.e., remains unaltered after sanitization); and *statistic preservation* as the preservation of specific properties of the database (e.g., numbers of visits to important locations), usually extracted from it with query functions. Our classification thus follows this idea, dividing into *data* and *statistics preservation*. Since assuring data realism is a significant utility condition, we also introduce the orthogonal category of *realism assurance*. The right-most columns of Table 2 show the types of metrics used for mechanisms explored in this work.

Data preservation: These metrics measure utility based on the number or proportion of data that is left unaltered after sanitization, or the extent to which it is changed. Technically speaking, not modifying the database would yield the highest possible utility, obviously at the cost of total privacy loss. We further distinguish two subclasses for data preservation: *total*, which looks at how much data remains exactly the same; and *close*, which instead measures some distance since, typically, perturbing trajectories slightly (e.g., moving locations by a few meters) does not strongly hinder the utility of the mechanism in question.

Location preservation [33] is a good example of total data preservation: one maintains high utility when the protected trajectories include many locations present in the original data and not fake ones. Similarly, some proposals evaluate utility as the number or percentage of suppressed trajectories, subsequences or locations [20, 33], or as the size of the restricted area of the map with perturbation [114].

A popular way [11, 21, 26, 55, 58, 72, 76, 97, 129] to quantify close data preservation is by using *similarity measures*, which output a value representing how different two trajectories are. For example, in mechanisms such that a one-to-one correspondence between the original and sanitized trajectories exists, we can use similarity measures to compute the average values between each pair. We deeply explore existing similarity measures in Appendix A. Traffic management is one exemplary use case that can benefit from this group of metrics.

Other close metrics include *map inference metric*, used in [114], which infers the geometry of the road maps drawn by trajectories between the original and sanitized databases. *Preservation range query*, presented in [26], represents the percentage of obfuscated locations in a data set that remain at a distance no greater than δ from their original counterpart.

Statistics preservation: In contrast with the previous categories, this one does not look at the preservation of the data comprising the database, but at specific extractable information. These statistics are extracted using query functions, and therefore the

relative error query function [1, 11, 17, 18, 31, 116, 128] is frequently employed to study their preservation. Given the query q, it computes the difference between the outputs when using the original database D and the sanitized D' as

$$\text{error}(q) = \frac{|q(D) - q(D')|}{\max\{q(D), b\}},$$

where b is the sanity bound used for extremely selective queries (usually chosen to be equal to 0.1% or 1% of $|D|$).

Since these queries can be defined to extract any information from the database, we find multiple diverse examples in the literature. Some of the most common ones relate to visitor numbers and location popularity. For example, *frequent sequential pattern mining* looks at the k most common subtrajectories in the original and sanitized database, either by seeing if they match over the databases [17, 18, 127] or by comparing the counts of such [11, 17, 114]. Similarly, *count queries* [17, 18, 76, 128] can be utilized to check whether the number of visitors to locations is retained or not. Additionally, some metrics tackle the preservation of *number of trajectories* [76], *most visited locations* [78], *hotspots* [26], *location popularity* [114], *flow density* [11, 114] and the start and end points distribution (*trip error* [114]).

Another popular metric type is *trajectory length preservation* [78, 114, 130]. Three variations have been suggested in the literature, varying in usefulness: preservation of the *total travel distance* (i.e., the sum of the physical length between locations), the trajectory *diameter* (i.e., the maximum physical length between any two of its points), and the total number of points in the trajectory.

These metrics are of special interest for commercial purposes, where specific information on trajectories is needed rather than whole trajectories. For example, vendors may be interested in placing their advertising banners on the busiest streets, and city hall may be interested in the distribution of start and end points to decide where to build parking lots. Note this information can be preserved and extracted from sanitized trajectories, without being similar to the original ones in all other respects.

Realism assurance: Finally, we introduce this category that measures the ability of an algorithm to output realistic values. It is motivated by the fact that some methods produce *geospatial inconsistencies* (i.e., with points in illogical places) or *unreachable* points (i.e., a consecutive pair of locations is unattainable in the given time [33]). Accordingly, reachability is a straightforward guarantee of realism, which can be checked by measuring the distance between consecutive points $(x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1})$ to see if they are indeed reachable, i.e., if $d((x_i, y_i), (x_{i+1}, y_{i+1})) \leq v(t_{i+1} - t_i)$ where v is the maximum velocity of the user. Similarly, the previously-mentioned map inference metric [114] can be used to check for geospatially incoherent points.

Observe that there currently are only a few metrics in the literature that fall into this last category, but we believe that checking or ensuring realism is essential when providing privacy mechanisms. Hence, we introduce this category to demonstrate this notable gap.

Conclusions on Utility Metrics: To sum up, mechanisms should naturally achieve good utility, and one needs to be aware that some metrics are better suited for different use cases. Notably, there is no universal utility metric for all applications, and therefore a single proposal can use multiple ones in its evaluation to widen its scope.

Data-preservation metrics are excellent for scenarios where the whole trajectory is considered, such as traffic management. Total data preservation is usually a stronger statement than its close counterpart; however, it can sometimes provide disproportionately poor values for unsuitable mechanisms. For example, if looking at location preservation, a total-preservation metric will output "no utility" given a mechanism that perturbs the coordinates of all points (such as in some DP mechanisms). In such cases, it may be more suitable to use a close variant instead. Statistics-preservation metrics are convenient for publishing information like popular locations or sequences, but they do not reflect the preservation of the whole structure of trajectories.

Assuring that the database contains realistic values is essential. Beyond reflecting good utility, it may furthermore complicate attacks such as those that aim at reconstructing original trajectories.

4 RISKS AND ATTACKS

Having explored how we can measure the utility of trajectory data, we now discuss possible privacy risks. The main goal of trajectory privacy is to protect against risks and threats when unintended actors get access to the data.

We illustrate the tangible risks associated with a lack of privacy protection in trajectory data in the following examples. The New York City taxi data set, which included around 173 million taxi trips and the corresponding tips [108], was published in 2013. Since then, plenty of attacks on this data, using *background knowledge*, quickly appeared: Tockar [106, 108] used paparazzi photos to link celebrities' identities to the corresponding trip in the data discovering where they went, which establishments they visited, and how much they tipped. Deneau [42] figured out that one could link stops with daily praying time to identify Muslim cab drivers. These examples are excellent representatives of two important privacy risk classes [59], *identity* and *attribute disclosure*. We review them in the context of trajectory data in the following subsections.

Furthermore, *sensitive location disclosure* represents a risk that does not refer to leaking private information relating to users, but rather to locations. Disclosure examples are the discovery of secret Israeli and US army bases through the publication of running trails recorded by Strava through soldiers' mobile apps [51, 57].

To show the privacy risks in human traces, we expose the possible attacks and threats of the literature. The attacks correspond to the major classification of Fung et al. [45], adapted by [62] (adding *group linkage attacks*), with our extension of *reconstruction* and *prediction attacks*. We also provide examples, some of which have previously been extensively surveyed in [41, 62].

4.1 Identity Disclosure

Identity disclosure is the primary risk: It happens when an adversary is able to assign an individual to their corresponding record or records in a database. Such assignation may be possible from the database alone (if it directly contains identifying information) by combining the database with external knowledge or auxiliary data, or by probabilistic inference.

Record linkage attacks (or **identity linkage attacks**) attempt to infer individuals' identities. *Re-identification attacks* are the simplest form of this type [81]. They utilize auxiliary information, i.e.,

information exposed through other means and thus available to the adversary. In particular, *personal context linking attacks* [52, 117] use known information about a victim (e.g., they have been to a coffee shop) to discover their trajectory in the database.

Some record linkage attacks aim to discover uniquely identifiable traits to determine the victim's path. In the case of trajectories, little information suffices to do so [29, 94]. Location probability distributions, mobility preferences and patterns, exposed locations, and physical encounters can each be used to design attack models, which detect the unique traits more successfully [30, 38, 44, 48, 94, 125, 126]. For example, experimentation shows that if the adversary knows the traveled distance, speed, and direction, then up to 95% of users can be uniquely identified [94]. We refer the interested reader to [41] for a comprehensive list of similar attacks.

Membership attacks (a generalization of **table attacks** [45, 62] for non-tabular data) aim to discover whether or not a specific individual is present in the database, regardless of whether their records can be directly identified. For example, if the database shows one trajectory leaving a home location, then an adversary can deduce an inhabitant participated in the database. Learning merely the presence or absence of an individual in a trajectory database can be a direct privacy threat (e.g., consider a database of trajectories with traffic violations). Well-known examples include adaptations of *membership linkage attacks* and *membership inference attacks* in trajectory data [93, 101].

4.2 Attribute Disclosure

The second risk is attribute disclosure: An adversary learns additional information about the previously unknown individual without necessarily identifying their exact record in the database. In trajectory data, this includes the whereabouts and temporal information (e.g., when no one is at home). The disclosure of the *user's spatial* and *temporal information* [117] is sensitive itself but can also be indirectly damaging, since it may be associated with semantic knowledge and values. Presence at a hospital for extended amounts of time allows adversaries to infer a user's health status; while being at a place and time where a specific protest is happening may leak information about a user's political opinions.

In **attribute linkage attacks**, adversaries aim to learn attributes by relying on their ability to unambiguously assign the victim to a set of records that share the same revealing attribute [45], or an exceptional distribution of attributes. In the example of Muslim taxi drivers mentioned above, the attacker inferred an attribute: the victims' religion, even though they did not identify anyone's trajectory. Sui et al. [102] observe that 40% of the records that cannot be immediately identified in their data and seem anonymous were instead homogeneous and directly disclose the shared attribute.

Users' most sensitive locations are another attribute that can be exposed, for example, point-clustering algorithms that can deterministically find them already exist [131]. Gambs et al. [46] demonstrate how this violates the privacy of sensitive attributes.

Group linkage attacks [62] discover connections between individuals. Relationships are particular attribute cases, and both social links and kinship can be inferred from correlated movement [22]. Their disclosure may entail different threats. Predisposition to hereditary diseases, communication between dissidents, homophily in friendships sharing religious and political views, or homosexual partnerships in certain jurisdictions are just a few prominent examples.

Another attack type is **probabilistic attacks**, which aims to improve the probabilistic belief on the sensitive information of a victim after accessing the published data [45]. One typical example is the *Bayesian inference attack*, where the attacker adversary the difference between prior and posterior beliefs about sensitive information, succeeding in the attack when this difference is high (or the posterior exceeds a chosen threshold). We describe in more detail its implications to trajectory data in Section 7.

Reconstruction attacks aim at rebuilding trajectories in the database. For example, Buchholz et al. [9] introduce a reconstruction algorithm that can construct trajectories closer to the original data than the perturbed one. Similarly, *filtering attacks* [113] also aim at reducing noise added. On the other hand, Xu et al. [120] develop an iterative attack that can exploit the uniqueness and regularity of human mobility to step-by-step recover individual's trajectories from mobility data without using any background knowledge.

Finally, we point out that the possibility of predicting a user's locations (**prediction attacks**) is also a threat, since attackers can discover the user's destination, probably even before they arrive. Additionally, adversaries can infer whether users will be home or not, and plan, e.g., a robbery. As an instance of this, Song et al. [98] demonstrate successful *movement pattern predictions* [46] with up to 93% average chance to correctly predict mobility behavior.

5 PRIVACY NOTIONS

There are two well-known families of privacy notions in SDC [59]: syntactic and semantic notions [23]. *Syntactic notions* specify conditions a sanitized database should exhibit; while *semantic notions*[1] describe guarantees that the mechanism chosen for releasing the data should satisfy [28].

Semantic notions can provide stronger privacy guarantees than syntactic notions because they do not require assumptions about the adversary's knowledge. Further benefits over syntactic notions are, for instance, that the sequential composition in DP holds: Specific subsequent publications of the same data yield well-defined leakage that can be controlled. We hence devote the rest of the paper to semantic notions under DP, and give an overview of syntactic notions in Appendix B.

5.1 Differentially Private Notions

Differential privacy (DP) [35] is the best-known semantic notion. It aims to hide the presence or absence of any user in the database such that an analyst can extract statistics about the whole population, while an adversary cannot learn more than a limited amount about any user. The difference between the output probability of a DP mechanism, given a database that contains a user's data and one that does not, is bounded. Thus, the publication of the anonymized output reveals only bounded information about individuals, since the inference capability of any attacker is restricted.

[1] Do not confuse "semantic privacy notions" with "semantic meaning" of a location. The term "semantic privacy" comes from the related cryptographic notion of semantic security, while the term "semantic meaning" of a location relates to its real-world definition and aspects (i.e., the location is a restaurant).

Formally, a randomized algorithm $\mathcal{M}$ is said to be *ε-differentially private* (ε-DP) [35] if for all *neighboring* databases $D, D' \in \mathcal{D}$ (i.e., differing in exactly one entry) and all measurable $S \subseteq \text{Range}(\mathcal{M})$,

$$\text{P}\{\mathcal{M}(D) \in S\} \leq \text{e}^{\varepsilon}\, \text{P}\{\mathcal{M}(D') \in S\}, \tag{5.1}$$

where $\mathcal{D}$ is the fixed universe of all possible input databases of $\mathcal{M}$.

The *privacy budget* $\varepsilon > 0$ represents a measure of the privacy loss after seeing the output. From Eq. 5.1, we obtain $\ln(\text{P}\{\mathcal{M}(D) \in S\}) - \ln(\text{P}\{\mathcal{M}(D') \in S\}) \leq \varepsilon$ for all measurable $S \subseteq \text{Range}(\mathcal{M})$, establishing thus a bound ε over the difference in distributions of outputs between two neighboring databases. Intuitively, the smaller ε, the stronger the provided privacy, i.e., if ε is small enough, then the difference between the two mentioned distributions is negligible. Thus, the attacker has no reasonable criteria to choose between the two possible input databases, limiting the amount of information that can be learned about any given individual.

One strong point of this notion is that it does not make any assumptions about the background knowledge of the attacker. DP is a *worst-case* guarantee [14], which means it protects the privacy of any database (including outliers) against the strongest attackers.

A popular variation, called *approximate DP* or *(ε, δ)-DP* [36] requires instead that $\text{P}\{\mathcal{M}(D) \in S\} \leq \text{e}^{\varepsilon}\, \text{P}\{\mathcal{M}(D') \in S\} + \delta$, relaxing the definition to ensure bounds for rare events. In this case, the probability of not achieving ε-DP (i.e., that Eq. 5.1 does not hold) is δ.

Additionally, both original and approximate DP offer two beneficial properties: *composability* ensures that the combination of multiple DP algorithms is still DP, and *post-processing* implies that subsequent processing does not affect the privacy of data published with DP guarantees. These are given by the *sequential* and *parallel composition* theorems, and the *post-processing* property [36].

Central vs. local DP. The original, *central* DP notion assumes the presence of a trusted party (data curator) who executes the mechanisms protecting the sensitive data. If no party with shared trust exists, it is necessary to distribute the curation to all participants. The corresponding *local ε-differential privacy* (ε-LDP) [36, 63], assumes every individual holds a database containing their records and shares them only after local obfuscation. They hence contribute partial answers to queries on the whole data, enforcing DP locally.

Formally, a randomized algorithm $\mathcal{M}$ that takes as input a user's record is said to be ε-LDP [119] if for all possible pairs of user's records x, x' and all measurable $S \subseteq \text{Range}(\mathcal{M})$, $\text{P}\{\mathcal{M}(x) \in S\} \leq \text{e}^{\varepsilon}\, \text{P}\{\mathcal{M}(x') \in S\}$.

This notion is stronger than central DP since there is no need to assume a trusted party. However, it is usually harder to achieve with the same utility constraints since each user needs to perturb their own record, which does not happen in the central case. Hence, the total amount of noise may be higher in the local scenario. The differences between these notions demand new hypotheses and conditions to satisfy them, as well as adapted mechanisms. A well-known example to achieve local DP in questionnaires is the randomized response.

Level of granularity. DP is a mathematical guarantee, so it is crucial to specify exactly what information is protected by it. The specification hinges on how the concept of neighboring databases is instantiated. Hence, various adaptations of the concept of *neighborhood* (i.e., what is considered a single entry in the database) have been suggested in the literature. We refer to the neighborhood definition as the *level of granularity* [36] of a DP notion. For example, the original DP notion aims to protect the entire existence of an individual's records or entries in a database, thus assuming a one-to-one correspondence between record and user.

Type of privacy	Difference between neighboring databases
User-level	A user's whole trajectories
Event-level	A spatio-temporal point visited by a user (an event)
w-event	A window of events over w consecutive timestamps
ℓ-trajectory	A sequence of ℓ consecutive spatio-temporal points from a single user
Element-level	A user's set of points belonging to the same cluster

Table 1: Granularity notions and their concept of neighborhood.

In trajectory data, where several points form each user's record, the concept of granularity has special relevance. The neighborhood definition directly impacts the privacy guarantee offered. We explore the most common granularity notions in the following paragraphs and provide a quick summary of these in Table 1.

User-level privacy corresponds to the original notion of DP. We consider two databases D and D' to be *user-level neighboring* if they only differ in the information attributed to a single user. For instance, if each user contributes a single trajectory, then two databases D and D' are considered user-level neighbors if they differ in one user's entire trajectory, either by removing/adding their trajectory (*unbounded* DP) or by exchanging it with another user's trajectory (*bounded* DP). In a setting where the same user contributes more than a single trajectory to the database, the neighborhood definition extends to cover all the trajectories of this user.

Event-level privacy appears as an adaptation of DP to streaming scenarios [36], but it is also applicable in a static context when the database is sequential, as in the trajectory case. Its goal is to hide the presence or absence of a single event from a sequence of observations contributed by an individual.

Definition 5.1 (Event-neighborhood)**.** For trajectory data, two streaming databases D and D' are *event-neighboring* if we obtain one from the other by changing a single spatio-temporal point. For example:

$$D = \begin{cases} & S_1 & S_2 & \cdots & S_m \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_m^{(1)} \\ T_2: & p_1^{(2)} & p_2^{(2)} & \cdots & p_m^{(2)} \\ & \vdots & \vdots & & \vdots \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_m^{(r)} \end{cases}, \quad D' = \begin{cases} & S_1 & S_2 & \cdots & S_m \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_m^{(1)} \\ T_2: & p_1^{(2)} & p_2^{(2)} & \cdots & p_m^{(2)} \\ & \vdots & \vdots & & \vdots \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_m^{(r)} \end{cases}.$$

In this context, event-level privacy aims to make it more difficult to determine whether a particular spatio-temporal point has been visited by a given user.

The guarantee this notion gives is that each point in the database remains inaccessible to an attacker. Also, due to the restricted definition of neighborhood, the sensitivity of a query function is never larger (usually smaller) than the sensitivity at user-level granularity. Therefore, for the same ε, mechanisms such as the Laplace or exponential need less obfuscation, so the utility should remain higher. This notion, however, comes with its own drawbacks.

First, it opens the risk of identity disclosure. Event-level privacy guarantees that if the attacker knows a single spatio-temporal point

of a single trajectory, the probability of re-identification is bounded by ε. However, if the attacker knows, for instance, $r > 1$ points, then the protection of said r points is instead bounded by $r\varepsilon$ [36]. Therefore, the attacker's chances of re-identifying can increase significantly, and the victim is no longer protected from identity disclosure. Real-life trajectories can contain hundreds of spatio-temporal records per individual, so the chances of re-identification are then almost certain.

Additionally, it does not fully cover attribute disclosure. If one location has been visited more than once by the same user, then these visits remain unprotected under event-level privacy. For example, if the user visited a hospital more than once, then the attribute "Has been at the hospital" can still be exposed.

Another problem inherent to this notion is its vulnerability to correlation attacks (see Section 7).

***w*-event privacy.** Also considering streaming databases, Kellaris et al. [65] suggest w-event privacy. This notion can be seen as the one that makes points of the database over w consecutive timestamps undetectable when seeing the output returned by the protection mechanism. Its definition of neighboring databases is the following:

Definition 5.2 (w-neighborhood)**.** Let w be a positive integer. Two stream prefixes $D_t = \{S_1, \ldots, S_t\}$ and $D'_t = \{S'_1, \ldots, S'_t\}$ are *w-neighboring*, if, for all $i \leq t$, S_i and S'_i are either equal or we can obtain one from the other by changing an entry of S_i, and all pair of indexes i, j corresponding to the latter case verify that $|i-j| < w$. This last condition means that all the differing S_i and S'_i must fit in a *w-window* (see Figure 1).

This definition captures settings where sensitive information is disclosed from a sequence of events of length w. It does not only protect the locations visited by a single user over w consecutive timestamps but also can protect those of different users. In terms of privacy, for values of w close to 1, w-event privacy approximates to event-level privacy, and for large values, it converges to user-level privacy. In terms of sensitivity, its lower bound is the event-level sensitivity, and its upper bound is the user-level one. Therefore, this notion protects more information than event-level privacy while allowing less noise addition than user-level, even though some of its deficiencies remain present.

The notion still leaks attributes (e.g., "Being at the hospital"), when these cannot be protected by the same w-window. For example, assume user u_1 in Figure 1 (where $w = 3$) is a compulsive gambler and visits the casino (red dot) multiple times a day. The sensitive information that u_1 has been at the casino is not protected as the red dots cannot fit into a unique w-window. Also, the user's identity is still unprotected if the attacker's knowledge exceeds the window.

Given that consecutive spatial points are usually more correlated, this new notion is also superior to event-level privacy against correlation attacks (see Section 7). However, the assumption of w-event privacy that trajectories are periodically recorded, may overestimate the number of consecutive protected locations. For instance, in Figure 1, where we have non-periodically recorded trajectories, the 3-window 5–7 cannot protect more than two locations of a single user.

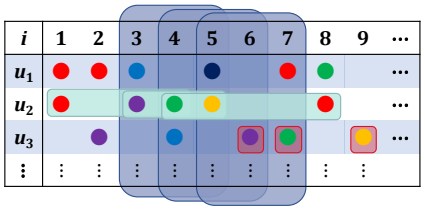

Figure 1: An example of a non-periodically recorded streaming database, colored dots represent different locations. The rounded boxes represent protection scopes of event-level (red), w-event (blue), and ℓ-trajectory privacy (green), for $w = \ell = 3$. Observe that the blue box (w-window) always spans w timestamps independent of how many points they include, and that the green box (ℓ-trajectory) always includes ℓ points independently of the number of timestamps it spans.

***ℓ*-trajectory privacy.** Cao and Yoshikawa [12] aim to overcome this last deficiency of w-event privacy, especially when users' trajectories are not periodically recorded. To tackle this issue, they extend the previous model to introduce ℓ-trajectory privacy, defined next.

Definition 5.3 (ℓ-trajectory neighborhood)**.** We say two databases are *ℓ-trajectory neighboring* if one is obtained from the other by only modifying all locations in a single ℓ-trajectory. Here, an *ℓ-trajectory* is defined as a sequence of ℓ successive spatio-temporal data points produced by the same user (see Figure 1).

The goal of the ℓ-trajectory privacy notion is to protect each sequence of ℓ points from the same user independently of the number of timestamps they span. Clearly, varying ℓ allows us to move closer to event-level ($\ell = 1$) or to user-level privacy ($\ell \to \infty$).

Although this notion overcomes the problem of w-event privacy of assuming periodically recorded trajectories, it does not address its other deficiencies.

Element-level privacy. The high cost and utility loss of user-level privacy motivated the definition of element-level privacy [5]. The authors argue that, in most cases, participation in a database (user-level privacy) is not sensitive. For instance, in a traffic study, participating in the database only discloses information such as "Having a car" or "Living in the area", which users can regard as insensitive. However, one may wish to avoid attribute disclosure. For example, suppose a person visits a hospital several times throughout the day and wants to keep it a secret. In this case, event-level privacy is not sufficient for the previously mentioned reasons, and neither w-event nor ℓ-trajectory privacy if the visits do not fit into the w-window or ℓ-trajectory. Therefore, the authors propose element-level privacy to address this situation.

The original proposal [5] models the data of a user u as a multiset of values $x^{(u)} = \{x_1^{(u)}, x_2^{(u)}, \ldots, x_{m_u}^{(u)}\}$, where each $x_i^{(u)}$ belongs to the universe of possible values $\mathcal{X}$. Then it considers a K-partition of the universe $\mathcal{X}$ into the clusters $c_1, \ldots, c_K$. These clusters are viewed as the *elements* to be protected. By definition, each $x_i^{(u)}$ belongs to one cluster c_j.

Definition 5.4 (Element-neighborhood)**.** Two databases D, D' are *element-neighboring* if they are equal except for a pair of users' data $x^{(u)} = \{x_1, \ldots, x_n\} \in D$, $x^{(u')} = \{x'_1, \ldots, x'_n\} \in D'$ such that

$$d_{\text{user}}(x^{(u)}, x^{(u')}) := \sum_{k=1}^{K} \mathbf{1}_{\{\{x_i | x_i \in c_k\} \neq \{x'_i | x'_i \in c_k\}\}} \leq 1,$$

where $\mathbf{1}_{\{\{x_i|x_i\in c_k\}\neq\{x'_i|x'_i\in c_k\}\}}$ denotes the indicator function that outputs 1 when the inequality holds and 0 in other case, with $\{x_i \mid x_i \in c_k\}$ being implicitly multisets.

Observe that by modifying the cluster selection, we can achieve user-level granularity by taking only one cluster, $c_1 = \mathcal{X}$.

The interpretation of ensuring element-level privacy is that we are hiding that each user has elements belonging to the cluster, independently of how many elements it includes.

We believe that this notion can be adapted to trajectory data. In the case of raw trajectories, we can cluster data points according to geographical zones and times. And in the case of semantic trajectories, we can choose the clusters according to semantic values, e.g., having a cluster for all health-related locations.

A challenge here is how to establish the clusters to provide real protection that covers all possible scenarios regarding the user's privacy desire. For instance, if we choose spatial areas as clusters, a question arises about the size we should take and which privacy guarantees we would have according to our selection. This extends when considering semantic trajectories: If we reduce a cluster to a specific hospital, we will protect the visits to this hospital, but if we instead include all hospitals in the cluster, we will then be protecting any visit to any hospital.

Since this notion is relatively new, there have not been many mechanisms achieving it and no adaptations specifically to trajectory data. Moreover, no comparison has been conducted against the other granularity notions explored in this section regarding utility and privacy.

Conclusions on Granularity Notions: In terms of privacy protection, user-level privacy is the strongest, followed by element-level, ℓ-trajectory, and w-event privacy. Finally, we fear event-level to be unreliable regarding trajectory privacy.

Although choosing user-level privacy may result in excessive loss of utility in the complex field of trajectory publication, none of the other granularity notions adapted to trajectory data analyzed here can provide convincing privacy guarantees. All of them allow identity disclosure, and none provide effective protection against attribute disclosure. Even if participation in a database was not sensitive information, leaking user attributes seems unacceptable in terms of privacy. Element-level privacy could be a promising attempt to protect against attribute disclosure. However, it has not been adapted to trajectory data yet. Therefore, it is difficult to assess the impact of this notion on the utility of anonymized trajectories.

6 DP MASKING MECHANISMS

Next, we examine masking algorithms that adapt trajectory databases for publication with DP guarantees. The state of the art we review in this work covers the static-context publication in which the sanitized database is released just once in its entirety, without subsequent updates. We classify them according to their fundamental concept. We provide an overview of our classification in Table 2, including information on the privacy notion they satisfy, their properties, and the utility metrics used for their evaluation. Observe that most of the reviewed proposals aim to achieve user-level DP.

Note that DP algorithms require a randomized approach since deterministic algorithms cannot achieve DP guarantees [36]. The two classical mechanisms to provide DP are the *Laplace* and *exponential mechanisms* [36]. Nearly all the algorithms presented in this section leverage these mechanisms in some way.

6.1 Noisy Counts

We include in this class the anonymization approaches that add Laplace noise to the count of trajectories or their subsequences.

Exploration tree. Chen et al. [18] first construct an exploration prefix tree from the trajectory database. Each node is labeled with a possible location, which can only be an element of a predefined finite set of locations (the *universe of locations*). Every possible prefix trajectory is represented uniquely as a walk from the root node to another (i.e., we represent prefix $\langle p_1, \ldots, p_m\rangle$ by the node obtained after walking through the tree following the labels: $root \rightarrow p_1 \rightarrow \cdots \rightarrow p_m$). This node stores the number of times (i.e., the *counts*) the prefix appeared in the database. The tree includes all the possible trajectories drawn from the universe of locations, including those not present in the database (i.e., with a count of 0).

This way the count of each prefix of length n is stored at the nth level of the tree. To guarantee DP, Laplace noise is added to the count of each node (including the 0 ones, potentially creating sequences not contained in the original data). Since each trajectory has only one prefix of length n, the sensitivity of the mechanism is 1. A node with a noisy count of 0 becomes a leaf; otherwise expands until a maximum allowed length.

Then, to release the trajectory database, we only need to explore the resulting tree. Based on the noisy prefix tree, we can draw the sanitized database by traversing it once, calculating the number r of trajectories terminated at each node, and appending r copies of the prefix saved in that node to the output. Since creating and exploring the tree are inverse operations, there is a one-to-one correspondence between the database and the prefix tree. Note we need a post-processing module to maintain the tree consistency (i.e., the sum of counts of descendant nodes cannot be higher than that of their ancestors).

In subsequent work, the same authors improved this approach by introducing an n-gram exploration tree [17] that looks at n-grams instead of prefixes, which leads to higher counts in each node and higher sensitivity. In this case, each trajectory could add its total length to a node count. Therefore, the sensitivity is the maximum trajectory length, l_{max}, allowed in the database (any trajectory longer than l_{max} is cut before introducing it in the data). The authors also add Laplace noise on the n-grams counts. Once again, by exploring the tree, we recover the perturbed version of the original trajectories, obtaining a sanitized trajectory database. The proposed solution [17] additionally offers the possibility of creating trajectories using a Markov process, where they compute the probabilities using the noisy counts. However, this option does not create a modified database from the original (masking) but instead generates synthetic data.

Other proposals modify these algorithms in various ways. Firstly, Wang and Kankanhalli [114] define sensitive zones and apply Chen et al.'s method [17] only to these zones, which provides better utility. However, their privacy notion is weaker since they do not provide DP for the whole database but only for sensitive zones.

Privacy notion		Classification		Ref.	Correct DP notion	Mech. (Δ): Laplace	Mech. (Δ): Exponential	Properties: Considers time	Properties: Unb. loc. univ.	Properties: Realism	Utility metrics: Total data preserv.	Close data preserv.: SM: *Euclidean*	Close data preserv.: SM: *Hausdorff*	Close data preserv.: SM: Other	Close data preserv.: Other	Statistics preserv.: *Loc. visit counts*	Statistics preserv.: *Freq. seq.*	Statistics preserv.: *Spatial density*	Statistics preserv.: Other	Utility metrics: Realism assurance
						Mech. (Δ)		Properties			Utility metrics									
User-level	ε-DP*	Noisy counts	Exploration tree	[114]		l_{max}	○		✓		●	○	○	○	●	●	●	●	●	●
	ε-DP			[18]		1	○				○	○	○	○	○	●	●	○	○	○
				[17]		l_{max}	○				○	○	○	○	○	●	●	○	○	○
				[31]		l_{max}	○				○	○	○	○	●	●	○	○	○	○
			Sequence tree	[130]	✗	●	○				○	○	○	○	○	○	○	○	●	○
				[128]	✗	●	○				○	○	○	○	○	●	○	○	○	○
				[124]	✗	●	○				○	○	○	○	○	○	○	○	●	○
			Trajectory count	[127]	✗	●	○		✓	✓	○	○	○	○	○	○	●	○	○	○
			Tree + Markov	[11]	✗	●	○		✓		○	●	○	○	○	○	●	●	○	○
		Clustering	Random centroid			○	○													
			Exp.: k-means	[21, 58]	✗	●	●		✓		○	○	●	○	○	○	○	○	○	○
				[72]	✗	●	●		✓		○	○	●	○	○	○	○	○	○	○
			Exp.: Hilbert curves	[55]	✗	●	●		✓		○	○	●	○	○	○	○	○	●	○
			Universal clustering	[129]	✗	●	○		✓		○	●	○	○	○	○	○	○	○	○
Event-level	$(0, \delta)$-DP	Sampling + interpolation		[97]		○	○		✓		○	●	○	●	○	○	○	○	○	○
User-level	(ε, δ)-DP			[76]		ΔX	○	✓	✓		○	○	○	●	○	●	○	○	●	○
	ε-LDP	Perturbation		[26]		○	Δd_w	✓	✓	✓	○	○	○	●	●	●	○	○	○	○

Table 2: Summary of explored DP-based mechanisms according to our classification and exact privacy notion they satisfy. "Correct DP notion" labels mechanisms that incorrectly claim DP. We show if the algorithm uses the Laplace or exponential mechanism, and the corresponding sensitivity (Δ) of correct proposals (sensitivity is not well-defined for the incorrect algorithms). Next, we cover basic properties: whether they consider time, allow for an unbounded location universe, and assure realism. We then specify which classes of utility metrics are used to evaluate the mechanism (cf. Section 3). We highlight the most representative metrics according to the selected mechanisms. "Close data preservation" includes two specific similarity measure (SM) types: Euclidean and Hausdorff distances. "Statistics preservation" includes "location visit counts" (including location popularity metrics), "frequent sequences" and "spatial density". For noisy counts and clustering, colored cells indicate the original proposals from which the others in each family stem. *It provides ε-DP only when restricted to certain spatial areas.

DPLG [31] constructs the same noisy n-gram tree (therefore, the sensitivity of each node count is l_{max}) but provides a non-uniformly distributed privacy level by regulating the amount of noise added, so the location will be more or less protected depending on the area of the map it is.

All the exploration-tree–based methods have some common problems: For instance, it is necessary to assume a fixed and discrete universe of possible locations and set the maximum length of trajectories. We need these strong assumptions to bound their sensitivity. Also, the size of the trees increases exponentially with the number of locations and allowed length of trajectories. Note that limiting length would considerably reduce utility. Hence, a small location universe is required to perform these methods, which is not usually the case in real-world applications. Additionally, the mechanisms only retain spatial information and counts, with the loss of temporal information further reducing utility.

The spatio-temporal correlations of human trajectories, their regularity, and self-similarity can be easily represented by auto-correlation models (see Section 7). Some of the new sequences generated by the processes do not follow realistic patterns and hence can easily be identified and removed from the data by the adversary. The accuracy of this attack depends on the quality of the adversary's correlation model. The Laplace mechanism, however, does not consider correlations and is bound to choose impossible or highly unlikely sequences when adding noise to the original zero counts of these hypothetical trajectories. A simple stochastic model aggregating road-map information and physical movement laws will suffice to eliminate these cases.

Sequence tree. More recent approaches try to build trees storing the counts of subsequences in each node instead of only one location (i.e., *sequence trees*). This is the case of NTPT [130]. This mechanism first tries to overcome data sparseness by simplifying the trajectories. By performing an optimal segmentation process, the trajectories are divided into sequences, and then, it constructs a prefix tree where each node stores a sequence. Afterward, it adds Laplace noise to the counts of each node.

Related approaches are presented in [124, 128], with the difference that they rely on a similarity factor. More specifically, they save sequences of spatio-temporal points in a tree structure according to the number of location points they have in common. As usual, they add Laplace noise to the count of each sequence node.

Trajectory count. Finally, one work considers the correlation between individuals in the database [127]. Here, the authors measure the correlation coefficient between the different trajectories in the database, which translate into privacy risk: the more correlated trajectories are, the more risk they pose. Therefore, they allocate different privacy budgets adding more Laplace noise to the counts of the risky ones.

We would like to note that all of the above suggestions [124, 127, 128, 130] suffer from a common formality mistake and do not provide DP. They output perturbed counts of only those segments, subsequences, or trajectories present in the original database, but do not change the output of hypothetical sequences with zero

counts, as in the exploration-tree–based methods we discussed. These conditions contradict the definition of DP, and thus cannot provide DP (we provide formal proofs of this in Appendix C). This is not reflected in the privacy analysis, as the authors provide proof of the DP tools they incorporated, such as the Laplace mechanism, but do not of the privacy met by their global algorithm. Consequently, if the count of the victim's trajectory is positive after perturbation, and this trajectory contains a quasi-identifier known by the attacker, such as their home or work, the victim and the rest of its path can still be identified.

Conclusions on Noisy Counts: We conclude that the only noisy-count mechanisms that achieve acceptable privacy guarantees are the original exploration-tree approaches [17, 18, 31, 114]. However, due to their high computational cost for large databases, we only see these methods used for cases with reduced universes, such as the analysis of public-transport lines of a city.

These algorithms excel at preserving statistics (e.g., location counts). This result is reflected in Table 2, where we see that many of the algorithms evaluate their utility using statistic-preservation metrics. On the other hand, we find fewer evaluations using data-preservation metrics and, in particular, no similarity measures.

6.2 Clustering

The next category contains mechanisms [11, 21, 55, 58, 72, 129] that cluster locations and subsequently release trajectories through these clusters with some perturbation to guarantee privacy.

They follow a common structure that consists of two privacy mechanisms: A generalization mechanism M_1, which generalizes the set of locations by grouping them into clusters, and a releasing mechanism M_2, which outputs resulting trajectories drawn from the generalized sets. To achieve DP publication, both M_1 and M_2 have to be DP.

Exponential clustering. Hua et al. [58] is the first proposal using clustering. Their idea for M_1 is to cluster and merge concurrent locations from different trajectories, following a probabilistic partitioning based on the exponential mechanism. Then, using the Laplace mechanism, M_2 connects the merged locations and forms the final generalized trajectories.

Specifically, the authors suggest a score function to measure distances between trajectories crossing the corresponding locations at each timestamp. Using the exponential mechanism and this score function, they choose one of the candidate partitions (into m groups) of Γ_i, the set of locations of the database at time i. Finally, the locations of each subset are clustered together and replaced by their corresponding centroid (see Figure 2).

After selecting a partition and replacing the locations with centroids, the location set Γ_i is replaced by a smaller one, $\tilde{\Gamma}_i$, which contains perturbed information. They build the new trajectories from this reduced set $\tilde{\Gamma}_i$ using the mechanism M_2, which draws sequences from $\tilde{\Gamma}_i$ at random. The counts are attributed following the Laplace mechanism until obtaining a sanitized database of the same size as the original.

To ensure privacy, this model is imported in [21] as the final part of their recurrent neural network. Later, Li et al. [72] design an M_2 algorithm with bounded Laplace noise. In [55], they propose a new private cluster mechanism M_1 based on Hilbert curves, where it is not necessary to fix the number of clusters in advance.

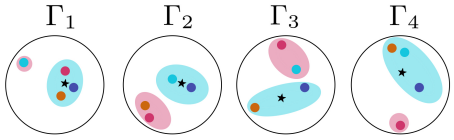

Figure 2: Example of how trajectory data are anonymized through clustering techniques. Different trajectories are represented in different colors, with points corresponding to the physical location over each timestamp. The colored areas represent the clusters defined by the selected partition, and the stars denote the centroids of each subset. In this case, trajectories are of length $|T| = 4$, and the selected partition contains $m = 2$ subsets.

After studying these approaches, we observe an issue applying the exponential mechanism. Recall that this mechanism assigns probabilistically for each input an output from $\mathcal{R}$, the set of abstract outputs, which is data independent. However, in these proposals, $\mathcal{R}$ is the set of possible partitions of $\Gamma_i \coloneqq \{l_i$ location at time i of any $T \in D\}$. If the database D is changed, then it is clear that the possible partitions are also changed, which means that $\mathcal{R}$ is not data independent. This fact breaks the formal DP proof for the M_1 mechanisms since the exponential mechanism is not well-defined if $\mathcal{R}$ is data dependent, and so Eq. 5.1 cannot be achieved (we provide a formal mathematical proof of this statement and an example that breaks the DP guarantees of these methods in Appendix C). The only way to avoid this issue would be to define a data-independent universe of locations, for instance, based on the city map, and output a partition of this universe. This way, the mechanism could achieve DP. Being independent of the actual patterns in the data could incur a significant utility loss in some scenarios.

Universal clustering. Recently, Zhao et al. [129] introduce a protection proposal independent of the specific clustering algorithm. It allows one to choose any preferred clustering and run it on the database without modification. They add Laplace noise to location coordinates (using the polar form) and to the counts of these data in the cluster. Finally, the authors calculate the noisy centroid according to the noisy counts and locations and release these centroids. The noisy count algorithm they use is the same as in [124, 128, 130], which we have shown to lack DP guarantees. Furthermore, following this scheme, we cannot release more than the corresponding centroids since there is no private way of establishing connections between centroids and thus forming trajectories without using the original data. The authors do not propose any mechanism for trajectory release (M_2).

Random centroid. Finally, we highlight DPTD [11], which introduces a generalization module that clusters the locations without consuming privacy budget (the proposed solution chooses a random location instead of the centroid). For the release method M_2, the authors adapt the noisy prefix tree structure presented in [17] to reduce the consumption of the privacy budget and provide higher utility. Instead of adding Laplace noise to the odd layers of the tree, they predict the new count with a Markov process. This Markov process uses the frequencies of the original database, apparently without protection (i.e., no noise or perturbation added to the frequencies). Although the authors attempt to reduce the privacy budget consumed, the generalization step indirectly uses the database

in its election of the centroid, thus breaking DP. The publications also contain neither analyses nor proofs of privacy, so the actual protection achieved remains unclear.

General problems. Apart from the privacy issues we have explained in each proposal, we find general problems. First, the generation of impossible trajectories challenges the utility of the resulting output. Specifically, the presented methods can create trajectories in which two consecutive locations are unreachable in the given time and unrealistic centroids placed at impossible locations, such as in the middle of a river or on top of a building.

Another limitation is that the used score function of the exponential mechanism only depends on physical distance and therefore does not consider time. These proposals are thus inapplicable for non-periodically recorded and variable-length trajectories, which represent a majority of real-world databases.

Similarly, a problem arises related to stationary sequences when disregarding time. When a driver stops, the spatial location remains constant during each timestamp until the car starts to move again (e.g., see Figure 2, where the dark blue point is constantly in the same location at each timestamp because it represents a stop position in the trajectory). The constant spatial points will be substituted by the corresponding centroids at each timestamp. However, since merging locations is only based on distances, the sanitized data will likely not reflect this stop. In Figure 2, we can see that the locations of the dark blue stationary trajectory change into different locations at each timestamp. This produces an apparent random movement that hides the stop.

Conclusions on Clustering: This category of approaches overcomes the applicability problem of those using trees (see Section 6.1), as they do not need to assume a small universe of locations. However, we can still identify several deficiencies: merging without considering time and using naïve mechanisms for releasing data (M_2) can yield poor utility and facilitate correlation attacks. Also, as mentioned above, all of these proposals contain erroneous DP analyses or proofs. It hence remains unclear which protection they provide.

Unlike in the noisy-counts algorithms, these clustering mechanisms evaluate utility mainly using similarity measures rather than statistic-preservation metrics (see Table 2), even though these last ones could be used in the utility evaluation.

In combination with the development of better release mechanisms and rigorous privacy analyses, these approaches promise to be a fruitful path for future potential research.

6.3 Sampling and Interpolation

Another type of mechanism is based on point sampling and interpolation [76, 97]. The sampling technique consists of selecting a subset of the database (in this case, trajectory points), while interpolation is used to counteract the size reduction due to sampling by reconstructing intermediate points of the trajectories. The sampling techniques used do not satisfy ε-DP, but rather (ε, δ)-DP, and interpolation is conducted without affecting the privacy guarantees.

Shao et al. [97] present two mechanisms, SFI and IFS, for ship-trajectory privacy based on these techniques. SFI first randomly samples points over each trajectory and then redraws trajectories using a cubic Bézier interpolation (the "a priori" mechanism). IFS first interpolates and then samples (the "a posteriori" mechanism). The mechanisms are proven to achieve event-level $(0, \delta)$-DP. In their experimentation, the authors conclude that SFI works better than IFS for small values of δ and not-so-smooth trajectories.

Similar to the mechanisms discussed in the previous subsections, this algorithm ignores the temporal dimension, and impossible trajectories can thus occur. Furthermore, even though SFI and IFS guarantee high utility for smooth ship trajectories, we believe this result might not extrapolate well for other trajectory types, like people or road vehicles, which can contain sharper turns and need to fit into a road network.

Another proposal is VTDP [76], which consists of a three-phased sampling with a final interpolation step and satisfies (ε, δ)-DP. Each of the sampling phases constructs from the previous following a well-known distribution. The first phase considers position and counts, the second additionally considers moving speed, and the third adds the temporal component. Interpolation is computed simply using the basic formulas between speed, acceleration, and time. The algorithm also uses the Laplace mechanism during the first phase to find how many elements points are to be sampled. The sensitivity of this mechanism is $\Delta X = \max_{D,D'} \|x_i - x_i'\|$, where x_i and x_i' are the optimal counts of points P_i returned by an optimization process depending on D and D', respectively. However, there is no bound or further analysis of this sensitivity. Without a bound, it is not possible to apply this mechanism to satisfy DP properly.

The mechanism aims at preserving the original distributions and maintaining high utility throughout. With this privacy guarantee, the probability of protection against attacks such as record linkage is only $1 - \delta$. However, the authors evaluate their proposal over a database consisting only of a section of an arterial road, which asks whether the mechanism will maintain the same utility results over other trajectory databases.

6.4 Local Perturbation

While LDP proposals for location privacy start to appear [111], we only find one protection mechanism [26] that perturbs semantic trajectories to satisfy ε-LDP. Recall that these trajectories are a time-ordered sequence of POIs visited by a user. The authors integrate public knowledge to improve the utility without affecting the privacy budget ε. The proposed mechanism utilizes this public knowledge to partition the set of all POIs into spatio-tempo-categorical regions, such that each contains some number of POIs.

The mechanism is divided into four parts: first, it generalizes every POI into the corresponding region; it partitions these new trajectories into n-grams, which are then individually perturbed following the exponential mechanism to ensure ε-LDP, where the score function is a distance function d_w defined over the three dimensions (see Appendix A.4); then trajectories are reconstructed by minimizing the distance function; and finally the mechanism returns to the initial domain by randomly picking a POI for each section, making sure that consecutive locations in a trajectory are reachable in the corresponding time.

This mechanism demonstrates several advantages over those described above. First of all, ε-LDP is a stronger privacy guarantee than ε-DP since there is no need for a trusted curator. Furthermore,

it does consider the temporal dimension (and the categorical dimension of the trajectories). It also takes into consideration publicly available information to improve the overall utility of the mechanism, without any effects on the privacy budget, and ensures that the published data is realistic.

However, it also faces some challenges: First, to adapt the mechanism to a multiple-release setting (i.e., the same user contributing more than one trajectory), the user needs to know in advance how many trajectories they want to share, to divide the overall privacy budget by this number [26]. Adapting this approach to a streaming scenario will encounter the same challenge.

Second, the sensitivity of the exponential mechanism, Δd_w, depends on the fixed data universe. This means that it can be reasonable in small spatial areas, short time intervals, and reduced semantics, but if we consider huge spatio-temporo-categorical domains, the amount of noise needed will spoil the utility results. The authors also point out in their utility analysis that the error increases with trajectory length. The mechanism hence lends itself to small regions, for instance, the mobility within a city, rather than databases covering large areas.

It is also worth mentioning that this approach has been presented as a solution for societal-contact-tracing applications. In other use cases (e.g., traffic management), driving patterns and traffic flow are more important than semantic values. Adapting the approach to fields such as these seems interesting, but has not yet been investigated.

7 CHALLENGES AND LIMITATIONS OF DP IN TRAJECTORY PRIVACY PROTECTION

DP has become the formal and de facto mathematical standard for privacy-preserving data release. Yet, recent works [19, 24, 25, 43, 64, 69, 77, 121] have demonstrated various challenges and shortcomings that this notion encounters when applied to trajectories. First, we discuss some challenges and difficulties of the application of DP to trajectories that are yet to be overcome in the literature.

Infinite streaming context. Trajectory data analysis usually requires users to continuously share spatio-temporal updates. One of the advantages of DP is its composition property. It allows publishing subsequent database updates with linearly increasing privacy loss: with r updates, the release consumes $r\varepsilon$ privacy budget. The main obstacle to protecting subsequent releases of dynamic data is that the overall privacy budget is consumed completely at some time [71]. The situation worsens when aiming to publish sanitized databases rather than global statistics since the corresponding sensitivity is usually much higher. The possibility of protection is finite in time, and parametrization gets complicated: the larger the number of releases, the smaller the ε assigned to each of them, and thus, the more noise added. This problem affects various use cases of trajectory data release. Traffic-jam prediction and avoidance are examples where users need to update their locations and trajectories in real time. Standard DP hence cannot be used sensibly in the streaming context, while granularity adaptations to this context, such as event-level and w-event privacy, still show serious privacy deficiencies (as mentioned in Section 5). Therefore, the DP adaptation to dynamic trajectory sharing is still an open challenge in the scientific community.

Outlier protection [54] is related to the significant utility loss incurred by the amount of noise that *outlying* sequences or trajectories (i.e., that they differ significantly from mostly any others) require to be protected. As we mentioned in Section 2, trajectories are high-dimensional and unique [29], increasing the chances of singling out or identifying records in comparison with simpler databases. In particular, the sensitivity of this type of data remains high. However, DP is a worst-case metric and it must therefore add larger amounts of noise to hide these outlying records. This is because, in most DP mechanisms, the noise added is directly proportional to the sensitivity and inversely to ε.

Therefore, if we assume the sensitivity is fixed, the only way of reducing noise is by increasing ε. This problem leads to two undesirable opposites: choosing a smaller ε to protect the outliers, which itself leads to lower utility in the whole of the database, or choosing a larger ε, leaving the outliers especially unprotected. Observe that this choice feels excessive since, with larger ε, non-outliers likely remain protected; but it is only the privacy concerns of possible outliers that impede this scenario because they can be outliers even after sanitization.

The challenge of finding a good trade-off between obfuscation and ε remains open in the literature. Some works [54] already proposed additional outlier-control mechanisms to ensure that these plausibly blend into a crowd of users' trajectories. Such techniques could help attain better ε while avoiding the associated immense protection lost.

On the other hand, we also have intrinsic limitations of the DP notion, especially notorious in the trajectory context, that require modifications of the metric itself.

First, we encounter the **Bayesian inference threat**, which implies prior knowledge of an attacker. Taking the example from [54]: Suppose that 10% of a population lives in a district. The prior expected percentage of patients from this neighborhood in the only hospital is around 10%. Imagine now that the released data shows that 70% of trajectories stopping in the hospital are from this district. Since the difference in values between the prior and posterior beliefs is notable, we assert that there is a privacy leakage (i.e., a health-related problem in the district). However, data should not disclose health-related information when the goal is to predict traffic jams, and there is no need to learn about health situations. DP by itself does not provide any guarantee against this phenomenon. We cannot measure how much we modify the distance between prior and posterior beliefs or if it is enough to hide sensitive information. Protection against this attack must ensure that the difference between prior and posterior about a sensitive attribute or information from data participants is sufficiently small.

This attack should not be confused with the *inference privacy fallacy* [68]. Bounding all the posterior vs. prior beliefs would end in zero utility and no possible inference process. However, we aim to protect the people participating in the database from sensitive inferences that are unnecessary for the data-analysis purpose.

Finally, problems regarding **correlation in trajectory data** in databases have recently been observed in several works [13, 19, 24, 69, 77, 121]. DP inherently assumes the database is a simple, independent random sample. This assumption implies that the database records are uniformly distributed (i.e., follow the same probability

Figure 3: The green location is naturally no sensible alternative for the original blue point. Jumping from one location to another far away in seconds is not possible in real life, which is easily modeled with correlations. Changing that location also would imply changing the nearby points. Map screenshot from © OpenStreetMap contributors [87].

distribution) and independent (in particular, non-correlated). As we explained in Section 2, this is not the case for trajectory data.

One problem for DP caused by correlation relates to the difference between theoretic and real-world sensitivity:

Example 7.1. Suppose that Alice and Bob are married and an adversary who wants to infer the origin of Alice's trajectory. The corresponding inference attack determines how probable the output database is, conditioned to Alice starting at a selected point or not, and chooses the answer that maximizes the probability. Now, given their relationship, Alice's and Bob's trajectories share points in their daily life. These could relate to their home or their favorite supermarket. The origins of Alice's and Bob's trajectories hence are highly correlated. Suppose we select a location and query the database for the number of trajectories starting at this point. If we assume independence, the sensitivity of such a query is 1 (user-level), as two neighboring databases can differ in a single user's trajectory, and each trajectory has only one origin. Therefore, ε-DP is satisfied by adding Laplace noise drawn from $\text{Lap}(\frac{1}{\varepsilon})$. However, in reality, Alice's and Bob's answers are positively correlated. Therefore, with very high probability, the difference in counts between a database where Alice started in the selected location and another where Alice did not is 2, since Bob's answer also changes. The correlation model, considered background knowledge, helps an attacker to infer Alice's record as the probability distributions will be further apart than the expected ε bound.

Cao et al. [13] demonstrate how this problem greater affects protection under event-level privacy due to the autocorrelation between nearby spatial points. As we see in Figure 3, each spatio-temporal point affects other nearby points, simply due to the laws of physics and external limitations, such as road networks. As we mentioned, event-level privacy aims to protect the existence of each spatio-temporal point in the database. However, if the attacker uses autocorrelation knowledge, then the difference between the output distributions of Eq. 5.1, conditioned to whether the target spatio-temporal point is in the database or not, will not be bounded by ε anymore. This helps the attacker to guess whether the point was originally in the database by just looking at the output.

Attribute correlations allow an adversary to invert simple perturbations: Applying time-series filters, such as the Kalman or Wiener filters, effectively removes the noise added by sanitization mechanisms, as shown by Wang et al. [113]. The post-processing property of DP should intuitively prevent such attacks. However, it relies on the independence of records and breaks due to correlation.

Some notions of DP attempt to take correlations into account to overcome this issue, such as *Bayesian DP* [121] or *dependent DP* [77]. Unfortunately, they have not been analyzed in the context of trajectory privacy yet, and their adaptation is all but straightforward.

8 CONCLUSIONS

Privacy in human traces is not a direct task since it is high dimensional, unique, and correlated. With this work, we offer a relevant systematization for the community on state of the art on private publication of trajectories. We examine how to represent these data and which aspects they can capture; and recompile, summarize, and analyze the most relevant privacy and utility metrics and DP masking mechanisms in the field. Additionally, we mention new research paths and point out mistakes in the current proposals to avoid their future repetition.

More precisely, we have developed a classification of utility metrics in the field and explained their applications and use cases. Then, we have discussed the current DP-based privacy notions in the context of trajectory data, highlighting their advantages and drawbacks and concluding their applications. Subsequently, we have conducted a comprehensive and systematic analysis of the current state of affairs, classifying the trajectory masking DP mechanisms into four main categories, where we discussed their privacy and utility issues and proved formal errors. And we have presented the main challenges and limitations that DP encounters due to the specific properties of trajectory data.

From our comparison and analysis, we can extract some general conclusions. First, there is a wide range of utility metrics that can be used in the evaluation of a mechanism. Significantly, there is no universal metric, and not every single one is suitable for all scenarios. We also reiterate the importance of publishing realistic data (or of using realism-assurance metrics) since unrealistic data hinders utility and are easily identifiable by attackers.

We point out that the literature presents many privacy mechanisms with apparent flaws. For example, we emphasize the relevance of considering the time in trajectory data protection, as per the discussions. Additionally, since many of the reviewed proposals do not provide DP, we would like to highlight the importance of carefully checking that a mechanism does so. Many of those proposals rely on a well-known DP mechanism but do not correctly define or adapt it, leading us to think the hypotheses of the exponential and Laplace mechanisms are not well-grounded. Among our reviewed proposals, we distinguish Cunningham et al.'s mechanism [26] for improving the pre-existent issues and finding a way of using public knowledge to enhance the utility of the mechanism.

Moreover, we pointed out the necessity of adapting DP to be robust against attacks, such as those based on correlation, which is currently notorious in trajectory DP mechanisms. We conclude the need for more robust metrics adapted to the mentioned trajectory properties.

Furthermore, independently of the mentioned flaws, much remains to be achieved in real-world scenarios. These current proposals cover just a few applications (i.e., they focus on semantic queries in a small location universe and societal contact tracing). Many other areas, such as driving patterns remain unexplored.

In summary, we believe that the research toward the privacy goal in the publication of human-mobility data remains an open and quite fruitful field.

ACKNOWLEDGMENTS

We would like to thank the reviewers and shepherd for their useful comments and suggestions in the improvement of this paper.

Javier Parra-Arnau is the recipient of a "Ramón y Cajal" fellowship funded by the Spanish Ministry of Science and Innovation. This work also received support from "la Caixa" Foundation (fellowship code LCF/BQ/PR20/11770009), the European Union's H2020 program (Marie Skłodowska-Curie grant agreement № 847648) from the Government of Spain under the project "COMPROMISE" (PID2020-113795RB-C31/AEI/10.13039/501100011033), and from the BMBF project "PROPOLIS" (16KIS1393K). The authors at KIT are supported by KASTEL Security Research Labs (Topic 46.23 of the Helmholtz Association) and Germany's Excellence Strategy (EXC 2050/1 'CeTI'; ID 390696704).

REFERENCES

[1] Osman Abul, Francesco Bonchi, and Mirco Nanni. 2008. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In *Proc. IEEE Int. Conf. Data Eng. (ICDE)*. 376–385.

[2] Osman Abul, Francesco Bonchi, and Mirco Nanni. 2010. Anonymization of moving objects databases by clustering and perturbation. *Inform. Syst.* 35, 8 (2010), 884–910.

[3] Charu C. Aggarwal. 2005. On k-Anonymity and the Curse of Dimensionality. In *Proc. Int. Conf. Very Large Databases (VLDB)* (Trondheim, Norway). 901–909.

[4] Pichamon Anantasech and Chotirat Ann Ratanamahatana. 2019. Enhanced Weighted Dynamic Time Warping for Time Series Classification. In *Proc. Int. Conf. Inform., Commun. Technol. (ICICT)*. 655–664.

[5] Hilal Asi, John Duchi, and Omid Javidbakht. 2022. Element Level Differential Privacy: The Right Granularity of Privacy. In *Proc. AAAI Workshop Priv.-Preserv. Artif. Intell. (PPAI)*.

[6] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc., Data Min. (KDD)* (Seattle, WA) *(AAAIWS'94)*. 359–370.

[7] Andrew J. Blumberg and Peter Eckersley. 2009. On Locational Privacy, and How to Avoid Losing it Forever. *Electron. Front. Found. (EFF)* (2009).

[8] Thomas Brinkhoff. 2002. A framework for generating network-based moving objects. *GeoInformatica* 6, 2 (2002), 153–180.

[9] Erik Buchholz, Alsharif Abuadbba, Shuo Wang, Surya Nepal, and Salil S Kanhere. 2022. Reconstruction Attack on Differential Private Trajectory Protection Mechanisms. *arXiv preprint* (2022).

[10] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. 2007. Efficient k-Anonymization Using Clustering Techniques. In *Proc. Adv. Databases: Concept, Syst., Appl. (DASFAA)*. 188–200.

[11] Sujin Cai, Xin Lyu, Xin Li, Duohan Ban, and Tao Zeng. 2021. A Trajectory Released Scheme for the Internet of Vehicles Based on Differential Privacy. *IEEE Trans. Intell. Transp. Syst.* (2021).

[12] Yang Cao and Masatoshi Yoshikawa. 2015. Differentially private real-time data release over infinite trajectory streams. In *Proc. IEEE Int. Conf. Mob. Data Manage. (MDM)*, Vol. 2. 68–73.

[13] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. 2017. Quantifying differential privacy under temporal correlations. In *Proc. IEEE Int. Conf. Data Eng. (ICDE)*. 821–832.

[14] Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, Ashwin Machanavajjhala, et al. 2009. Privacy-preserving data publishing. *Found., Trends Database* 2, 1–2 (2009), 1–167.

[15] Lei Chen and Raymond Ng. 2004. On the Marriage of Lp-Norms and Edit Distance. In *Proc. Int. Conf. Very Large Databases (VLDB)* (Toronto, Canada). 792–803.

[16] Lei Chen, M. Tamer Özsu, and Vincent Oria. 2005. Robust and Fast Similarity Search for Moving Object Trajectories. In *Proc. ACM SIGMOD Int. Conf. Manage. Data (MOD)* (Baltimore, Maryland). 491–502. https://doi.org/10.1145/1066157.1066213

[17] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *Proc. ACM Conf. Comput., Commun. Secur. (CCS)*. 638–649.

[18] Rui Chen, Benjamin Fung, and Bipin C Desai. 2011. Differentially private trajectory data publication. *arXiv preprint* (2011).

[19] Rui Chen, Benjamin Fung, Philip S Yu, and Bipin C Desai. 2014. Correlated network data publication via differential privacy. *VLDB J.* 23, 4 (2014), 653–676.

[20] Rui Chen, Benjamin CM Fung, Noman Mohammed, Bipin C Desai, and Ke Wang. 2013. Privacy-preserving trajectory data publishing by local suppression. *Inform. Sci.* 231 (2013), 83–97.

[21] Si Chen, Anmin Fu, Jian Shen, Shui Yu, Huaqun Wang, and Huaijiang Sun. 2020. RNN-DP: A new differential privacy scheme base on Recurrent Neural Network for Dynamic trajectory privacy protection. *J. Netw. Comput. Appl.* 168 (2020), 102736.

[22] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-Based Social Networks. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc., Data Min. (KDD)* (San Diego, California, USA). 1082–1090. https://doi.org/10.1145/2020408.2020579

[23] Chris Clifton and Tamir Tassa. 2013. On Syntactic Anonymity and Differential Privacy. *Trans. Data Priv.* 6, 2 (2013), 161–183.

[24] Chris Clifton and Tamir Tassa. 2013. On syntactic anonymity and differential privacy. In *Proc. IEEE Int. Conf. Data Eng. Workshop (ICDEW)*. 88–93.

[25] Graham Cormode. 2011. Personal privacy vs population privacy: learning to attack anonymization. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc., Data Min. (KDD)*. 1253–1261.

[26] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. 2021. Real-world trajectory sharing with local differential privacy. *arXiv preprint* (2021).

[27] Chenglong Dai, Dechang Pi, Stefanie I Becker, Jia Wu, Lin Cui, and Blake Johnson. 2020. CenEEGs: Valid EEG selection for classification. *ACM Trans. Knowl. Discov. Data* 14, 2 (2020), 1–25.

[28] Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, and Pierangela Samarati. 2012. Data Privacy: Definitions and Techniques. *Int. J. Uncertain., Fuzz., Knowl.-Based Syst.* 20 (2012), 793–818.

[29] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scient. Rep.* 3, 1 (2013), 1–5.

[30] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. 2008. Identification via Location-Profiling in GSM Networks. In *Proc. ACM Workshop Priv. Electron. Soc. (WPES)*. 23–32.

[31] Fatemeh Deldar and Mahdi Abadi. 2021. A differentially private location generalization approach to guarantee non-uniform privacy in moving objects databases. *Knowl.-Based Syst.* 225 (2021), 107084.

[32] Clemens Deußer, Steffen Passmann, and Thorsten Strufe. 2020. Browsing Unicity: On the Limits of Anonymizing Web Tracking Data. In *Proc. IEEE Symp. Secur., Priv. (SP)*. 777–790. https://doi.org/10.1109/SP40000.2020.00018

[33] Josep Domingo-Ferrer and Rolando Trujillo-Rasua. 2012. Microaggregation- and permutation-based anonymization of movement data. *Inform. Sci.* 208 (2012), 55–80. https://doi.org/10.1016/j.ins.2012.04.015

[34] Yulan Dong and Dechang Pi. 2018. Novel Privacy-preserving algorithm based on frequent path for trajectory data publishing. *Knowl.-Based Syst.* 148 (2018), 55–65.

[35] Cynthia Dwork. 2006. Differential privacy. In *Proc. Int. Colloq. Automata, Lang., Program. (ICALP)*. 1–12.

[36] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Found., Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.

[37] ECML/PKDD. 2015. *Taxi Trajectory Prediction (I)*. https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

[38] Stephan Escher, Markus Sontowski, Knut Berling, Stefan Köpsell, and Thorsten Strufe. 2021. How well can your car be tracked: Analysis of the European C-ITS pseudonym scheme. In *Proc. IEEE Veh. Technol. Conf. (VTC-Spring)*. 1–6.

[39] European Commission. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). *OJ L 119* (2016).

[40] Ziquan Fang, Yuntao Du, Xinjun Zhu, Lu Chen, Yunjun Gao, and Christian S. Jensen. 2021. ST2Vec: Spatio-Temporal Trajectory Similarity Learning in Road Networks. *Comput. Res. Repos. (arXiv CoRR)* abs/2112.09339 (2021). arXiv:2112.09339 https://arxiv.org/abs/2112.09339

[41] Marco Fiore, Panagiota Katsikouli, Elli Zavou, Mathieu Cunche, Françoise Fessant, Dominique Le Hello, Ulrich Aïvodji, Baptiste Olivier, Tony Quertier, and Razvan Stanica. 2020. Privacy in Trajectory Micro-Data Publishing: A Survey. *Trans. Data Priv.* 3 (2020).

[42] Lorenzo Franceschi-Bicchierai. 2015. Redditor cracks anonymous data trove to pinpoint Muslim cab drivers. *Mashable* (2015). https://mashable.com/archive/redditor-muslim-cab-drivers

[43] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *Proc. Conf. USENIX Secur. Symp.* (San Diego, CA). 17–32.

[44] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. 2011. Evaluating the Privacy Risk of Location-Based Services. In *Proc. Int. Financ. Cryptogr., Data Secur. (FC)*, Vol. 7035.

[45] Benjamin Fung, ke Wang, Rui Chen, and Philip Yu. 2010. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv.* 42 (2010). https://doi.org/10.1145/1749603.1749605

[46] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2010. Show Me How You Move and I Will Tell You Who You Are. In *Proc. ACM SIGSPATIAL Int. Workshop Secur., Priv. GIS & LBS (SPRINGL)*, Vol. 4. 34–41. https://doi.org/10.1145/1868470.1868479

[47] Seeta Peña Gangadharan. 2013. How can big data be used for social good? *The Guardian* (2013). https://www.theguardian.com/sustainable-business/how-can-big-data-social-good accessed on 2021-01-18.

[48] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. 2013. Exploiting Innocuous Activity for Correlating Users across Sites. In *Proc. ACM Int. WWW Conf.* 447–458.

[49] Marco Gramaglia, Marco Fiore, Angelo Furno, and Razvan Stanica. 2021. GLOVE: Towards Privacy-Preserving Publishing of Record-Level-Truthful Mobile Phone Trajectories. *ACM/IMS Trans. Data Sci.* 2, 3, Article 21 (2021), 36 pages. https://doi.org/10.1145/3451178

[50] Marco Gramaglia, Marco Fiore, Alberto Tarable, and Albert Banchs. 2017. Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories. In *Proc. Joint Conf. IEEE Comput., Commun. Soc. (INFOCOM)*. 1–9.

[51] David Gritten. 2022. Strava app flaw revealed runs of Israeli officials at secret bases. *BBC* (2022). https://www.bbc.com/news/world-middle-east-61879383

[52] Marco Gruteser and Dirk Grunwald. 2003. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. ACM Int. Conf. Mob. Syst., Appl., Serv. (MobiSys)* (San Francisco, California). 31–42. https://doi.org/10.1145/1066116.1189037

[53] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. 2018. Differentially private and utility preserving publication of trajectory data. *IEEE Trans. Mob. Comput.* 18, 10 (2018), 2315–2329.

[54] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, Lei Yu, and Wenqi Wei. 2018. Utility-aware synthesis of differentially private and attack-resilient location traces. In *Proc. ACM SIGSAC Conf. Comput., Commun. Secur. (CCS)*. 196–211.

[55] Qilong Han, Zuobin Xiong, and Kejia Zhang. 2018. Research on trajectory data releasing method via differential privacy based on spatial partition. *Security, Commun. Netw.* (2018).

[56] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M Procopiuc, and Divesh Srivastava. 2015. DPT: differentially private trajectory synthesis using hierarchical reference systems. *VLDB J.* 8, 11 (2015), 1154–1165.

[57] Alex Hern. 2018. Fitness tracking app Strava gives away location of secret US army bases. *The Guardian* (2018). https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

[58] Jingyu Hua, Yue Gao, and Sheng Zhong. 2015. Differentially private publication of general time-serial trajectory data. In *Proc. Joint Conf. IEEE Comput., Commun. Soc. (INFOCOM)*. 549–557.

[59] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric S. Nordholt, Keith Spicer, and Peter-Paul de Wolf. 2012. *Statistical Disclosure Control.* Wiley.

[60] Jung-Rae Hwang, Hye-Young Kang, and Ki-Joune Li. 2005. Spatio-temporal Similarity Analysis Between Trajectories on Road Networks. In *Proc. Perspect. Concept. Model. (ER)*. Berlin, Heidelberg, 280–289.

[61] Hongbo Jiang, Jie Li, Ping Zhao, Fanzi Zeng, Zhu Xiao, and Arun Iyengar. 2021. Location Privacy-Preserving Mechanisms in Location-Based Services: A Comprehensive Survey. *ACM Comput. Surv.* 54, 1, Article 4 (jan 2021), 36 pages.

[62] Fengmei Jin, Wen Hua, Matteo Francia, Pingfu Chao, Maria Orlowska, and Xiaofang Zhou. 2021. A Survey and Experimental Study on Privacy-Preserving Trajectory Data Publishing. *TechRxiv preprint* (2021). https://doi.org/10.36227/techrxiv.13655597.v1

[63] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[64] Shiva P Kasiviswanathan and Adam Smith. 2014. On the 'semantics' of differential privacy: A Bayesian formulation. *J. Priv., Confid.* 6, 1 (2014).

[65] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially Private Event Sequences over Infinite Streams. *VLDB J.* 7, 12 (2014), 1155–1166. https://doi.org/10.14778/2732977.2732989

[66] Eamonn Keogh and Chotirat Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowl., Inform. Syst.* 7 (2005), 358–386. https://doi.org/10.1007/s10115-004-0154-9

[67] Eamonn J. Keogh and Michael J. Pazzani. 2000. Scaling up dynamic time warping for datamining applications. In *Proc. ACM SIGKDD Int. Conf. Knowl. Disc., Data Min. (KDD)*.

[68] Daniel Kifer, John M Abowd, Robert Ashmead, Ryan Cumings-Menon, Philip Leclerc, Ashwin Machanavajjhala, William Sexton, and Pavel Zhuravlev. 2022. Bayesian and Frequentist Semantics for Common Variations of Differential Privacy: Applications to the 2020 Census. *arXiv preprint* (2022).

[69] Daniel Kifer and Ashwin Machanavajjhala. 2011. No free lunch in data privacy. In *Proc. ACM SIGMOD Int. Conf. Manage. Data (MOD)*. 193–204.

[70] Xiangjie Kong, Menglin Li, Kai Ma, Kaiqi Tian, Mengyuan Wang, Zhaolong Ning, and Feng Xia. 2018. Big trajectory data: A survey of applications and services. *IEEE Access* 6 (2018), 58295–58306.

[71] Bruno C. Leal, Israel C. Vidal, Felipe T. Brito, Juvêncio S. Nobre, and Javam C. Machado. 2018. δ-DOCA: Achieving privacy in data streams. In *Proc. Int. Workshop Data Priv. Manage. (DPM)*, Vol. 11025. 279–295.

[72] Meng Li, Liehuang Zhu, Zijian Zhang, and Rixin Xu. 2017. Achieving differential privacy of trajectory data publishing in participatory sensing. *Inform. Sci.* 400 (2017), 1–13.

[73] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. IEEE Int. Conf. Data Eng. (ICDE)*. 106–115.

[74] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. 2016. *Differential Privacy: From Theory to Practice.* Morgan & Claypool.

[75] James J. Little and Zhe Gu. 2001. Video retrieval by spatial and temporal structure of trajectories. In *Proc. SPIE, Storage, Retrieval Media Databases*, Vol. 4315. 545 – 552. https://doi.org/10.1117/12.410966

[76] Bingyu Liu, Shangyu Xie, Han Wang, Yuan Hong, Xuegang Ban, and Meisam Mohammady. 2021. VTDP: Privately Sanitizing Fine-Grained Vehicle Trajectory Data With Boosted Utility. *IEEE Trans. Depend., Secure Comput.* 18, 6 (2021), 2643–2657. https://doi.org/10.1109/TDSC.2019.2960336

[77] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. 2016. Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples. In *Proc. Symp. Netw. Distrib. Syst. Secur. (NDSS)*, Vol. 16. 21–24.

[78] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2020. A Survey on Deep Learning for Human Mobility. *arXiv preprint* (2020).

[79] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-Diversity: Privacy beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data* 1, 1 (2007), 3–es.

[80] Nehal Magdy, Mahmoud Sakr, Tamer Abdelkader, and Khaled Elbahnasy. 2015. Review on trajectory similarity measures. In *Proc. IEEE Int. Conf. Intell. Comput., Inform. Syst. (ICICIS)*. https://doi.org/10.1109/IntelCIS.2015.7397286

[81] Mohamed Maouche, Sonia Ben Mokhtar, and Sara Bouchenak. 2017. AP-Attack: A Novel User Re-Identification Attack On Mobility Datasets. In *Proc. EAI Int. Conf. Mob., Ubiquitous Syst.: Comput., Netw., Serv. (MobiQuitous)*. Association for Computing Machinery, New York, NY, USA, 48–57. https://doi.org/10.1145/3144457.3144494

[82] Pierre-François Marteau. 2009. Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *IEEE Trans. Pattern Anal., Mach. Intell.* 31, 2 (2009), 306–318. https://doi.org/10.1109/TPAMI.2008.76

[83] Ronaldo Mello, Vania Bogorny, Luis Alvares, Luiz Santana, Carlos Ferrero, Angelo Augusto Frozza, Geomar Schreiner, and Chiara Renso. 2019. MASTER: A multiple aspect view on trajectories. *Trans. GIS* (2019). https://doi.org/10.1111/tgis.12526

[84] Anna Monreale, Gennady Andrienko, Natalia Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. 2010. Movement Data Anonymity through Generalization. *Trans. Data Priv.* 3 (2010), 91–121.

[85] Anna Monreale, Roberto Trasarti, Dino Pedreschi, Chiara Renso, and Vania Bogorny. 2011. C-safety: A framework for the anonymization of semantic trajectories. *Trans. Data Priv.* 4 (2011), 73–101.

[86] Mehmet Nergiz, Maurizio Atzori, Yücel Saygın, and Barış Güç. 2009. Towards Trajectory Anonymization: A Generalization-Based Approach. *Trans. Data Priv.* 2 (2009), 47–75.

[87] OpenStreetMap contributors. 2022. Planet dump retrieved from https://planet.osm.org.

[88] Shira Ovide. 2020. Just Collect Less Data, Period. *New York Times* (2020). https://www.nytimes.com/2020/07/15/technology/just-collect-less-data-period.html accessed on 2021-01-18.

[89] Ruggero Pensa, Anna Monreale, Fabio Pinelli, and Dino Pedreschi. 2008. Pattern-Preserving k-Anonymization of Sequences and its Application to Mobility Data Mining. In *Proc. Int. Workshop Priv. Locat.-Based Appl. (PiLBA)*, Vol. 397.

[90] Tarlis Tortelli Portela, Francisco Vicenzi, and Vania Bogorny. 2019. Trajectory Data Privacy: Research Challenges and Opportunities. In *Proc. Braz. Symp. GeoInf. (GEOINFO)*.

[91] Giorgos Poulis, Spiros Skiadopoulos, Grigorios Loukides, and Aris Gkoulalas-Divanis. 2014. Apriori-based algorithms for k^m-anonymizing trajectory data. *Trans. Data Priv.* 7 (2014), 165–194.

[92] Vincent Primault, Antoine Boutet, Sonia Ben Mokhtar, and Lionel Brunie. 2018. The Long Road to Computational Location Privacy: A Survey. *Comput. Res. Repos. (arXiv CoRR)* (2018).

[93] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. *Comput. Res. Repos. (arXiv CoRR)* (2017). http://arxiv.org/abs/1708.06145

[94] Luca Rossi and Mirco Musolesi. 2014. It's the Way You Check-in: Identifying Users in Location-Based Social Networks. In *Proc. ACM Conf. Online Social Netw. (COSN)*. 215–226.

[95] Pierangela Samarati. 2001. Protecting respondents identities in microdata release. *IEEE Trans. Knowl. Data Eng.* 13, 6 (2001), 1010–1027. https://doi.org/10.1109/69.971193

[96] Pierangela Samarati and Latanya Sweeney. 1998. *Protecting privacy when disclosing information: k-Anonymity and its enforcement through generalization*

and suppression. Tech. Rep. SRI Int.
[97] Dongxu Shao, Kaifeng Jiang, Thomas Kister, Stephane Bressan, and K.-L Tan. 2013. Publishing Trajectory with Differential Privacy: A Priori vs. A Posteriori Sampling Mechanisms. In *Proc. Int. Conf. Database, Expert Syst. Appl. (DEXA) (Lecture Notes Comput. Sci. (LNCS), Vol. 8055).* 357–365.
[98] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of Predictability in Human Mobility. *Science* 327 (2010), 1018–21. https://doi.org/10.1126/science.1177170
[99] Jordi Soria-Comas and Josep Domingo-Ferrer. 2016. Big Data Privacy: Challenges to Privacy Principles and Models. *Data Sci. Eng.* 1, 1 (2016), 21–28.
[100] Roniel S. De Sousa, Azzedine Boukerche, and Antonio A. F. Loureiro. 2020. Vehicle Trajectory Similarity: Models, Methods, and Applications. *ACM Comput. Surv.* 53, 5, Article 94 (sep 2020), 32 pages. https://doi.org/10.1145/3406096
[101] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2021. Synthetic Data–Anonymisation Groundhog Day. *arXiv preprint* (2021).
[102] Kaixin Sui, Youjian Zhao, Dapeng Liu, Minghua Ma, Lei Xu, Li Zimu, and Dan Pei. 2016. Your trajectory privacy can be breached even if you walk in groups. In *Proc. IEEE/ACM Int. Symp. Qual. Serv. (IWQoS).* 1–6.
[103] Latanya Sweeney. 2002. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *Int. J. Uncertain., Fuzz., Knowl.-Based Syst.* 10, 5 (2002), 571–588. https://doi.org/10.1142/S021848850200165X
[104] Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. 2021. A comparative analysis of trajectory similarity measures. *GIScience, Remote Sens.* 58, 5 (2021), 643–669. https://doi.org/10.1080/15481603.2021.1908927
[105] Ben Tarnoff. 2018. Big data for the people: It's time to take it back from our tech overlords. *The Guardian* (2018). https://www.theguardian.com/technology/2018/mar/14/tech-big-data-capitalism-give-wealth-back-to-people accessed on 2021-01-18.
[106] Anthony Tockar. 2014. Riding with the Stars: Passenger Privacy in the NYC Taxicab Dataset. https://agkn.wordpress.com/author/atockar/
[107] Kevin Toohey and Matt Duckham. 2015. Trajectory similarity measures. *ACM Spec. Interest Group Spatial Inform. (SIGSPATIAL Special)* 7 (2015), 43–50. https://doi.org/10.1145/2782759.2782767
[108] J.K. Trotter. 2014. Public NYC Taxicab Database Lets You See How Celebrities Tip. *Gawker* (2014). https://www.gawker.com/the-public-nyc-taxicab-database-that-accidentally-track-1646724546
[109] Zhen Tu, Kai Zhao, Fengli Xu, Yong Li, Li Su, and Depeng Jin. 2019. Protecting Trajectory From Semantic Attack Considering k-Anonymity, l-Diversity, and t-Closeness. *IEEE Trans. Netw., Serv. Manage.* 16, 1 (2019), 264–278. https://doi.org/10.1109/TNSM.2018.2877790
[110] Michail Vlachos, Dimitrios Gunopulos, and George Kollios. 2002. Discovering similar multidimensional trajectories. In *Proc. IEEE Int. Conf. Data Eng. (ICDE).* 673–684.
[111] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. 2022. L-SRR: Local Differential Privacy for Location-Based Services with Staircase Randomized Response. In *Proc. ACM SIGSAC Conf. Comput., Commun. Secur. (CCS).* 2809–2823. https://doi.org/10.1145/3548606.3560636
[112] Haozhou Wang, Han Su, Kai Zheng, Shazia Sadiq, and Xiaofang Zhou. 2013. An Effectiveness Study on Trajectory Similarity Measures. In *Proc. Australas. Database Conf. (ADC)* (Adelaide, Australia). 13–22.
[113] Hao Wang, Zhengquan Xu, Shan Jia, Ying Xia, and Xu Zhang. 2021. Why current differential privacy schemes are inapplicable for correlated data publishing? *World Wide Web* 24 (2021), 1–23.
[114] Nana Wang and Mohan S Kankanhalli. 2020. Protecting sensitive place visits in privacy-preserving trajectory publishing. *Comput., Secur.* 97 (2020), 101949.
[115] Sheng Wang, Zhifeng Bao, J. Culpepper, Xie Zizhe, Qizhi Liu, and Xiaolin Qin. 2018. Torch: A Search Engine for Trajectory Data. In *Proc. ACM SIGIR Conf. Res., Develop. Inform. Retrieval.* 535–544. https://doi.org/10.1145/3209978.3209989
[116] Weiya Wang, Geng Yang, Lin Bao, Ke Ma, Hao Zhou, and Yunlu Bai. 2021. Travel Trajectory Frequent Pattern Mining Based on Differential Privacy Protection. *Wirel. Commun., Mob. Comput.* (2021).
[117] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. 2012. A classification of location privacy attacks and approaches. *Pers., Ubiquitous Comput.* 18 (2012), 163–175.
[118] Wikipedia. 2022. Metric (mathematics) — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Metric%20(mathematics)&oldid=1071302350. [Online; accessed 14-February-2022].
[119] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. 2020. A Comprehensive Survey on Local Differential Privacy. *Security, Commun. Netw.* (2020), 29. https://doi.org/10.1155/2020/8829523
[120] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data. In *Proc. ACM Int. WWW Conf.* (Perth, Australia). 1241–1250. https://doi.org/10.1145/3038912.3052620
[121] Bin Yang, Issei Sato, and Hiroshi Nakagawa. 2015. Bayesian differential privacy on correlated data. In *Proc. ACM SIGMOD Int. Conf. Manage. Data (MOD).* 747–762.
[122] Yuqing Yang, Jianghui Cai, Haifeng Yang, Jifu Zhang, and Xujun Zhao. 2020. TAD: A trajectory clustering algorithm based on spatial-temporal density analysis. *Expert Syst. Appl.* 139 (2020), 112846.
[123] Haitao Yuan and Guoliang Li. 2019. Distributed In-memory Trajectory Similarity Search and Join on Road Network. In *Proc. IEEE Int. Conf. Data Eng. (ICDE).* 1262–1273. https://doi.org/10.1109/ICDE.2019.00115
[124] Shuilian Yuan, Dechang Pi, Xiaodong Zhao, and Meng Xu. 2021. Differential privacy trajectory data protection scheme based on R-tree. *Expert Syst. Appl.* 182 (2021), 115215.
[125] Bin Zan, Zhanbo Sun, Macro Gruteser, and Xuegang Ban. 2013. Linking Anonymous Location Traces through Driving Characteristics. In *Proc. ACM Conf. Data, Appl. Secur., Priv. (CODASPY).* 293–300.
[126] Hui Zang and Jean Bolot. 2011. Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. In *Proc. ACM Annual Int. Conf. Mob. Comput., Netw. (MobiCom).* 145–156.
[127] Jianzhe Zhao, Jie Mei, Stan Matwin, Yukai Su, and Yuancheng Yang. 2020. Risk-aware individual trajectory data publishing with differential privacy. *IEEE Access* 9 (2020), 7421–7438.
[128] Xiaodong Zhao, Yulan Dong, and Dechang Pi. 2019. Novel trajectory data publishing method under differential privacy. *Expert Syst. Appl.* 138 (2019), 112791.
[129] Xiaodong Zhao, Dechang Pi, and Junfu Chen. 2020. Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Syst. Appl.* 149 (2020), 113241.
[130] Xiaodong Zhao, Dechang Pi, and Junfu Chen. 2020. Novel trajectory privacy-preserving method based on prefix tree using differential privacy. *Knowl.-Based Syst.* 198 (2020), 105940.
[131] Changqing Zhou, Dan Frankowski, Pamela Ludford Finnerty, Shashi Shekhar, and Loren Terveen. 2004. Discovering personal gazetteers: An interactive clustering approach. In *Proc. ACM Int. Symp. Adv. Geogr. Inform. Syst. (GIS).* 266–273. https://doi.org/10.1145/1032222.1032261

A SIMILARITY MEASURES

Similarity measures, which include *distance functions*, output a value quantifying how similar two input trajectories are. Their use in trajectory protection mechanisms is usually one of the following two. First, they can be used within the privacy mechanism to, for example, determine which trajectories should be clustered and merged, considering it preferable to cluster the most similar ones. Secondly, they are also closely related to utility metrics. High similarity can be an indicator of high utility after sanitization.

Here we provide a classification of similarity measures. Compilations of trajectory similarity functions have already been studied and compared [80, 100, 104, 107, 112], but here we are limiting ourselves to those that lend themselves to trajectory privacy. We also provide a few types not considered by the aforementioned surveys, specifically the similarity measures that split dimension- and point-wise.

The first separation we consider is the distinction between spatial and spatio-temporal similarity measures (Subsections A.1 and A.2, and columns 2 and 1 in Figure 4, respectively). We then explore the similarity measures defined over road networks (Subsection A.3). Finally, we discuss similarity measures that can be split dimension- and point-wise (Subsection A.4). The latter defines the measures for each dimension independently (spatial, temporal, categorical, etc.) and then "adds" the values up. The main possible dimensional measures are included in columns 2–4 in Figure 4, where only the spatial and categorical ones have a history of being used as independent measures. The point-wise division of trajectories is represented row-wise in Figure 4.

We further recompile the similarity measures in Table 3. The three principal properties of interest when looking at these measures is (i) whether they allow comparing trajectories of different length; (ii) whether they allow for time shifting, that is, expanding

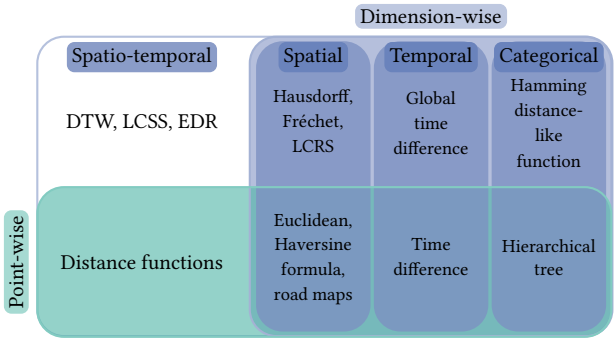

Figure 4: Overview of similarity measures classified according to whether they are split dimension- and/or point-wise, with a few examples. Dimension-wise–split similarity measures consider at least two dimensions in their computation, with only spatial and categorical similarity measures having been used as independent metrics.

and contracting the trajectories in time to better match one another (see Figure 5); and (iii) whether they are greatly affected by outlying locations (i.e., *robust to noise*). The effect of outliers or "noisy" locations in similarity measures that are not robust to noise can provide counterintuitive examples, e.g., a pair of trajectories equal in all-but-one location, which is far away, can be seen as completely different under these measures.

Furthermore, while the complexity of computing the measures is a relevant issue for their application, we focus on their properties concerning privacy and utility. Note that as a consequence of their complex structure, some measures are not *metrics* in the mathematical sense [118], but this is a condition only exploited to speed up their retrieval process [15, 82].

A.1 Spatial Similarity Measures

The variation of the *Euclidean distance* for data sequences is the most well-known metric. It calculates the distance between two trajectories of the same length by computing the physical distance between every set of points. Analogous functions with any other L^p-norm can also be defined, as well as variations that use the root square mean error. Even though the Euclidean distance has the lowest computational cost, it is considered a very brittle similarity measure in the literature [66]. It does not allow comparison of trajectories of different lengths, is greatly affected by outlying locations and is unable to time shift. The Euclidean distance, or variations thereof, are used as utility metrics in [11, 97, 129].

Two other measures in this subcategory are the *Hausdorff* and *Fréchet distances*, the former being previously used in trajectory privacy [21, 58, 72, 78]. Both distances allow different-length comparisons, but they are ultimately decided by the physical distance of a unique pair of points, which not only implies that outliers have a huge effect on it but that the rest of the information is also ultimately neglected.

Note that spatial measures should not be used alone, since measures must not rely only on the spatial coordinate, but must consider the rest of the trajectories' dimensions to guarantee both strong privacy and high utility. For example, all these measures will output that two trajectories running at different time intervals over the same route are the same. Nevertheless, spatial similarity measures can be used in dimension-wise splitting measures (see Subsection A.4).

A.2 Spatio-Temporal Similarity Measures

Another important family of measures is based on time series and edit distance. These measures consider time and allow for *local time shifting* by aligning locations through a minimal number of delete, insert and match operations. They are also able to compare trajectories of different lengths.

DTW [6, 66] is a similarity measure that recursively finds similar patterns between trajectories, and aligns them by locally contracting and expanding their temporal dimension. This allows single locations in one trajectory to be aligned with multiple of the other (see Figure 5). More specifically, the DTW distance value is defined as the minimum of the sum (or any L^p-norm) of the distances between all alignments.

Figure 5: Comparison between Euclidean distance and DTW that shows the local-time-shifting method [66]. In each figure, the top and bottom trajectories are compared. The vertical and horizontal axis of each trajectory corresponds to position and time, respectively, and the thin lines show the alignments made between both trajectories.

Essentially, there are two types of alignments: aligning the first pair of unaligned locations in both trajectories (called *match*) or aligning the first unaligned location of one with the last aligned location of the other (called *insertion* and *deletion*). Locations cannot be skipped in the alignment process, meaning outliers still affect the DTW value.

The basic definition of DTW does not have a bound, limit, or penalty associated with the number of insertions or deletions, although variations that do exist [6, 66]. For example, since time shifting is not bounded (one can shift locations as easily by five minutes as by five days), Marteau [82] defines a similar metric called TWED. They introduce a *stiffness* parameter (it controls the scope of time shifting, i.e., we can consider DTW to have no stiffness, and the Euclidean distance to have "infinite" stiffness), and also adds penalties for deletion. Consequently, TWED introduces a gap penalty when it aligns samples for which the index values are too far off to favor the alignment of those with close indexes.

Multiple variations of DTW exist. Anantasech and Ratanamahatana [4] define EWDTW, which follows a more intuitive approach and solves an overcompression phenomenon sometimes present in DTW. PDTW [67] provides similar results to DTW while also being effective [112] against outliers by cutting and clustering trajectories into pieces. Little and Gu propose a separation of spatial and temporal information by splitting trajectories into path and speed curves and define a measure that applies a DTW-based approach to both.

LCSS [110] is a similar measure proposed to overcome the effect of outlying locations. The LCSS value corresponds to the length of the largest pair of subtrajectories such that each pair of locations

Similarity measure	Checks similarity by comparing	Road network	Dimensions	Can compare different lengths	Allows local time shifting	Robust to noise	Is a metric	Computational cost
Euclidean distance (and L^p-norms)	Points		S		–		✓	$O(n)$
Hausdorff distance	Shape		S	✓	–		✓	$O(n \log(n))$
Fréchet distance	Shape		S	✓	–		✓	$O(nm \log(nm))$
Dynamic time warping (DTW) [6, 66]	Points (time series)		S-T	✓	✓			$O(nm)$
Time warp edit distance (TWED) [82]	Points (time series)		S-T	✓	✓		✓	$O(nm)$
Enhanced weighted DTW (EWDTW) [4]	Points (time series)		S-T	✓	✓			$O(nm)$
Piecewise DTW (PDTW) [67]	Points (time series)		S-T	✓	✓	✓		$O(NM)$ N, M final lengths
Little and Gu [75]	Movement speed and path		S-T	✓	✓			$O(nm)$
Longest common subsequences (LCSS) [110]	Points (time series)		S-T	✓	✓	✓		$O(nm)$
Edit distance on real sequences (EDR) [16]	Points (time series)		S-T	✓	✓	✓		$O(nm)$
Linear spatio-temporal distance (LSTD) [2]	Points (time series)		S-T	✓	✓	✓		$O(n+m)$
Edit distance with real penalty (ERP) [15]	Points (time series)		S-T	✓	✓		✓	$O(nm)$
Hwang et al. [60]	Time at intersections	✓	S-T	✓				$O(nm\|P\|)$ P={intersections}
Longest overlapping road segment (LORS) [115]	Road segments	✓	S	✓	–	✓		$O(nm)$
Longest common road segment (LCRS) [123]	Road segments	✓	S	✓	–	✓		$O(nm)$
Spatio-temporo-categorical distance [26]	Points		S-T-C		✓		✓	$O(nm)$

Table 3: Comparison between similarity measures (based on [80]). For dimensions, "S", "T", and "C" stands for spatial, temporal, and categorical, respectively; and, for computational cost, n and m correspond to the length of the two compared trajectories.

is at a bounded spatio-temporal distance away from each other (the higher the value, the higher the similarity); and it is then normalized by dividing it by the length of the shortest trajectory. In addition to local time shifting, LCSS allows locations to remain unaligned, ensuring robustness to noise while also providing a more intuitive notion of similarity by giving more weight to similar subsequences [110]. However, LCSS allows for gaps of any size, which can cause inaccuracies [16].

Akin to these measures, we find the subfamily based on *edit distance*, which intuitively is the function that counts the minimum number of edits needed to change one trajectory into the other. For example, EDR [16] counts the number of insert, delete and replace operations needed to do so. This measure is more exact than LCSS while still being robust to noise, as it assigns penalties to the gaps between trajectories according to their length [16]. The measure also can compare different length trajectories, but this can inflate the edit distance as each location needs to be edited in or out [107]. Abul et al. [2] use EDR in their trajectory-anonymization mechanism, while also defining a variation, LSTD, which consists of a similarity function with the same computational cost as the Euclidean distance, but with all the benefits of EDR. Another popular measure is ERP [15], a non–robust-to-noise metric that uses real distances between points as the penalty to time shifting [80] but does require normalization to a reference point, which minimizes its uses as a utility metric.

A.3 Similarity Measures over Road Networks

Another type of similarity measures includes those that consider trajectories in road maps instead of over empty Euclidean spaces. Road networks are viewed as directed graphs with edges and nodes representing roads and their intersections, respectively. Trajectories are thus seen as directed walks over this graph [100]. Since privacy mechanisms over road trajectories can be influenced by the similarity measure chosen, Hwang et al. [60] argue that using one not defined over road networks is inappropriate. We explore a few examples in the literature.

The first instance of similarity measure over road networks is used for similar-trajectories searching in data sets [60]. Given a specific road intersection, the authors define similarity as whether both trajectories cross it or not. A refined distance function is also given, which outputs the L^p-norm of the time difference at a preselected set of intersections. However, this preselection of locations limits the use of the similarity measure, since it needs to be defined in a case-by-case scenario, and the whole measure outputs ∞ if a point not belonging to one of the trajectories is selected.

LORS [115] and LCRS [123] are two variations of LCSS for road networks. These, instead of measuring the length of the largest subsequence of close spatio-temporal points, measure the physical length of matching roads. The difference between LORS and LCRS is that the latter does include a normalizing step, consisting of dividing the result by the number of points in the union of both trajectories (Jaccard similarity coefficient). LORS and LCRS are proven to be more effective than DTW, LCSS, EDR, and ERP, with LCRS working a little better than LORS [123]. The majority of the advantages and deficiencies of LCSS are inherited by these, but they do not consider the temporal domain in their computation unlike LCSS (although a variation of LCRS to include it can easily be implemented [123]).

Finally, in this subcategory, we also find similarity measures based on deep learning, such as ST2Vec [40].

A.4 Splitting Similarity Measures Dimension- and Point-Wise

We now enter to study the similarity measures that split trajectories dimension- and/or point-wise. None of the similarity measures we reviewed make any split, with exception of Little and Gu's measure [75] and the Euclidean distance, the last of which is a 1-dimensional point-wise–split similarity measure.

Dimension-wise. It is possible to split the measure dimension-wise, that is, defining independently the spatial distance, the temporal distance, etc., and then "adding" them up. This split allows the definition of simple spatial or temporal functions while still taking into account all dimensions of the trajectory. A difficulty of this method is deciding how to "add" them since there are multiple non-trivial variations [33], such as weighted or simple L^p sums.

When considering the dimensions independently, there are not many more options for the temporal one other than a simple time difference. On the other hand, for the spatial dimension, there is a larger variety of metrics, such as the Hausdorff distance we introduced. For the categorical dimension, a measure resembling the Hamming distance could also be theoretically defined.

Point-wise. We can also define similarity measures over two trajectories of equal length point-by-point, i.e., we take the first points of each trajectory and compute the distance between them, then the second of each, and successively until done. The final distance is computed as the sum of these values. Note that this is highly useful when comparing algorithms that only permute trajectories (without addition or suppression) since the sanitized trajectory can be compared to the original to study its utility.

Dimension- and point-wise. The combination of both is also popular: for the spatial dimension, the point-wise Euclidean distance is the most frequently used metric, while the Haversine formula, which considers latitude and longitude coordinates, is used in large-scale trajectories where the earth's curvature should be taken into account, such as flight routes. Road maps can also be used to measure the distance between two points, which can be useful in certain scenarios. Categorical distance functions also exist for semantic trajectories: in [26], the authors define the difference in the semantic meaning between two locations using a 3-level hierarchy with pre-established values, as shown in Figure 6. This is a simple way of measuring potentially complex information while also considering semantic proximity.

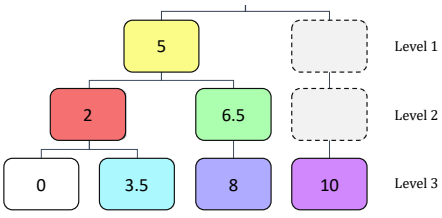

Figure 6: Categorical distance between the white element with the other levels [26]. A lower level implies a more specific semantic field. For example, red could be "Place to eat or drink", with white being "Bar" and blue "Restaurant". Values are assigned in an intuitive way (e.g., d_c(Bar, Restaurant) < d_c(Bar, Church)).

Actually, the authors of this paper ([26]) define a spatio-temporo-categorical distance as $d_w := \sqrt{d_s^2 + d_t^2 + d_c^2}$, where d_s is the Euclidean distance, d_t the time difference, and d_c the aforementioned categorical distance. This distance function is used in their privacy-protection process and proves its place as one of the most precise similarity measures for trajectory privacy.

B SYNTACTIC NOTIONS UNDER k-ANONYMITY AND ITS EXTENSIONS

In this section, we briefly review trajectory anonymization and sanitization with *syntactic* privacy notions. More precisely, we discuss privacy mechanisms under the notion of *k-anonymity* [95, 96] (i.e., is achieved when the information of any individual in the database is indistinguishable from $k-1$ others) and its extensions, such as *l-diversity* [79] and *t-closeness* [73], which are classical representatives in the field. Regarding trajectory data, several attempts have been made to translate or adapt these notions to better represent the sequential nature of trajectories. For example, a database is (k, δ)-*anonymous* [1] if for any trajectory, there exist $k-1$ other trajectories such that at every timestamp, the corresponding locations are in no more than $\delta/2$ distance away from each other. We can then place these k trajectories in a "cylinder" of radius δ. Other notions include *c-safety* [85], $(K, C)_L$*-privacy* [20], k^m*-anonymity* [91] and $k^{\tau,\epsilon}$*-anonymity* [50].

There exist three main general anonymization techniques to enforce syntactic privacy in trajectory data [90]: *suppression*, removing location samples or entire trajectories that cause privacy issues; *generalization*, making records indistinguishable from others by reducing the trajectories' precision or by grouping samples into larger ranges; and *(perturbative) masking*, which comprises a multitude of techniques including *data perturbation*, based on noise addition, location *merging* or *clustering*, or the creation of new entries by probabilistic *condensation*, just to name a few. Suppression and generalization techniques are also categorized into *non-perturbative masking* [59] since they preserve the truthfulness of data without distorting it, albeit losing information. Here, the use of the term "generalization" refers to its original definition [95, 103]. Nowadays, this term is also used to define the technique which additionally returns the generalized data to its original domain [84], mimicking the clustering techniques. For example, after generalization, we can define this step as a simple substitution of the region into a randomly chosen point from it. However, we classify this as a perturbative approach, since it creates distortions in the data. We will make this distinction between "clustering" and "generalization" to make clear if the approach is perturbative or not.

Some anonymization algorithms combine several of the techniques mentioned above. For example, suppression is often employed with the others, such as with generalization [49, 85, 91, 109], clustering [1, 2, 34, 86], and other masking techniques [33, 89]. Nevertheless, we still find some proposals revolving around one technique, such as local suppression [20], generalization [91], and clustering [84].

The given techniques have been deeply studied, analyzed, and frequently criticized, and we condense the corresponding arguments below.

Suppression is specifically useful in combination with others, where it helps to remove hard-to-anonymize locations or trajectories [1], such as those which are isolated or correspond to only one user. However, if used by itself, it can drastically change the size of the sanitized database, thus leading to significant privacy losses.

Generalization can also be deficient if applied inadequately. Models that generalize only the spatial dimension of trajectories, such as in [34, 85], are susceptible to attacks on the others, as these may still hold sensitive information. Also, the way that the generalization "regions" are defined is crucial, since inappropriate choices can lead to significant information loss due to unnecessary generalization [10], or result in data that is susceptible to attacks using background knowledge. To make matters worse, these methods are ineffective for databases with long trajectories due to the *curse of dimensionality* [3, 32]. Generalization also works poorly by itself, as shown by two experiments conducted in [1, 49].

Regarding *masking techniques*, Gramaglia et al. [50] state that to preserve the truthfulness of data, one cannot rely on randomized, perturbed, permuted, or synthetic data, since the addition of fictitious data introduces unpredictable biases in the final sanitized data sets. Furthermore, this type of mechanism (e.g., [84, 86]) can also lead to the creation of impossible trajectories, with unreachable locations or geospatial inconsistencies.

Although syntactic notions can, in general, provide high-utility data when compared to DP, they present major problems in terms of privacy. These notions assume the attacker knows some background knowledge, i.e., which attributes of the database are known, and which are not. They are also susceptible to various well-known attacks (e.g., k-anonymity falls victim to attribute linkage attacks). These two shortcomings, together with the fact that they do not enjoy composability [99] nor post-processing properties, limit the application of syntactic technology to continuously protect trajectory data. Furthermore, data sets with sparse or short trajectories pose a substantial challenge for these privacy methods. In these cases, the data must be deleted or modified extensively, leading to an inevitable large utility loss.

C FORMAL PROOFS

In this section, we provide some propositions as support to our claims made in Section 6. In these results, we will use Dwork and Roth's definition of database [36], defined as a multiset drawn from $\mathcal{X}$, the universe of database rows (represented too by their histograms from $\mathbb{N}^{|\mathcal{X}|}$). To simplify notation, we use $\mathcal{D}$ to denote a set of finite databases.

Problems with noisy-counts algorithms. We first show that a meaningful DP mechanism cannot simply change the counts of the elements in the database. This is essentially what happens in the protection mechanisms of [124, 127–130], which prevents them from satisfying DP formally.

Proposition C.1. *Let $\mathcal{M}$ be a randomized algorithm whose domain is $\mathcal{D}$. Suppose $\mathcal{M}$ changes the counts of the rows of $D \in \mathcal{D}$ (where it is possible to change a positive count into 0, but not the other way around). If $\mathcal{M}$ is ε-DP, then $\mathcal{M}$ is the void algorithm (i.e., it outputs the empty set independently of the input).*

Proof. Let $\mathcal{M}$ be an ε-DP algorithm, as described in the statement. By definition, the output domain of $\mathcal{M}$ is a subset $\mathcal{S} \subseteq \mathcal{D}$.

Fix $D \in \mathcal{D}$. For every $x \in D$, denote $k_x < \infty$ as the number of times x appears in D and D_x as the database obtained after removing all elements x from D. For every $x \in D$, there exists a sequence of neighboring databases of $\mathcal{D}$:

$$D = D_0 \to D_1 \to \cdots \to D_{k_x-1} \to D_{k_x} = D_x,$$

i.e., D_{i-1} and D_i are neighboring for all $i \in \{1, \ldots, k_x\}$. Then, since $\mathcal{M}$ is ε-DP, we obtain for all measurable $S \subseteq \mathcal{S}$ and $x \in D$ that

$$\mathrm{P}\{\mathcal{M}(D) \in S\} \leq \mathrm{e}^{\varepsilon}\,\mathrm{P}\{\mathcal{M}(D_1) \in S\} \leq \mathrm{e}^{2\varepsilon}\,\mathrm{P}\{\mathcal{M}(D_2) \in S\} \leq$$
$$\leq \cdots \leq \mathrm{e}^{(k_x-1)\varepsilon}\,\mathrm{P}\{\mathcal{M}(D_{k_x-1}) \in S\} \leq \mathrm{e}^{k_x\varepsilon}\,\mathrm{P}\{\mathcal{M}(D_x) \in S\} = 0.$$

Let $S_D \subseteq \mathcal{S}$ be the set of all possible outputs of $\mathcal{M}(D)$. It is clear that $\mathrm{P}\{\mathcal{M}(D) \in S_D\} = 1$. Furthermore, S_D is contained in the discrete set $\{S \text{ multiset} \mid \text{for all } x \in S,\, x \in D\}$, and therefore S_D is discrete, and

$$\mathrm{P}\{\mathcal{M}(D) \in S_D\} = \sum_{s \in S_D} \mathrm{P}\{\mathcal{M}(D) = s\}.$$

For every non-empty $s \in S_D$, we select an element $x \in s$. By the previous inequalities, we obtain that

$$\mathrm{P}\{\mathcal{M}(D) = s\} \leq \mathrm{e}^{k_x\varepsilon}\,\mathrm{P}\{\mathcal{M}(D_x) = s\} = 0,$$

since $x \notin D_x$ and $x \in s$. Therefore,

$$1 = \mathrm{P}\{\mathcal{M}(D) \in S_D\} = \sum_{s \in S_D} \mathrm{P}\{\mathcal{M}(D) = s\} = \mathrm{P}\{\mathcal{M}(D) = \varnothing\}.$$

Since $\mathcal{M}(D)$ is a discrete random variable, it proves that it can only output the empty set. Then, we repeat the proof for every possible database $D \subseteq \mathcal{D}$, proving that $\mathcal{M}$ is the void algorithm. □

In general, a DP mechanism needs to be able to output any possible output independently of the database. We formalize this statement with the precise hypotheses in Propositions C.2 and C.3, which cover the bounded and unbounded scenarios of DP. Recall that in *unbounded* DP, two databases are neighboring if we obtain one from the other by adding or removing one element; and that in *bounded* DP, these are neighboring if we obtain them instead by replacing one element with another [74].

Proposition C.2. *Let $\mathcal{M}$ be a randomized algorithm that satisfies unbounded ε-DP, $\mathcal{D}$ its domain, and* $\mathrm{Range}(\mathcal{M})$ *the set of all possible outputs of $\mathcal{M}$. Then, given any measurable $S \subseteq \mathrm{Range}(\mathcal{M})$, if there exist $D \in \mathcal{D}$ such that $\mathrm{P}\{\mathcal{M}(D) \in S\} > 0$, it is also true for all other $D' \in \mathcal{D}$.*

Proof. Consider a measurable $S \subseteq \mathrm{Range}(\mathcal{M})$ such that there exist $D \in \mathcal{D}$ in a way that $\mathrm{P}\{\mathcal{M}(D) \in S\} > 0$. We then proceed by *reductio ad absurdum*: that is, we assume that there exists $D' \in \mathcal{D}$ such that $\mathrm{P}\{\mathcal{M}(D') \in S\} = 0$ and we will end in a contradiction.

Since we assume all databases are finite, there exists a finite sequence of neighboring databases from D to D' of length k. As in the proof of Proposition C.1, we obtain

$$\mathrm{P}\{\mathcal{M}(D) \in S\} \leq \mathrm{e}^{k\varepsilon}\,\mathrm{P}\{M(D') \in S\} = 0.$$

This contradicts that $\mathrm{P}\{\mathcal{M}(D) \in S\} > 0$. □

Proposition C.3. *Let $\mathcal{M}$ be a randomized mechanism that satisfies bounded ε-DP, $\mathcal{D}$ its domain, and* Range($\mathcal{M}$) *the set of all possible outputs of $\mathcal{M}$. Then, given any measurable $S \subseteq$ Range($\mathcal{M}$), if there exist $D \in \mathcal{D}$ such that $\mathrm{P}\{\mathcal{M}(D) \in S\} > 0$, it is also true for all other $D' \in \mathcal{D}$ such that $|D'| = |D|$.*

Proof. This proof is the same as that of Proposition C.2, but we must impose that $|D| = |D'|$ to ensure that there is a sequence of neighboring databases between D and D'. □

Problems with clustering algorithms. Another problem of the proposals [21, 55, 58, 72] presented in Section 6 was related to the application of an "exponential mechanism" without a formal proof of DP. We elaborate deeply here on this problem.

The exponential mechanism [36] selects the best element of a certain given set $\mathcal{R}$, the range of this mechanism. The best assignments for each database are chosen using a *score function u*, which associates scores to each element in the database: the higher the score, the higher its chances to be chosen. More formally, given $D \in \mathcal{D}$, the exponential mechanism outputs $r \in \mathcal{R}$ with probability proportional to $\exp\left(\varepsilon \frac{u(D,r)}{2\Delta u}\right)$, where $u : \mathcal{D} \times \mathcal{R} \longrightarrow \mathbb{R}$ is the aforementioned score function and

$$\Delta u \coloneqq \max_{r \in \mathcal{R}} \max_{\substack{\text{neighb.} \\ D, D'}} |u(D,r) - u(D',r)|$$

is its sensitivity.

In the mentioned proposals [21, 55, 58, 72], the score function is not well-defined, which results in a contradiction in the DP proof of the claimed exponential mechanism. In the original framework [58] (from which the others stem out), the exponential mechanism is used to output the centroids of the partitions of the location set at every timestamp i. In this work, the score function is defined as $u : \mathcal{D} \times \tau \longrightarrow \mathbb{R}$, with τ being the set of partitions of the locations set at time i of a specific database D. The previous expression is not well-defined, since τ depends on the chosen element $D \in \mathcal{D}$, and varies when changing to another D, as mentioned in the paper. As a direct consequence, Δu is not theoretically computable (even if fixing D, since the definition compares two different databases), and an exponential mechanism cannot be defined. Hence, one cannot claim the algorithm ensures DP via the exponential mechanism.

This error leads to some anomalies in the suggested proposal. First, the cluster size does not affect the privacy guaranteed: i.e., we can choose to partition into sets of size 1, which would simply be a mechanism outputting the original unmodified database, providing no privacy. Secondly, $u(D,r) \leq 1$ for all possible combinations, would imply that the absolute difference between any possible score function is at most 1. If the exponential mechanism were correctly applied, it would mean that changing the whole database has the same effect as changing one record, which is highly improbable.

Having explained why the mechanism is not the exponential mechanism, we discuss why it is not DP. We know that given two different sets, S and S', their sets of partitions into m groups, $\mathcal{P}_S^m$ and $\mathcal{P}_{S'}^m$, are disjoint. For example, consider $S = \{1,2,3\}$ and $S' = \{1,2\}$. The only partition of S' into two clusters is $P_{S'} = \{\{1\},\{2\}\}$, while for S we have $P_S^{(1)} = \{\{1,2\},\{3\}\}$, $P_S^{(2)} = \{\{1,3\},\{2\}\}$ or $P_S^{(3)} = \{\{2,3\},\{1\}\}$. It is then easy to see that $\mathcal{P}_S^2 \cap \mathcal{P}_{S'}^2 = \emptyset$.

More formally, consider two neighboring databases $D, D' \in \mathcal{D}$ and their respective location set at time i, Γ_i and Γ_i'. Let $\mathcal{P}, \mathcal{P}' \subseteq$ Range(M) be the set of all possible partitions of Γ_i and Γ_i', respectively, into m groups. As mentioned, $\mathcal{P} \cap \mathcal{P}' = \emptyset$, so

$$1 = \mathrm{P}\{\mathcal{M}(D) \in \mathcal{P}\} \leq \mathrm{e}^{\varepsilon}\, \mathrm{P}\{\mathcal{M}(D') \in \mathcal{P}\} = 0,$$

resulting in a contradiction with the definition of DP. As we already proved in Proposition C.2, if an output is possible for a database, it needs to be possible for all the remaining ones, which simply cannot happen if the range of outputs is data dependent.

Therefore, the privacy mechanisms in [21, 55, 58, 72] do not provide DP as they do not apply correct exponential mechanisms because their abstract range of outputs is completely dependent on the input database. Furthermore, we give a small attack example:

Example C.4. We propose a simple trajectory database consisting of three users, and we focus on any specific timestamp. We are working with a set of three locations $\Gamma = \{l_1 = (2,2), l_2 = (5,2), l_3 = (5,5)\}$. The mechanism $\mathcal{M}$ clusters databases into 2 and outputs its centroid according to the Euclidean distance (essentially, Hua et al.'s proposal [58] with $m = 2$).

Assume a strong attacker that knows all the values except their target l_1 (which we can do according to the definition of DP). We show that, with the released information, the attacker obtains the target data with total accuracy.

The attacker knows l_2 and l_3, and that the possible clusters that $\mathcal{M}$ can compute are $\mathcal{P} = \{P_1, P_2, P_3\}$ with $P_1 = \{\{l_1\},\{l_2,l_3\}\}$, $P_2 = \{\{l_1,l_2\},\{l_3\}\}$ and $P_3 = \{\{l_2\},\{l_1,l_3\}\}$. So, the attacker knows also a priori that the mechanism will output one of the following centroid sets: $C(P_1) = \{?,(5,3.5)\}$, $C(P_2) = \{?,(5,3)\}$, or $C(P_3) = \{(5,2),?\}$, with ? denoting the coordinates they cannot predict. Suppose the mechanism computes $\{\{l_2\},\{l_1,l_3\}\}$ (unknown to the attacker) and their centroids, and then releases $C = \{(5,2),(3.5,3.5)\}$. Now, the attacker can compare their computation with the released centroid set, and conclude that the only possible partition is P_3. Additionally, since $(3.5,3.5)$ is released and corresponds to the middle point between $l_1 = (x_1,y_1)$ and $l_3 = (x_3,y_3)$, the attacker can easily recover l_1:

$$(3.5,3.5) = \left(\frac{x_1+x_3}{2}, \frac{y_1+y_3}{2}\right) = \left(\frac{x_1+5}{2}, \frac{y_1+5}{2}\right).$$

Therefore,

$$l_1 = (x_1,y_1) = (2 \cdot 3.5 - 5, 2 \cdot 3.5 - 5) = (2,2),$$

which allows the attacker to reconstruct the original database in its entirety.