Counterbalance: A Declarative Tool for Privacy-Preserving Statistical Chart Authoring based on Differential Privacy

Xumeng Wang, Shuangcheng Jiao, and Chris Bryan, and Yining Wang

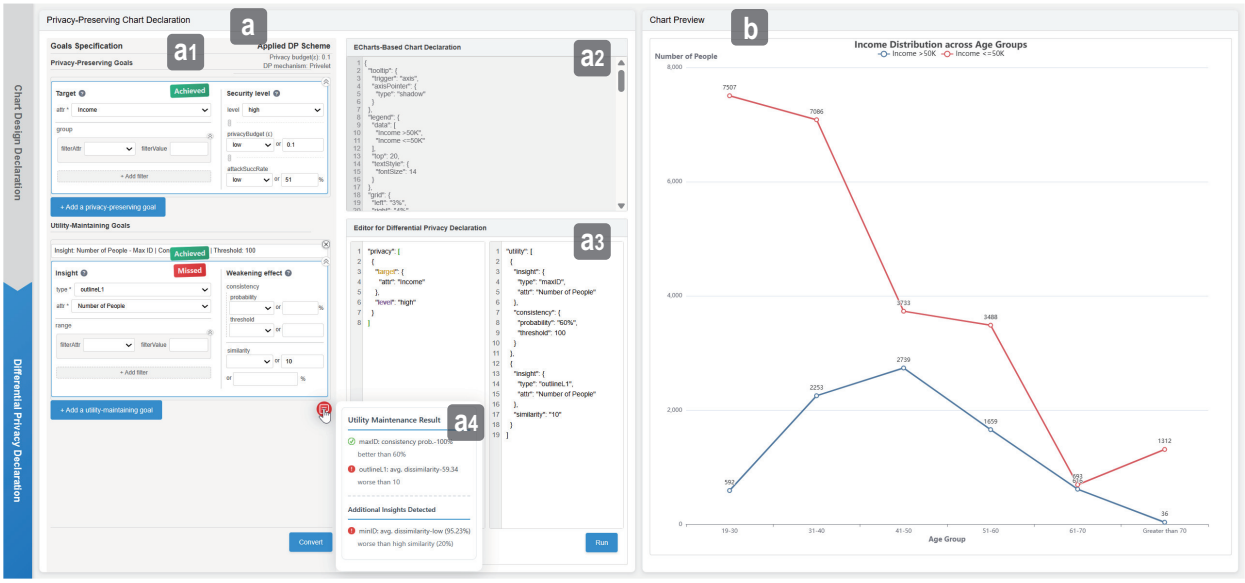

Figure 1: The interface of Counterbalance in the differential privacy declaration stage supports creating privacy-preserving charts based on declarations of expected privacy preservation. (a) The editor for differential privacy declaration view consists of (a1) panels for declaring goals for privacy preservation and utility maintenance, (a2) an ECharts-based editor displaying the base chart configuration defined in the first phase, (a3) a code editor showing editable code converted from declared goals, and (a4) completion report explaining how the applied DP scheme satisfies the declared requirements. (b) Chart view rendering the privacy-preserving visualization.

Abstract—Visualizations are widely used to communicate insights about data, and many authoring tools now exist to simplify the chart creation process. Unfortunately, when datasets contain sensitive information, such charts and tools can risk privacy exposure. For many dataset owners, it is a significant challenge to balance (i) effectively representing desired insights and (ii) ensuring privacy when visualizing their data. We investigate this by studying how to integrate privacy mechanisms into the chart authoring process. Based on a pre-study with domain experts, we design a compact-yet-inclusive declarative **grammar** for specifying differentially private visualizations. Using this, we develop Counterbalance, an authoring tool that supports creating privacy-preserving charts in a declarative manner. Notably, Counterbalance employs a novel scheme recommendation model to identify the implementation details of differential privacy approaches that successfully achieve user-defined goals such as preserving sensitive information while still effectively communicating desired insights. The tool also supports post-generation feedback on goal achievement as a way to validate goal achievement and mitigate potential misuse (such as inadvertent deductive disclosure) based on ambiguous declarations. Based on our process of designing, implementing, and empirically evaluating Counterbalance, we discuss how declarative approaches can support data owner accessibility, provide security guarantees, and promote human-in-the-loop privacy-preserving visualization.

Index Terms—Chart authoring, privacy preservation, declarative tool.

1 Introduction

Visualization represents a widely used strategy for analyzing and communicating about data, and there now exist a variety of software tools that support authoring visualizations. As an example, declarative and grammar-based tools [39] support creating visualizations in an expressive and user-friendly manner—e.g., by providing a set of input declarations that specify design or task requirements that a chart should satisfy, which are used by a rendering component to output one or more charts that meet these requirements.

Unfortunately, such tools can stand at odds with an increasingly important concern that many of today's datasets contain personal or otherwise sensitive information. In many cases, dataset owners cannot release or publish the raw data, but they still must create visualizations that illustrate the information they contain [6]. However, naively plotting data points or statistical patterns can expose the underlying records to privacy exposure [6]. Such risks become more pronounced given current regulatory trends, such as the European Union's GDPR [52] and California's CPRA [9], which emphasize that dataset owners are responsible for ensuring the anonymity of data subjects while managing and sharing information, including via visualization.

- *Xumeng Wang, Shuangcheng Jiao and Yining Wang are with DISSec, Nankai University. E-mail: wangxumeng@nankai.edu.cn, jiaoshuangcheng@mail.nankai.edu.cn, yiningwang@mail.nankai.edu.cn.*
- *Chris Bryan is with SCAI, Arizona State University. E-mail: cbryan16@asu.edu.*
- *Xumeng Wang is the corresponding author.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

A common strategy for protecting privacy (both for data release and privacy-preserving visualization) is the use of anonymization methods. In particular, differential privacy (DP) [12, 59] has become a widely used mechanism that introduces controlled randomness (i.e., noise) to datasets, and has previously been demonstrated as an effective technique for creating privacy-preserving visualizations that protect sensitive information in the underlying dataset (e.g., [63]). Unfortunately, the application of DP, particularly in scenarios where dataset owners must create privacy-preserving visualizations, is a non-trivial process for several reasons:

(i) For dataset owners who are applying DP mechanisms to create privacy-preserving visualizations for public release (who might be non-experts in the technical and mathematical foundations of privacy algorithms [6]), the "correct" application of DP can be highly non-intuitive. For example, differential privacy employs random mechanics to generate noise, and exerts a random impact on the statistical result. The causal relationships between the input settings for a DP method and its underlying effects and output results can appear unintuitive and nonlinear (e.g., see [13,45] and our pre-study in Section 3.3). Moreover, during privacy-preserving visualization, users must balance the trade-offs between the security of private information and the representation utility of insights and statistical patterns in the data [55, 63]. Making this problem even more challenging, the application of DP to a dataset is oftentimes performed prior to, and without considering, any subsequent visualization authoring steps, which makes it difficult to optimize the DP process specifically for visualization-driven goals and outcomes while still maintaining appropriate "safety".

(ii) By adding noise to the visualized data, DP increases the uncertainty in the data patterns and insights that can be conveyed. Moreover, there now exist several variations of methods for implementing differential privacy [15], and a given variant might outperform others at maintaining desired patterns or insights for the given statistical result or achieving certain privacy-preserving requirements [20]. Currently, there are no established guidelines to follow when selecting a variant for a given user goal or requirement, particularly within visualization scenarios. These intermingling challenges of uncertainty and variant and parameter selection become even more pronounced when users have multiple patterns or insights to present, which is a common use case when publishing visualizations [44].

(iii) Several tools now exist for creating differentially private visualizations (e.g., [26, 41]), but they primarily employ a hypothetical deductive approach (see Sec. 3.1). While this approach can support various *what if?* workflows, whereby users can interactively vary DP parameters and observe their effects and outcomes on a visualization, there is little support for knowing what the "correct" configuration is to achieve a desired outcome state or set of visualization goals. In contrast, declarative methods (e.g., [24]), whereby users specify the end design or state of a differentially private scenario, employ a reverse deduction approach (i.e., inferring input specifications from desired output goals to achieve the desired result). However, this latter approach has been relatively underexplored as an accessible and intuitive solution for data owners creating privacy-preserving visualizations.

Motivated by these challenges, we investigate how to accessibly and expressively integrate DP mechanisms into visualization authoring via a declarative approach. In particular, the requirement for privacy preservation imposes two critical challenges on the process of authoring visualizations via a declarative approach. First, the designer of the chart (who has access to the raw dataset) must provide details about the required levels of privacy preservation, while also needing the end result of the visualized data to appropriately display desired insights and statistical patterns [63]. Unfortunately, current declarative and domain-specific language (DSL) approaches lack formalisms to define privacy rules and implementation semantics. Second, dataset owners need to understand risks in data sharing and evaluate privacy-utility tradeoffs in created visualizations [6]. Put another way, insight is of paramount importance in information visualization. However, given that a single visualization often encapsulates multiple distinct insights [3], quantifying the utility loss under noise perturbation remains a nuanced challenge. If a tested privacy mechanism fails to meet the user's expectations, there must be mechanisms to notify users and support them in refining and re-designing the visualization.

As a part of our investigation, we develop and present Counterbalance, a declarative tool designed to facilitate the creation of visualizations while safeguarding individual privacy. To establish a lightweight yet comprehensive declarative framework for this, we conducted a two-phase pre-study. In the first phase, we systematically analyzed existing practices to identify important fields for declaring both privacy preservation goals and chart representation requirements. We then engaged potential users to gather insights on preferred implementation approaches for these fields. The developed Counterbalance tool integrates a privacy-preserving module into ECharts [34], a widely-adopted DSL. We automatically configure privacy-preserving visualizations that strictly adhere to users' declarations by integrating a built-in transform field for raw data manipulation in ECharts, which then injects DP mechanisms into the data processing pipeline. Our pipeline can align with users' declarations and automatically seek appropriate implementation details for DP, which decreases the difficulties in generating privacy-preserving charts. Furthermore, Counterbalance offers feedback on the fulfillment of user-defined goals or any necessary compromises, significantly reducing the risk of improper application of DP techniques. To validate our approach, we evaluate Counterbalance via a set of performance tests, an empirical user study to ascertain Counterbalance's strengths and weaknesses, and a pair of use cases to demonstrate further feasibility. We also discuss lessons learned and how they can apply both to the visualization and privacy communities, such as the nuance intersection of optimizing visualization design while considering privacy semantics *in situ*, and how declarative approaches have the potential to support accessibility for privacy-preserving visualization.

2 Related Work

2.1 Domain-Specific Languages for Visualization

Over the last several years, there has been a surge in the development and use of DSLs to scaffold the authoring of visualizations (see McNutt [39] for an overview), and there now exist many DSL tools and libraries, which embeds both general purpose (e.g., [47]) and for specific application and domain scenarios (e.g., [35, 38]). Importantly, the design of a DSL must balance across multiple competing factors, including expressiveness, flexibility, efficiency, and learning costs [21].

DSLs can be either formal or colloquial [39]. Formal DSLs refer to those with strictly defined syntactic structures, where all declarations must adhere to a fixed schema. For instance, Vega-Lite [47] requires users to declare a Cartesian plot by specifying a four-tuple consisting of data, transforms, mark types, and encodings. In contrast, colloquial DSLs allow multiple equivalent expressions for the same concept, which are then normalized by the system. Their colloquial nature lies in the flexibility of expression, not in the informality of syntax. While they are not bound by a rigid, formalized framework, they typically support data mapping to aspects of particular chart forms [39]. While this can allow for more customizable and diverse charts, it can also lead to user errors, such as ambiguous descriptions of charts that the DSL cannot effectively understand. One strategy for mitigating this (and also reducing the learning curve for designers) is the use of chart templates; for example, ECharts [34] provides a variety of chart templates for customization and scaffolding, whereby if a target chart type is included in the template library, users can reuse its partially implemented details.

Relevant to this paper, our pre-study investigation into the declarative preferences of potential users (Sec. 3.3) indicated a more colloquial and flexible style of expression was desired for chart authoring. However, there are no existing colloquial DSLs that support DP in the chart authoring process. This meant that a highly expressive colloquial language would need to be developed based on a systematic template collection. As a part of our work, we collect declarative templates from existing practices for assessing representative charts' effectiveness or implementing DP approaches.

Be they formal or colloquial, existing DSL-based authoring tools (e.g., [8,34,36,37,47,48]) identify diverse balance points to satisfy users with different domain and design knowledge, visualization expectations, and authoring preferences. Tools can also be built on top of existing

DSLs (e.g., by module wrapping or functionality extension); a recent example of this is Cicero [29], which reuses modules in Vega-Lite, such as the visual encoding module, and extends declarative functionality for visualization transformation. In our case, Counterbalance extends the colloquial DSL ECharts [34] with a novel privacy-preserving module to safeguard sensitive information in charts by DP.

2.2 Interactive Privacy Preservation

Privacy-preserving schemes can be formulated by specifying implementation details or describing the target effectiveness. For the first case, implementation details that need to be specified depend on the privacy-preserving techniques employed. For example, syntactic privacy approaches [18,50] are implemented based on equivalence classes, which could be interactively split from the entire dataset [55]. When applying DP, users must specify the target for noise injection, which can either be a set of aggregated query results or individual data records. Importantly, both syntactic privacy and differential privacy have a variety of methods available (e.g., k-anonymity or l-diversity for syntactic privacy and Laplace mechanism or Gaussian mechanism for differential privacy). To formulate a privacy-preserving scheme, users need to identify the specific methods and parameters for the implementation.

The target effectiveness can be described from the perspective of privacy guarantee or the perspective of utility maintenance. The first perspective is mainly described by inputting values of indicators, such as the number of indistinguishable equivalence classes for syntactic privacy [54, 55] and the privacy budget for DP [24, 41]. Compared with privacy guarantee, descriptions of utility maintenance can be more diverse because data patterns or graphical patterns in charts have a variety of types. Moreover, there are inherent tradeoffs between privacy and utility, and several prior works have explored how to balance and optimize these competing factors. For example, Dasgupta et al. [11] examined privacy-utility tradeoffs across several metrics designed to help characterize the overall informativeness in privacy-preserving visualizations (specifically for parallel coordinates and scatterplots), and Panavas et al. [42] explored how various parameters (privacy levels, binning amounts, etc.) impacted the visual utility of DP scatterplots. Likewise, tools such as ViP [41] allow users to measure utility preservation by directly examining the probability distribution of the random noise magnitude. DPKnob [24] goes a step further by allowing users to derive the privacy budget by specifying the desired confidence level or interval width for the size of random noise. DPVisCreator [63] relies on the marginal distribution of specified record groups to steer the data reconstruction process for preserving sensitive information in charts. Unfortunately, these works tend to characterize utility maintenance solely in terms of individual features or metrics such as distribution similarity, which overlooks distribution-irrelevant patterns, such as rank, extrema, and variance. They also tend to employ a hypothetical deductive approach, whereby users specify the anonymization parameters and observe the result. In contrast, Counterbalance supports a wide range of patterns for maintaining utility, and we explore the feasibility of a reverse deductive approach in achieving privacy-utility tradeoffs, whereby users can describe the end state or goal of privacy via a declarative grammar, and Counterbalance identifies the best set of input specifications to achieve the desired result.

3 Requirements Analysis

In this section, we first provide a formal definition of DP, and then describe our requirements analysis activities, where we first identified several candidate fields for invoking DP within a declarative workflow, and then empirically distilled preferred practices about how to declare and utilize these fields to create DP-based visualizations.

3.1 Differential Privacy Definition

DP [12] employs a random mechanism K to prevent privacy leakage from database D by making the information derived from D indistinguishable from any database D' that differs from D by a single record. An indistinguishable case can be described by the following formula:

$$Pr[K(D) \in S] \leq \exp(\varepsilon) \times Pr[K(D') \in S],$$

where S denotes an arbitrary subset of the range of K and ε is a parameter used for regulating the indistinguishability degree, that is, the level of privacy preservation. In practice, using a smaller value of ε can activate a more strict privacy-preserving standard. A privacy-preserving standard can be achieved by multiple mechanisms, which lead to different utility losses. A group of settings consisting of the value of privacy budget ε and the mechanism K can specify an implementation of DP. However, specifying the exact and appropriate implementation details in this way (this is the hypothetical deductive approach discussed in Sec. 1) can require a high level of expertise and/or trial and error.

3.2 Identifying Potential Declarative Fields for DP

While DP is sometimes used in "query" scenarios where data customers cannot access the raw data and are returned a DP-protected version of it (e.g., [56]), we are interested in "public release" scenarios [64] where dataset owners and custodians who have access to the raw data create and publish DP-protected visualizations of datasets (e.g., [4, 63]). This type of scenario is becoming increasingly common across a variety of application domains, including healthcare [49], social sciences [23], and business [28]. When creating DP visualizations for public release, dataset owners must balance tradeoffs like privacy preservation, dataset utility, and the visual insights or information they want to communicate in the created charts. As mentioned in Sec. 1, this process can be challenging for several reasons, including that the relationships between DP parameters and output visualizations might be non-intuitive (and nonlinear), and that DP mechanisms are often applied *prior* to visualization authoring, which makes it difficult to consider both visualization authoring and safety in parallel.

In such scenarios, the effectiveness of a privacy scheme can be considered from two high-level perspectives: *privacy* and *utility*. To address these perspectives from a declarative approach (i.e., via a visualization grammar), we first conducted a requirements analysis to identify a set of potential "required fields" that would be useful to declare within a grammar for authoring DP visualizations. For each field, we identified potential "feasible practices," which represent viable ways to express the field. To identify potential required fields and feasible practices, we conducted a meta-analysis of existing studies (e.g., [31, 60, 62, 63]) and privacy-preserving visual analytics tools and techniques (e.g., those discussed in Sec. 2.2).

Below, we briefly describe the set of identified required fields and feasible practices for declaring those fields (these are also shown in Fig. 2). These items were subsequently used in a pre-study (Sec. 3.3) to empirically validate a requirements set for specifying DP visualizations in a declarative grammar-based manner.

3.2.1 Declarations for Privacy Preservation

As mentioned in Sec. 3.1, a user could specify their privacy-preserving needs by directly assigning a privacy budget value for ε. Alternatively, privacy requirements can be considered inversely (via the "reverse deduction approach" discussed in Sec. 1). For example, although DP mechanisms have aspects of randomness when introducing noise, the causality between the value of the privacy budget and the privacy-preserving effectiveness can be inferred based on its indistinguishable effects [32]. As such, users could declaratively specify a desired end state, and use that information to "work backwards" and solve for a suitable value for ε. There are many ways to define various aspects of such an end state, such as setting a maximum success rate for a given type of privacy attack [32] (e.g., this could be accomplished by declaring two fields: attack type and the success rate). On the other hand, datasets with different levels of sensitivity could accommodate customized applications of sanitization (e.g., DPKnob [24] achieves this by letting users describe the target information to be preserved, such as a subject group and corresponding attributes of interest). Ultimately, we identified four high-level candidate fields for declaring privacy preservation (Fig. 2(a)), each with one or more feasible practices for declaratively specifying the field. As an example, the first potential required field deals with *target information* (i.e., sensitive information held by potential targets), with two common feasible practices for

specifying: (i) the user specifies sensitive information by choosing one or more attributes, or (ii) all attributes are assumed sensitive by default.

3.2.2 Declarations for Utility Maintenance

The application of DP will distort (i.e., based on noise) a chart's rendered marks and channels, with the byproduct of altering data patterns and insights that are visible. When considering utility-related goals for privacy-preserving visualizations (the *utility maintenance*, as discussed in Sec. 2.2), prior work (e.g., [55, 63]) has stressed two aspects: (i) despite distortions, it is critical that the visualization maintain desired insights, and (ii) it is important to properly balance (or, at minimum, contextualize) the application of noise based on its "weakening effects" that might inhibit these insights.

Visualization insights. We consider three aspects to visualization insights [5]: type, attribute, and range. For insight *type*, we reviewed and categorized what types of insights users in previous works learned from privacy-preserving charts, based on their data patterns and graphical patterns, as shown in Table 1. *Data patterns* refer to insight types that can be numerically quantified, such as a maximum value. *Graphical patterns* are insight types involving entangled details that cannot be covered by a single value. For the insight *attribute*, we reviewed what types of data attributes are used to describe the insight; likewise, the insight *range* pertains to the subset or subspace of the dataset (or data points) containing the insight.

Table 1: Common insights that represent utility-related goals for privacy-preserving charts. We remark how each type of insight could be described from the perspectives of consistency judgment (C: C1 - whether the described object(s) would be changed and C2 - whether the value change would exceed a threshold) and similarity quantification (S: S1 - the significance difference of corresponding patterns and S2 - the value difference of a metric).

Insight Types		Descriptions	Remarks
Data patterns	Identity	Whether a data record has an identity, e.g., maximum, outlier.	C1, S1
	Dispersion	Measures of differences, e.g., range, variance.	C2, S2
	Shape	Measures of distribution, e.g., skewness and long-tailedness.	C2, S2
	Metric values	E.g., count, maximum, minimum, mean.	C2, S2
Graphical patterns	Clusters	Division of clusters.	C1, S1
	Structure	Distribution outline.	C2, S2

Weakening effects. We also identified two common methods to describe the distortion effects caused by DP schemes: *consistency level* and *similarity* (see Fig. 2), both of which can be consolidated into potential declarations that describe the weakening effects of DP on a dataset's visual utility.

Consistency level evaluates distortion through a binary determination of whether the original pattern is preserved. For example, DPKnob [24] transforms the accuracy measurement for values disturbed by DP into a binary question by setting a threshold for tolerable error. If the perturbation exceeds the threshold, the disturbed value is considered inaccurate, which negates the consistency judgment. Given the inherent randomness of DP mechanisms, multiple tests can be conducted to derive a consistency probability, providing a statistical measure of overall performance.

Similarity focuses on quantifying differences between original and perturbed patterns. As the randomness mechanism of DP schemes perturbs the patterns visualized in charts, comparing the patterns between the original chart and the perturbed chart can be used to assess utility maintenance based on the privacy preservation [54, 55]. Given the stochastic nature of DP, the similarity should be measured by averaging multiple test results using either absolute or relative values.

3.3 Distilling Preferred Practices for Potential Users

To validate that Sec. 3.2's identification of potential fields and practices was ecologically sound, we next conducted an evaluation with five experts at the intersection of the visualization and DP communities. The five participants (two focused on visualization, and three on DP), had a robust understanding of theory and applications in their specific field through participation in the research and development of projects of high-tech companies for at least a year. Each was also educated at or beyond the master's level, and had at minimum a moderate level of expertise and knowledge about the other field (visualization or DP).

Each participant completed a questionnaire based on items from the analysis in Sec. 3.2, supplemented by 15-minute semi-structured follow-up interviews. Importantly, while we do not consider this pre-study as a full sampling of the community, we highlight that it reflects the diverse expectations and expertise levels of potential users.

Questionnaire design. Our questionnaire began with an introduction to declarative objects and their potential application to creating charts that integrate DP mechanisms. Subsequent questions probed (i) which declarative fields were considered required for declaratively creating DP visualizations, and (ii) what was the most feasible preferred practice (or strategy) to provide specifications for these fields. As shown in Fig. 2, for each question we adopted a progressive questioning strategy: only the respondents who felt a field was required (left column) were asked follow-up questions about the field's feasible practices (right column) to identify their preferred strategies about how to make this declarative activity feasible. To gather broad feedback on the preferred and feasible practice questions, experts were asked to check all applicable options. To minimize ambiguity, survey respondents were also provided examples for each description of the required field and feasible practice.

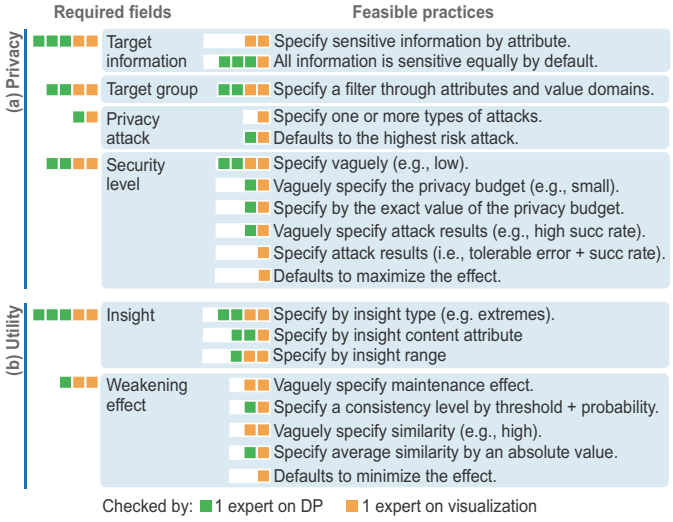

Figure 2: Summarized questionnaire responses from our pre-study with visualization and DP experts.

Results. For fields relating to privacy (Fig. 2(a)), all five experts (5/5) felt that *target information* needed to be a required field. For its corresponding feasible practices, both visualization experts felt that specifying sensitive information by attributes should be enforceable, while the DP experts also highlighted that treating all information as equally sensitive is a common practice in their field.

Similarly, 4/5 experts felt that *target group* and *security level* were required fields. Specifically for *security level*, the most popular preferred practice was to vaguely specify the security level; other options received mixed preferences. For example, one DP expert opposed specifying an exact value for the privacy budget, citing the risk of misuse by users who lack expertise on DP. One visualization expert told us that they would have a more concrete idea of privacy-preserving goals when declaring privacy-preserving goals based on attack results.

For the *privacy attacks* field, only 2/5 experts marked this as necessary; feedback from the follow-up interviews suggested this type of declaration could default to the highest risk attack, with a justification that privacy threats and attack vectors are unpredictable.

For responses about utility maintenance (Fig. 2(b)), all five experts (5/5) felt *insight* was a necessary field, though there was variability in how they preferred to declare it. *Weakening effects* was considered necessary by 3/5 of the experts. During the interviews, two DP experts raised concerns that the process of determining such declarations (both for *insights* and *weakening effects*) through data previewing could lead to privacy leakage [45], because the privacy guarantee provided by randomness could become unreliable if the randomly generated results can influence user judgment and trigger re-generation. However, visualization experts argued that it is essential to preview and identify valuable insights prior to finalizing a visualization for public release [53].

Balancing this trade-off led, in part, to introducing DP noise in Counterbalance's preview stage (see Sec. 4.2.3) to ensure a strict privacy guarantee. More broadly, the feedback from this pre-study was used to help design, define, and implement a novel grammar for defining privacy-preserving and utility-maintaining requirements (see Sec. 4.2).

4 COUNTERBALANCE

In this section, we first discuss the design considerations of Counterbalance, and then describe the details of the grammar and tool.

4.1 Design Considerations

Based on Sec. 3's requirements analysis, we distilled a set of five design considerations (**DC**s) for declaratively authoring privacy-preserving visualizations using DP.

DC1: Declaring goals instead of specifying implementation details. Declaring a privacy budget and DP mechanism directly (the hypothetical deductive approach) can be challenging given the potentially unintuitive, nonlinear, and varied nature of DP mechanisms [12], particularly for users without a technical/mathematical background in DP [6]. To support a more user-friendly, reverse deductive approach, users should be able to declare their goals, with the solutions generating implementation details that automatically achieve these goals.

DC2: Multiple goals may exist. At a high level, DP visualization requires balancing two distinct goals: privacy preservation and utility maintenance. Within either of them, there may exist multiple subgoals. For example, privacy goals can be customized for distinct target groups [54], while visualizations often present multiple insights [24]. Thus, the expected solutions need to simultaneously seek multiple (potentially competing) goals of varying priority.

DC3: Using a colloquial DSL. As discussed in our pre-study results in Sec. 3.3, potential users of our tool preferred to declare their goals and practices in a variety of ways (including due to individual differences in expertise, declarative habits, goals, etc.). To accommodate diverse preferences, solutions should enable flexible expressions for goal declaration. This promotes a colloquial DSL design (as opposed to a formal DSL, see Sec. 2.1) to better align formal syntax with naturalistic and flexible human expression.

DC4: Feedback on goal achievement is necessary. It is well known that the application of privacy-preserving approaches may not always achieve an equilibrium between safeguarding privacy and maintaining full utility [32,55]. As such, it is critical that users can comprehend any failure(s) or exposed risks, and are able to revise their goals accordingly. A versatile solution should not only respond to multiple declarations but also provide insightful feedback on the fulfillment status of each goal. Given that creating privacy-preserving charts may involve multiple goals (as in **DC2** above) and diverse declarative methods, this process may also lead to errors [39]. Therefore, users must also be able to verify that their declarations are correctly interpreted by the solution.

DC5: Assistance for goal declaration. Taking declarations for *security level* as an example (Fig. 2), a user could exactly declare a privacy budget; alternatively, security might be inferred contextually by specifying potential attack results. While declarative grammars can provide a precise format to define these specifications, manual composition (e.g., via a command line or JSON file) would likely impose a steep learning curve, as textual programming forces users to mentally map their high-level intent to rigid syntactic structures, relying on "recall" rather than "recognition" [46]. This cognitive gap is particularly challenging in the context of DP, where translating abstract goals into exact grammar parameters remains unintuitive. This cognitive gap is particularly challenging in the context of DP, where translating abstract goals into exact grammar parameters remains unintuitive. By employing a form-based interaction panel, novice users can intuitively align the expression methods they prefer with the rigorous structure of declarative grammars. A parallel view that toggles between a syntax editor and form-based controls allows novices to first learn grammar through forms and then graduate to direct editing, while still accommodating experts who prefer syntax from the start.

4.2 Design of Counterbalance

We operationalize **DC1**–**DC5** by developing Counterbalance, whose interface is shown in Fig. 1. Counterbalance generates privacy-preserving charts via three novel contributions: (Sec. 4.2.1) a declarative grammar for DP schemes, (Sec. 4.2.2) a DP scheme formulation model, and (Sec. 4.2.3) interfaces and user experiences for interactively authoring and reviewing privacy-preserving visualizations. Counterbalance's grammar works by declaring the goals of privacy preservation and utility maintenance. According to the extracted goals, Counterbalance formulates an appropriate DP scheme through a decision-making model, which is used to generate a privacy-preserving chart. Counterbalance inspects how each user-declared goal is satisfied and generates a completion report for the employed DP scheme, allowing human-in-the-loop review, validation, and export for publishing.

4.2.1 A Declarative Grammar of Differential Privacy

Building upon the chart generation capabilities of ECharts [34] (one of the most popular and established colloquial DSLs), Counterbalance integrates a goal-oriented grammar for DP scheme customization (**DC1**) to endow charts with privacy protection. Users can articulate multiple goals (**DC2**) for both utility and privacy, each described by a tuple. Leveraging the findings from our pre-study questionnaire, we define a 2-tuple framework for privacy and a 2-tuple framework for utility.

$$\begin{aligned} \text{Privacy:} &\quad < target(attr*, group), security\ level > \\ \text{Utility:} &\quad < insight(type*, attr*, range), weakening\ effect > \end{aligned}$$

As shown in Fig. 1(a3), Counterbalance supports colloquial-like descriptions (**DC3**) of each component in tuples through key-value pairs in a JSON format. Components with multiple options for declarative methods (e.g., *security level* and *weakening effect*) can use different keys to specify declarative methods for component description, as shown in Fig. 3. For instance, the key for *weakening effect* could be either *consistency* or *similarity* (see Table 1). The key for *security level* could be *privacyBudget*, *attackSuccRate* [24], or *level*. The key of *target* is defined by a target attribute (required) and a target group (which defaults to all). The key of *insight* comprises an insight type (required), insight attribute (required), and insight range (defaults to all). Both the target group and insight range consist of attribute-value pairs that serve to constrain the scope of the *target* and *insight*, respectively. If multiple declarative methods are declared simultaneously in the JSON, Counterbalance will identify multiple goals by decomposing the JSON into a series of tuples.

As discussed in Sec. 2.1, one risk of colloquial DSLs is that their declarations can be prone to user errors such as missing components or invalid declarations. To mitigate this, Counterbalance employs default values to complete the declaration and provides feedback on how declared goals are interpreted to expose underlying errors (see Sec. 4.2.3 below). Default values are set and applied by prioritizing the security of privacy and encompassing all records (pertaining to both target groups and target information) as protected entities to ensure comprehensive safeguarding (see Fig. 3).

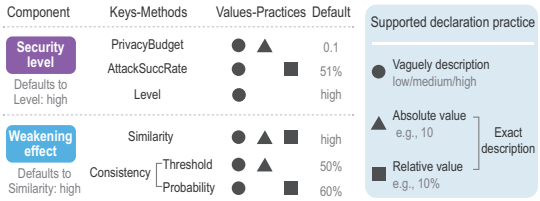

Figure 3: Grammar details of the components with multiple declarative methods. Note that the key of "Threshold" is not required when declaring utility-maintaining requirements with the method of C1 (see Tab. 1).

4.2.2 Differential Privacy Scheme Formulation

To formulate an appropriate DP scheme, we propose a novel four-part model that (a) extracts user-specified goals and determines the privacy budget, (b) generates and (c) evaluates potential DP schemes that satisfy the scenario, and (d) recommends the DP scheme that's predicted to perform best as the appropriate mechanism to use for the current user-declared privacy and utility-maintaining goals. The process is shown in Fig. 4 and described below.

(a) Privacy goals extraction. We begin by determining the value of the privacy budget according to any user-specified privacy-preserving goals, which narrows down the space of feasible DP schemes. We integrate goals which are associated with the same target information by solving skyline problems [7] as illustrated in Fig. 4(a). As mentioned in Sec. 4.2.1, related goals can be declared for a certain target. We implement the goal integration process across targets of the same attribute because, when drawing charts, the statistical results of each target information take all individuals as statistical objects. For each target information, we identify the dominant goal corresponding to the highest security level (or say, the lowest privacy budget) from all corresponding goals declared by users. According to the dominant goal, we calculate the upper threshold for the privacy budget ε and apply it in the DP scheme for the target information.

(b) DP schemes examination. DP schemes can achieve varying levels of utility when applied to data with different distribution features. For instance, EFPA [2] has been shown to result in significantly higher utility for sparse data compared to other DP mechanisms given the same privacy budget. To evaluate potential DP schemes, we generate simulated statistical results focusing on distribution features that either relate to specific insights (e.g., max values) or that influence DP utility. The generation process continues until the dataset spans the entire value range for each feature dimension, ensuring sufficient variability. Specifically, we monitor the distribution of generated samples to verify that the combination of features achieves a stratified coverage across the target space. We conceptualize distribution features as dimensions to form a data space, from which we draw samples to produce data for testing purposes, as shown in Fig. 4(b).

Next, we generate an ensemble of DP schemes by matching the privacy budget from 0.1 to 1 with candidate DP mechanisms. Specifically, we consider nine candidate DP mechanisms (Laplace [14], HB [43], PRIVELET$^+$ [57], DAWA [33], PHP [2], DPCube [58], AHP [61], MWEM [19], and EFPA [2]) which represent popular and state-of-the-art techniques from each of the categories summarized by Hay et al. [20]. We implement between 40–200 tests for each scheme on each test data and summarize the examination result as a probability distribution (as the effectiveness of DP schemes is affected by the randomness of noise). The reason for the variance in the number of tests for each scheme is based on the privacy budget. Specifically, the privacy budget varies because when generating a DP scheme, we match the DP mechanism with different values of privacy budgets. As a lower budget introduces greater uncertainty, more tests are required to obtain stable and reliable results. We quantify the weakening effects as the examination result for each DP scheme in achieving a specific utility-maintaining goal. To further consider the intents of users, we apply different quantification approaches for goals declared by the two declarative methods for weakening effects (see Table 1). For goals based on consistency judgments, the weakening effects are quantified as the probability of failing to maintain the specified insight. For goals declared based on similarity, the weakening effects are measured as the average dissimilarity relative to the true value.

(c) Utility-aware induction. In the next module, we summarize how each DP mechanism can satisfy utility requirements given different privacy budgets and statistical results with different distribution features. Kotsogiannis et al. [30] demonstrated how decision tree models can be used to recommend DP mechanisms for query tasks (e.g., range queries) according to data features. Inspired by this, we adapt decision tree models to satisfy diverse requirements declared by users, as shown in Fig. 4(c). To ensure the accuracy of recommendations, we first divide underlying requirements by grouping utility-maintaining goals according to the type of insights and the declarative method. Next, we conquer each group of goals by training a decision tree based on examination results. During the training process, we identify appropriate partition nodes based on a distribution-related indicator, namely kurtosis [25] (see Sec. 6) or the value of the privacy budget. To improve training efficiency, we filter distribution features according to the examination results before training. To support scheme formulation based on multiple utility-related considerations, we keep the overall performance of all DP schemes as the output of decision trees. We calculate overall performance based on the probability that a DP mechanism can achieve a near-optimal performance due to the stochastic nature of DP mechanisms. Following Hsu et al. [22], we define near-optimality as yielding a weakening effect in performance that is less than 10% above the optimal mechanism. When training the decision tree, we judge whether two branches can be merged based on the similarity between the vectors encoding the near-optimal probability of all DP mechanisms.

(d) DP scheme selection. Finally, we determine the DP schemes by integrating the output of the decision trees (Fig. 4(d)). For each utility-maintaining goal specified by users, we leverage the corresponding decision tree to predict the performance of all mechanism candidates. Each prediction is achieved by analyzing the privacy budget allocated by the privacy goals extraction module and the kurtosis of the input statistical result, which our empirical results demonstrate to be the most effective metric for distinguishing between differential privacy mechanisms. To comply with DP constraints, an additional 10% privacy budget is allocated for the data query to the value of kurtosis based on the classic Laplace mechanism. We compute an overall near-optimal probability weighted by declared weakening effects for each mechanism to assess how each mechanism can satisfy all user-declared utility requirements. Ultimately, the DP scheme with the highest probability of being overall near-optimal is selected as the recommended DP mechanism. Given the value of the privacy budget determined through privacy goals extraction (Fig. 4(a)), we formulate the DP scheme for user-declared requirements.

4.2.3 Interface and Workflow

In the Counterbalance interface (shown in Fig. 1), users create privacy-preserving charts through two phases. (For a step-by-step demonstration of these phases in action, please also see the supplementary video.) In the Chart Design Declaration phase, users specify the ECharts configuration options to initiate a chart. Because Counterbalance is built atop the ECharts DSL, it supports flexible and accessible visualization authoring including what types of charts are used (line charts, bar charts, pie charts, etc.) and how the charts are styled and themed. Seeking a strict DP guarantee, the preview chart is generated using a DP scheme that consumes a user-specified privacy budget (defaulting to $\varepsilon = 0.2$, EFPA mechanism). Next, in the DP Declaration phase, users determine DP goals for the initial visualization. Below, we detail these phases.

The Differential Privacy Declaration interface comprises two linked views: one for specifying DP declarations (Fig. 1(a)) and another for the resulting privacy-preserving visualization (Fig. 1(b)). The declaration view includes an ECharts-based editor (Fig. 1(a2)) displaying the base chart configuration defined in the first phase. Users specify goals (**DC5**) via an interactive panel (Fig. 1(a1)) or directly in a grammar-based declaration editor (Fig. 1(a3)). Mandatory grammar inputs (Sec. 4.2.1) are marked with an asterisk (*). Alternative practices or formats are

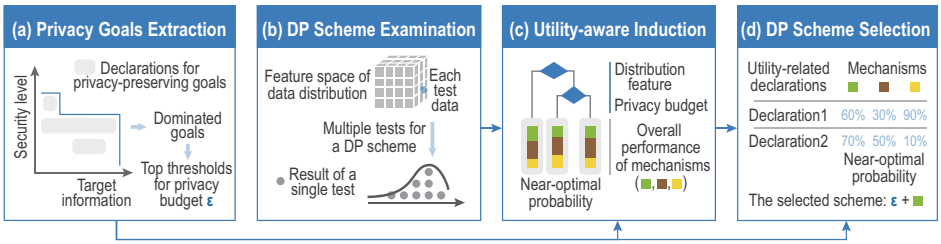

Figure 4: The pipeline of the DP scheme formulation model employed by Counterbalance, which consists of (a) privacy goals extraction for privacy budget calculation, (b) DP scheme examination for effectiveness assessment, (c) utility-aware induction for DP mechanism, and (d) DP scheme selection considering all utility-maintaining goals).

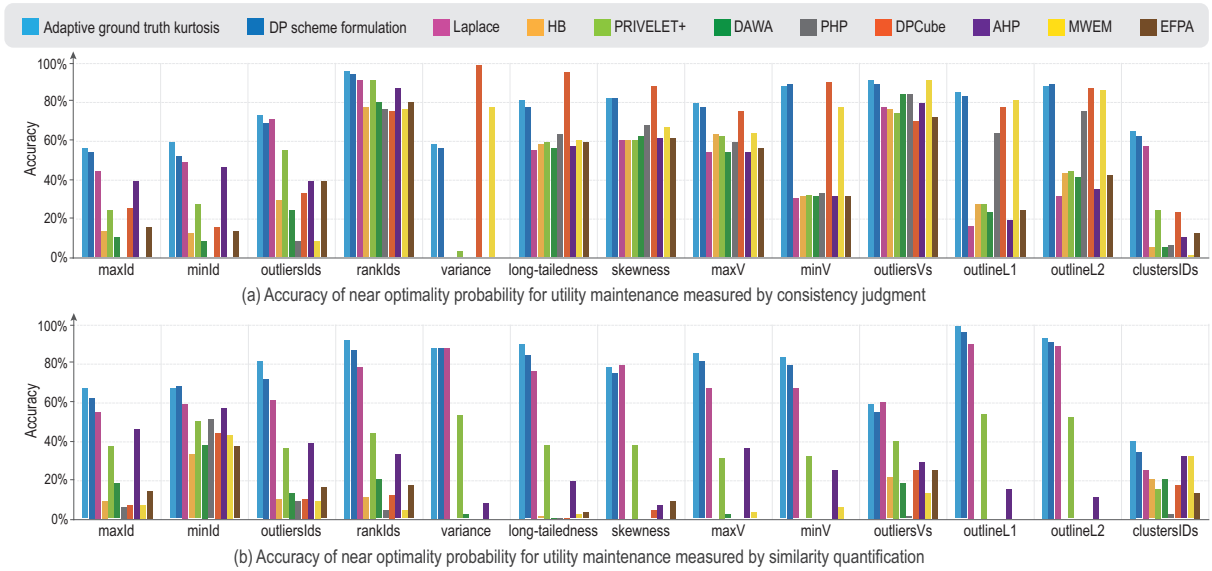

Figure 5: Accuracy comparison of near optimality probability for utility maintenance among DP mechanisms.

linked by a chain icon and an "or" indicator, auto-synchronizing once one is filled. A "Convert" button translates the panel's configuration into the editor's syntax.

According to the formulated DP scheme, Counterbalance provides feedback as a completion report (Fig. 1(a4)), where red and green icons highlight whether the preset goals are met. Feedback includes the implementation details of the formulated DP scheme and the assessment result of goal achievements from the perspective of DP schemes (**DC4**). Additionally, it evaluates other potential insights against a default high threshold, reporting only the unmet ones to highlight unintended utility loss. Counterbalance provides no assessment on the output chart processed by the DP scheme because ensuring randomness is a prerequisite for DP to safeguard privacy. Also, the output is a random instance decided by the employed DP scheme. Once a DP scheme is approved, users can employ the output from the scheme without alteration. If satisfied with the assessment result of the DP scheme, they can export the privacy-preserving chart (Fig. 1(b)). Otherwise, they can return and modify the declaration until the assessment result is satisfactory.

5 Evaluation

We validate the effectiveness of Counterbalance in several ways. First, we report a performance test for the DP scheme formulation model, and next conduct a user study[1] to evaluate the tool's interface and user experience. For additional demonstration of this tool, please also see the Appendix where we report two real-world exploration processes from the user study as use cases, as a way to highlight representative examples of how to use Counterbalance.

[1]The Institutional Review Board determined that this study was exempt from full review.

5.1 Performance Test

The DP scheme formulation model (described in Sec. 4.2.2) supports the declaration of multiple utility-maintaining goals. To evaluate its response capabilities, we tested the quality of the DP schemes formulated by our model in two cases: (i) a single requirement is declared, and (ii) multiple requirements are declared. In each test, we evaluate the model's performance by comparing the effectiveness of utility maintenance by using fixed DP mechanisms with the mechanisms dynamically formulated by our model. To ensure a robust evaluation, we compare our proposed approach, an adaptive mechanism formulated based on the queried value of kurtosis, against ten baseline mechanisms. These baselines include static applications of the nine predefined DP candidate mechanisms listed in Sec. 4.2.2, as well as an idealized adaptive mechanism that relies on the ground-truth kurtosis. To systematically evaluate model robustness, we tested various types of statistical insights across a 5×5 parameter grid. This grid consisted of 25 distinct combinations, crossing five kurtosis levels ($\{-2, -1, 0, 1, 2\}$) with five privacy budgets ($\{0.1, 0.3, 0.5, 0.7, 0.9\}$). Each combination underwent 200 independent trials to ensure statistical reliability.

Testing for single requirements. We simulated the underlying declaration for a single utility-maintaining requirement and tested the maintenance ability of each mechanism. Each declaration was formed by combining various insights with one of two declaration methods (consistency judgment and similarity quantification). We tested 13 types of insights for each declaration method. These are listed in Fig. 5, where each declaration method is a row and each insight type, such as maxId and minId (see the Supplemental Materials for full definitions), is groupings in the grouped bar chart. We measured the maintenance ability by accuracy and the probability that the mechanism is near-optimal (i.e., less than 10% above the optimal case). As shown in

Fig. 5, the baseline DP mechanisms exhibit significant performance variations when meeting different utility-maintaining requirements. In contrast, the DP mechanisms formulated by our model (labeled DP Scheme Formulation in the figure) generally achieve both high and stable performance. However, mechanisms selected based on query results generally underperform those selected based on ground truths because the queried result could be uncertain (please refer to Sec. 6).

Testing for multiple requirements. We next assessed the quality of DP schemes for multiple requirements through overall weakening effects. Weakening effects for consistency judgment are calculated by the probability of being perturbed, while those for similarity quantification are calculated by relative differences. To simulate underlying declarations, we randomly generated 1–3 utility-maintaining requirements based on random combinations of insight, declaration methods (i.e., consistency judgment and similarity quantification), and weakening effects. The results are shown in Fig. 6, where the DP mechanisms formulated by our model achieved lower weakening effects than any fixed DP mechanism. Specifically, compared with the optimal fixed mechanism (e.g., AHP with an average weakening effect of 27%), the DP mechanism formulated by our model reduces this effect to 22% (based on ground truth) and 23% (based on queried results), representing 18% and 16% improvements, respectively.

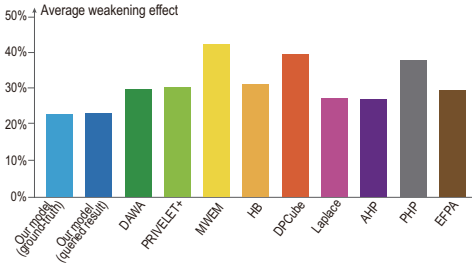

Figure 6: Comparison of weakening effects for an indefinite number of utility-maintaining requirements among DP mechanisms.

5.2 User Study

Next, to empirically evaluate Counterbalance, we conducted a mixed-methods user study that observed how users articulate privacy protection requirements based on data contexts. The goal of this study was to assess *if* and *how* Counterbalance could support the accessible creation and understanding of privacy-preserving charts. Given that there are no similar baseline authoring tools to compare against (e.g., tools like Tableau, PowerBI, and "vanilla" ECharts do not support invoking privacy rules), our design was a *single approach evaluation study* [51].

Study Design. To conduct the study, we first prepared a set of three case scenarios. Each contains a set of ready-to-use ECharts declarative configuration options and underlying requirements of privacy preservation corresponding to the used data which can be imported into Counterbalance. To properly simulate real-world chart sanitization and publication workflows, we designed each case based on an adult income dataset [40] with several individual descriptive dimensions such as income, age, and education level. Case details can be found in the Supplemental Materials. We recruited 15 participants aged 20-24 (M= 23.0, SD= 1.25) from Nankai University. Our recruitment intentionally mirrored the user demographics from Sec. 3.3's pre-study: seven participants had previously taken university-level visualization courses and had moderate expertise in DP; the other eight participants had expert knowledge in DP, and had moderate expertise in visualization. All participants also had prior experience in using declarative tools to author visualizations. Each participant received compensation ($7 USD) for taking part in the following three stages, which took approximately 40 minutes in total per participant.

(i) *Introduction and training*: Participants were briefed via a demo on Counterbalance's functionalities and how to create visualizations using the grammar. This introduction took 10 minutes on average. Next, a hands-on warm-up task was completed using a sample case to familiarize users with Counterbalance and ensure they encountered no initial usability barriers. In this process, they were also allowed to refer to an API document and receive help from a study proctor (this proctor was also available in the subsequent session). When participants felt ready, they proceeded to the next stage.

(ii) *Tool exploration*: Participants were tasked to explore Counterbalance by declaring the requirements of DP schemes based on new charts and corresponding scenarios. To capture authentic requirements, we encouraged users to work with their own familiar datasets or chart types. Alternatively, they could select cases from our pre-defined collection and datasets. The exploration process took about 20 minutes on average, during which participants also employed think-aloud protocol to verbalize their thought processes. We recorded the declarations, interaction behaviors, and verbalizations of each participant.

(iii) *Post-study reflection*: Participants completed a questionnaire consisting of ten questions adapted from the system usability scale (SUS) (see Fig. 7) to assess the usability of Counterbalance. Considering that the declarative grammar is the core conceptual contribution, our questions focus primarily on the perceived usability of the grammar within the context of using the interactive tool. The questionnaire was paired with a set of open-ended prompts to solicit additional qualitative insights and feedback.

Figure 7: The summarized results of our user study questionnaire.

Results. While the exploration phase did not require creating a specific number of charts or covering a certain set of insights (as we wanted to assess usability and creativity in an "open" scenario), we note that participants on average created 2 charts covering 2 insights each. The most commonly used utility-maintaining requirements were based on three types of data insights: the data's maximum (the maxID value in Fig. 5), and using the L1 and L2 norms to measure distribution outlines (outlineL1 and outlineL2 in Fig. 5).

When creating visualizations, the majority of participants reported that Counterbalance facilitated the effortless application of differential privacy across these diverse visualization types. For example, when using our pre-defined case templates, which described a set of financial data, participants generally created three types of visualizations: line charts (such as depicting income distribution by age), stacked bar charts (such as showing income distribution by gender and family structure), and pie charts. We also observed that feedback from Counterbalance, such as additional insights detected, could remind participants of insight types that they had overlooked, providing an opportunity to complement utility requirements.

In terms of the post-study questionnaire, Counterbalance received consistently excellent usability ratings, as shown in Fig. 7. In particular, participants rated that the grammar-based approach was easy to learn and use (**Q4**, **Q5**), and provided flexibility and coverage (**Q1**, **Q8**).

Qualitative feedback provided further context and confirmation about this. For example, multiple participants specifically discussed how Counterbalance could be highly valuable for users who need to create privacy-preserving visualizations but lack a deep understanding of DP, or need to apply it in a more nuanced manner.

Interestingly, one participant discussed how Counterbalance supports the reverse deduction approach (discussed in Sec. 1) as opposed to a hypothetical deductive approach (as a way to iteratively validate hypotheses against empirical data) whereby users could directly declare a DP mechanism (such as setting the privacy budget value ε).

This participant also noted that, even if the direct declaration of a DP mechanism was not used, it could still be included for visibility and explainability. We plan to support such functionality in the future.

Regarding user interface and experience aspects, questionnaire responses indicated Counterbalance was overall intuitive and easy to use (**Q2**, **Q9**), though some qualitative feedback provided suggestions for further improving the experience. For example, some participants noted that learning insight parameter options required time for new users, and suggested that more explicit helper mechanisms (such as on-demand tooltips, or auto-complete suggestions) could better guide users toward data-specific insights. For example, one participant highlighted that when a goal is not met and the "missed" indicator appears, the completion report could provide specific modification suggestions on how to fix the issue. To further illustrate how Counterbalance was used in this study, please see the Appendix, where we report two use cases of participants from the study using the tool to solve a privacy scenario.

6 Discussion

Declarative grammars and tools can enable users to determine implementation details based on their specific needs [39], and Counterbalance represents (to our knowledge) a first-of-its-kind approach for authoring DP visualizations in a declarative style. Our process in designing, implementing, and evaluating Counterbalance indicates it effectively reduces technical barriers to invoking advanced DP mechanisms, and that it supports performance, flexibility, expressivity, and nuanced control over what types of insights can be preserved in secure visualizations intended for public release. Below, we discuss several insights and lessons learned based on the process of designing, implementing, and evaluating Counterbalance.

Design implementation and limitations. Counterbalance successfully supports the design considerations outlined in Sec. 4.1. We achieved DC1 by integrating a goal-oriented grammar that abstracts complex DP parameters into customizable schemes. This underlying structure (see Sec. 4.2.1) is pivotal for DC2, as it allows users to articulate and balance multiple, often competing, goals regarding utility and privacy—a capability validated through our user study and use cases. To lower the barrier for entry (DC3), the system supports colloquial-like descriptions via JSON key-value pairs, enabling users to define tuple components intuitively without deep algorithmic expertise. Transparency is ensured by providing detailed feedback on the formulated DP schemes and explicitly assessing goal achievement (DC4), helping users understand the consequences of their privacy settings. Finally, the user interface directly supports these tasks by exposing the declarative grammar through interactive panels (DC5), which our evaluation participants found instrumental in bridging the gap between high-level privacy intent and low-level technical specification. While these features cover all DCs, two aspects remain less supported. First, when a declared scheme fails, Counterbalance currently only reports the failure and requires manual adjustment, lacking an automated fallback. Second, the system does not evaluate the final output chart or visually convey data perturbation. Therefore, future work should explore intelligent assistance for failure handling (e.g., suggesting alternative schemes or relaxing constraints) as well as integrate uncertainty visualization [27] to indicate the reliability of released statistics.

Balancing flexibility, power, and accessibility. It is worth clarifying that Counterbalance primarily improves the accessibility of invoking DP mechanisms, rather than accelerating decision-making about privacy-utility trade-offs. To achieve this accessibility, Counterbalance employs a configuration panel that allows interactive grammar specification without syntax errors, and a continuous feedback loop that helps users intuitively grasp semantic connections between parameters. However, risks remain for novice users who might inadvertently specify a non-sensitive attribute that is highly correlated with a sensitive one, causing unnoticed privacy leakage. The idea of protecting against such risks can be explored in future work, such as integrating functionality to support against risks not explicitly defined by the user. These functions could be invoked as a part of the "more explicit helper mechanisms." For example, prior work has explored proactively assessing the risks and exposures for created charts [10] even if not originally declared by the user, or simulating future adversarial attacks [53]. Similar functionality could be integrated into Counterbalance in the future.

Counterbalance also enables a variety of insights to be declared as the requirements of utility maintenance (e.g., see Fig. 2). However, when users lack clarity in defining potential requirements—for instance, if they are uncertain about what requirements they should specify for a given scenario—the built-in functionalities may remain underutilized. In our user study, some participants explored only a small subset of available insights because they did not fully understand each insight type, despite having access to an API document and a study proctor. In many real-world data release scenarios (e.g., [1, 16, 17]), users would already know the types of insights they want to show and the risks they want to protect against, as this is typically a prerequisite. That said, many DSLs and declarative tools, such as D3 and ECharts, provide both API documentation and example galleries to help users understand the supported declarative content. Some participants suggested providing insight examples, both to aid comprehension of the available declarative options and to encourage further exploration of the tool's capabilities.

Budget consumption for mechanism counseling. As mentioned in Sec.4.2.2, our DP scheme formulation model determines the value of the privacy budget and selects a DP mechanism to suggest a DP scheme. Both stages consume privacy budget; the second incurs an additional cost. To provide a privacy guarantee, DP imposes strict controls on the use of information within datasets. Our model can effectively formulate high-performing DP mechanisms based on the noisy statistical results. However, accessing these results in advance consumes a portion of the privacy budget, leaving only the remaining budget available for chart generation under the suggested DP scheme. Compared to using a fixed default mechanism, the budget reduction due to mechanism selection further increases utility loss. Nevertheless, even if users attempt to manually select an optimal strategy based on prior experience, they must still expend a privacy budget to assess data characteristics. Relying exclusively on a default mechanism is also suboptimal, as our analysis reveals significant performance variations among DP mechanisms when applied to data with differing kurtosis. For example, the Laplace mechanism introduces negligible distortion to large values but substantially perturbs small values. In future work, we intend to explore how to optimize aspects of this process.

Distribution features for mechanism selection. In determining which data distribution features should inform the DP scheme formulation model, we examined both distribution features directly related to insight types and those affecting the performance of DP mechanisms. The former can be quickly identified through the variables involved in corresponding quantitative formulas. However, our review revealed a limited correlation between these features and optimal mechanism selection. For example, we constructed a dimensional space for mechanism recommendation for the significance of maximum based on data features, including the maximum, the value of the second largest, the size, and the sum. The examination results demonstrated that when there are numerous candidate mechanisms, even different data located at the same point in the dimensional space do not necessarily share an optimal mechanism. Regarding distribution features that could affect the performance of DP mechanisms, we collected empirical evidence such as DAWA, which demonstrates optimal performance when processing uniformly distributed data [30]. Through this evaluation, we identified kurtosis as a particularly discriminative feature. Our model intentionally limited the consideration of additional distributional features, as their marginal contribution to mechanism selection would not justify the corresponding privacy budget expenditure required for their queries, though future work can probe how to integrate such features while minimizing such expenditures.

7 Conclusion

We propose and implement Counterbalance, a declarative tool that enables users to generate differential privacy-preserving visualizations while maintaining desired utility requirements. Counterbalance combines a declarative framework derived from surveys of existing practices and real-world user preferences, a DP scheme formulation model that can yield adaptive DP schemes according to user declarations, and a

prototype tool validating requirement fulfillment with users. Our evaluations help demonstrate that the adaptive DP schemes generated by Counterbalance help mitigate weakening effects on insight representation compared to static mechanism implementations, and our empirical evaluations confirm the practical usability of our overall approach. In all, Counterbalance represents a visualization authoring tool that lowers technical barriers for authoring privacy-preserving visualizations, by supporting more accessible and expressive declarations of both chart designs and privacy requirements. The codebase for Counterbalance can be found at https://github.com/Vanellope7/Counterbalance.

REFERENCES

[1] J. M. Abowd. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2867–2867, 2018. doi: 10.1145/3219819.3226070 9

[2] G. Acs, C. Castelluccia, and R. Chen. Differentially Private Histogram Publishing through Lossy Compression. In *2012 IEEE 12th International Conference on Data Mining*, pp. 1–10. doi: 10.1109/ICDM.2012.80 6

[3] E. Adar and E. Lee. Communicative Visualizations as a Learning Problem. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):946–956, 2021. doi: 10.1109/TVCG.2020.3030375 2

[4] D. Avraam, R. Wilson, O. Butters, T. Burton, C. Nicolaides, E. Jones et al. Privacy preserving data visualizations. *EPJ Data Science*, 10(1):2, 2021. doi: 10.1140/epjds/s13688-020-00257-4 3

[5] L. Battle and A. Ottley. What Do We Mean When We Say "Insight"? A Formal Synthesis of Existing Theory. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6075–6088, 2024. doi: 10.1109/TVCG.2023.3326698 4

[6] K. Bhattacharjee, M. Chen, and A. Dasgupta. Privacy-Preserving Data Visualization: Reflections on the State of the Art and Research Opportunities. *Computer Graphics Forum*, 39(3):675–692, 2020. doi: 10.1111/cgf.14032 1, 2, 5

[7] S. Borzsony, D. Kossmann, and K. Stocker. The Skyline Operator. In *Proceedings 17th International Conference on Data Engineering*, pp. 421–430. IEEE, 2001. doi: 10.1109/ICDE.2001.914855 6

[8] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data-driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 2

[9] P. Bukaty. *The California Privacy Rights Act (CPRA)–An implementation and compliance guide*. IT Governance Publishing, 2021. doi: 10.2307/j.ctv1kv1d14 1

[10] J.-K. Chou, C. Bryan, and K.-L. Ma. Privacy Preserving Visualization for Social Network Data with Ontology Information. In *2017 IEEE Pacific Visualization Symposium*, pp. 11–20, 2017. doi: 10.1109/PACIFICVIS.2017.8031573 9

[11] A. Dasgupta, M. Chen, and R. Kosara. Measuring Privacy and Utility in Privacy-Preserving Visualization. *Computer Graphics Forum*, 2013. doi: 10.1111/cgf.12142 3

[12] C. Dwork. Differential Privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer, 2006. doi: 10.1007/11787006_1 2, 3, 5

[13] C. Dwork, N. Kohli, and D. Mulligan. Differential Privacy in Practice: Expose your Epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019. 2

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Third Theory of Cryptography Conference*, pp. 265–284. Springer, 2006. doi: 10.1007/978-3-540-32732-5_32 6

[15] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042 2

[16] K. El Emam and F. K. Dankar. Evaluating the utility of differentially private health data: a case study. *Transactions on Data Privacy*, 4(3):159–194, 2011. 9

[17] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067. doi: 10.1145/2660267.2660348 9

[18] B. C. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 42(4):1–53, 2010. doi: 10.1145/1749603.1749605 3

[19] M. Hardt, K. Ligett, and F. McSherry. A Simple and Practical Algorithm for Differentially Private Data Release. *Advances in Neural Information Processing Systems*, 25:2339–2347, 2012. doi: 10.5555/2999325.2999396 6

[20] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang. Principled Evaluation of Differentially Private Algorithms using DPBench. In *Proceedings of the 2016 International Conference on Management of Data*, pp. 139–154, 2016. doi: 10.1145/2882903.2882932 3 2, 6

[21] J. Heer and M. Bostock. Declarative Language Design for Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1149–1156, 2010. doi: 10.1109/TVCG.2010.144 2

[22] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce et al. Differential Privacy: An Economic Method for Choosing Epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pp. 398–410. doi: 10.1109/CSF.2014.35 6

[23] H. Jiang, J. Pei, D. Yu, J. Yu, B. Gong, and X. Cheng. Applications of Differential Privacy in Social Network Analysis: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):108–127, 2021. 3

[24] S. Jiao, J. Cheng, Z. Huang, T. Li, T. Xie, W. Chen et al. DPKnob: A Visual Analysis Approach to Risk-aware Formulation of Differential Privacy Schemes for Data Query Scenarios. *Visual Informatics*, 8(3):42–52, 2024. doi: 10.1016/j.visinf.2024.09.002 2, 3, 4, 5

[25] D. N. Joanes and C. A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998. 6

[26] M. F. S. John, G. Denker, P. Laud, et al. Decision Support for Sharing Data using Differential Privacy. In *2021 IEEE Symposium on Visualization for Cyber Security*, pp. 26–35. New Orleans. doi: 10.1109/VizSec53666.2021.00008 2

[27] M. Kay. ggdist: Visualizations of Distributions and Uncertainty in the Grammar of Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):414–424, 2024. doi: 10.1109/TVCG.2023.3327195 9

[28] M. Khavkin and E. Toch. Differential Privacy Configurations in the Real World: A Comparative Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(11):6334–6351, 2025. 3

[29] H. Kim, R. Rossi, F. Du, E. Koh, S. Guo, J. Hullman et al. Cicero: A Declarative Grammar for Responsive Visualization. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–15. doi: 10.1145/3491102.3517455 3

[30] I. Kotsogiannis, A. Machanavajjhala, M. Hay, and G. Miklau. Pythia: Data Dependent Differentially Private Algorithm Selection. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1323–1337, 2017. doi: 10.1145/3035918.3035944 6, 9

[31] P.-M. Law, A. Endert, and J. Stasko. Characterizing Automated Data Insights. In *2020 IEEE Visualization Conference*, pp. 171–175, 2020. doi: 10.1109/VIS47514.2020.00041 3

[32] J. Lee and C. Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pp. 325–340. Springer, 2011. doi: 10.1007/978-3-642-24861-0_22 3, 5

[33] C. Li, M. Hay, G. Miklau, and Y. Wang. A Data- and Workload-Aware Algorithm for Range Queries Under Differential Privacy. *arXiv preprint arXiv:1410.0265*, 2014. 6

[34] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang et al. ECharts: a declarative framework for rapid construction of web-based visualization. *Visual Informatics*, 2(2):136–146, 2018. doi: 10.1016/j.visinf.2018.04.011 2, 3, 5

[35] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan. GoTree: A Grammar of Tree Visualizations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13. Association for Computing Machinery. doi: 10.1145/3313831.3376297 2

[36] J. K. Li and K.-L. Ma. P4: Portable Parallel Processing Pipelines for Interactive Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1548–1561, 2018. doi: 10.1109/TVCG.2018.2871139 2

[37] J. K. Li and K.-L. Ma. P5: Portable Progressive Parallel Processing Pipelines for Interactive Data Analysis and Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1151–1160, 2019. doi: 10.1109/TVCG.2019.2934537 2

[38] S. LYi, Q. Wang, F. Lekschas, and N. Gehlenborg. Gosling: A Grammar-based Toolkit for Scalable and Interactive Genomics Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):140–150, Jan. 2022. Conference Name: IEEE Transactions on Visualization

and Computer Graphics. doi: 10.1109/TVCG.2021.3114876 2

[39] A. M. McNutt. No Grammar to Rule Them All: A Survey of JSON-style DSLs for Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):160–170, 2022. doi: 10.1109/TVCG.2022.3209460 1, 2, 5, 9

[40] A. Mulye. Adult Income Dataset | From Scratch. https://www.kaggle.com/code/aditimulye/adult-income-dataset-from-scratch/notebook, 2021. 8

[41] P. Nanayakkara, J. Bater, X. He, et al. Visualizing Privacy-Utility Trade-Offs in Differentially Private Data Releases. *Proceedings on Privacy Enhancing Technologies*, 2022(2):601–618, 2022. doi: 10.2478/popets-2022-0058 2, 3

[42] L. Panavas, T. Crnovrsanin, J. L. Adams, J. Ullman, A. Sargavad, M. Tory et al. Investigating the visual utility of differentially private scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):5370–5385, 2024. doi: 10.1109/TVCG.2023.3292391 3

[43] W. Qardaji, W. Yang, and N. Li. Understanding Hierarchical Methods for Differentially Private Histograms. *Proceedings of the VLDB Endowment*, 6(14):1954–1965, 2013. doi: 10.14778/2556549.2556576 6

[44] B. Saket, A. Endert, and Ç. Demiralp. Task-based Effectiveness of Basic Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(7):2505–2512, 2019. doi: 10.1109/TVCG.2018.2829750 2

[45] J. Sarathy, S. Song, A. Haque, T. Schlatter, and S. Vadhan. Don't Look at the Data! How Differential Privacy Reconfigures the Practices of Data Science. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–19. doi: 10.1145/3544548.3580791 2, 5

[46] A. Satyanarayan and J. Heer. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum*, 33(3):351–360, 2014. doi: 10.1111/cgf.12391 5

[47] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017. doi: 10.1109/TVCG.2016.2599030 2

[48] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive Vega: A Streaming Dataflow Architecture for Declarative Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):659–668, 2016. doi: 10.1109/TVCG.2015.2467091 2

[49] Z. Sun, Y. Wang, M. Shu, R. Liu, and H. Zhao. Differential privacy for data and model publishing of medical data. *IEEE Access*, 7:152103–152114, 2019. 3

[50] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. doi: 10.1142/S0218488502001648 3

[51] M. Tory. User Studies in Visualization: A Reflection on Methods. In *Handbook of Human Centric Visualization*, pp. 411–426. Springer, 2014. 8

[52] P. Voigt and A. Von dem Bussche. *The EU General Data Protection Regulation (GDPR)*. Springer Cham, 2017. doi: 10.1007/978-3-319-57959-7 1

[53] X. Wang, C. Bryan, Y. Li, R. Pan, Y. Liu, W. Chen et al. Umbra: A Visual Analysis Approach for Defense Construction against Inference Attacks on Sensitive Information. *IEEE Transactions on Visualization and Computer Graphics*, 28(7):2776–2790, 2022. doi: 10.1109/TVCG.2020.3037670 5, 9

[54] X. Wang, W. Chen, J.-K. Chou, C. Bryan, H. Guan, W. Chen et al. GraphProtector: A Visual Interface for Employing and Assessing Multiple Privacy Preserving Graph Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):193–203, 2019. doi: 10.1109/TVCG.2018.2865021 3, 4, 5

[55] X. Wang, J.-K. Chou, W. Chen, H. Guan, W. Chen, T. Lao et al. A Utility-aware Visual Approach for Anonymizing Multi-attribute Tabular Data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):351–360, 2018. doi: 10.1109/TVCG.2017.2745139 2, 3, 4, 5

[56] X. Wang, S. Jiao, and C. Bryan. Defogger: A visual analysis approach for data exploration of sensitive data protected by differential privacy. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3

[57] X. Xiao, G. Wang, and J. Gehrke. Differential Privacy via Wavelet Transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, 2011. doi: 10.1109/TKDE.2010.247 6

[58] Y. Xiao, L. Xiong, L. Fan, and S. Goryczka. DPCube: Differentially Private Histogram Release through Multidimensional Partitioning. *arXiv preprint arXiv:1202.5358*, 2012. 6

[59] J. Xu, Z. Zhang, X. Xiao, et al. Differentially Private Histogram Publication. In *2012 IEEE 28th International Conference on Data Engineering*, pp. 32–43. Arlington. doi: 10.1109/ICDE.2012.48 2

[60] D. Zhang, A. Sarvghad, and G. Miklau. Investigating Visual Analysis of Differentially Private Data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1786–1796, 2021. doi: 10.1109/TVCG.2020.3030369 3

[61] X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie. Towards Accurate Histogram Publication under Differential Privacy. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 587–595. doi: 10.1137/1.9781611973440.68 6

[62] Y. Zhao, Y. Zhang, Y. Zhang, X. Zhao, J. Wang, Z. Shao et al. LEVA: Using Large Language Models to Enhance Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 31(3):1830–1847, 2025. doi: 10.1109/TVCG.2024.3368060 3

[63] J. Zhou, X. Wang, J. K. Wong, et al. DPVisCreator: Incorporating Pattern Constraints to Privacy-preserving Visualizations via Differential Privacy. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):809–819, 2023. doi: 10.1109/TVCG.2022.3209391 2, 3, 4

[64] T. Zhu, G. Li, W. Zhou, and S. Y. Philip. Differentially Private Data Publishing and Analysis: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, 2017. doi: 10.1109/TKDE.2017.2697856 3