Declarative AI-Assisted Range Aggregation Query Framework with Differential Privacy

Hong Guan, Yancheng Wang, Yingzhen Yang, Chaowei Xiao[†], Jia Zou
Arizona State University, University of Wisconsin at Madison[†]

***Abstract*—Using Differential Privacy (DP) to provide a formal privacy guarantee for range aggregation queries has attracted tremendous attention recently given the increasing privacy concerns and the wide usage of range aggregation queries in internet-of-things (IoT) applications, spatial-temporal data management, interactive analytics and visualizations, etc. Our vision for advancing this field aligns with two key trends: (1) AI-Assisted: Applying AI/ML models to replace data to answer range aggregation queries can better balance the privacy costs, accuracy (utility), and query speed, than existing approaches; and (2) Declarative Workflow: Current methods for implementing DP in privacy-preserving range aggregation queries excel in specific scenarios, and thus it is crucial to develop a declarative privacy-aware query optimizer capable of automatically identifying the optimal approach to ensure DP guarantees. By embracing these trends, we foresee a redefinition of database query optimizers from a privacy-focused perspective. This shift will address existing challenges in integrating DP with data management, simplify adoption, and enhance the effectiveness of DP in a wide range of data-intensive applications.**

I. Introduction

Range aggregation queries play a pivotal role in data analytics and database systems [1], facilitating the summarization, analysis, and visualization [2] of data across numeric, spatial, or temporal intervals. They are extensively employed across diverse domains such as IoT [3], finance [4], healthcare [5], [6], and e-commerce. Ensuring privacy preservation in range aggregation queries is crucial for responsibly leveraging sensitive data, enabling organizations to conduct meaningful analyses while safeguarding privacy, adhering to regulations, and maintaining trust [7].

Among privacy-preserving techniques, Differential Privacy (DP) [8] stands out for range aggregation queries over private data due to its unique advantages:

• **Strong Privacy Guarantees**: DP ensures that the inclusion or exclusion of any individual's data has a negligible impact on the query results, minimizing the risk of re-identification or data inference attacks. This compliance aligns with regulations such as the General Data Protection Regulation (GDPR) [9] and the California Consumer Privacy Act (CCPA) [10].

• **Efficiency and Scalability**: DP can be implemented efficiently for range aggregation queries, enabling scalability to large datasets with lower computational overhead than techniques like secure multi-party computation (SMPC) [11] or cryptographic operations [12].

• **Configurable Privacy-Accuracy Trade-offs**: DP's configurable privacy budgets (e.g., parameter ϵ) allow data analysts to balance privacy and accuracy effectively.

Most state-of-the-art (SOTA) DP-based range aggregation query methods [13]–[17] rely on data partitioning to balance privacy and accuracy. For instance, techniques like recursive, adaptive, or hierarchical partitioning assume uniform distributions of events within bins. They estimate the count of events per bin, add noise, and compute aggregate results using ratios of overlapping areas. However, such methods often yield significant errors when applied to real-world datasets with skewed distributions.

AI/ML offers promising advancements for DP-enabled range aggregation query processing [18], [19]. For example, in an aggregation query like `SELECT count() FROM T WHERE a < A <= a + size AND b < B <= b + size`, AI/ML models can predict `count()` based on input features (`a, b, size`). By adding Laplace noise to training data, privacy can be preserved while leveraging correlations between query parameters and results [18]. This approach improves accuracy for non-uniform data distributions under the same privacy budget (ϵ).

Despite its potential, integrating AI/ML with DP-based range aggregation remains nascent due to key challenges:

1. Modeling Challenges:

(C1-1) Addressing data updates, insertion, and deletion is complex due to issues like catastrophic forgetting and the difficulty of accurately unlearning deleted data.

(C1-2) Modeling large-scale complex data is challenging, as achieving high accuracy can worsen the privacy-accuracy trade-off compared to simpler approaches like adding noise to query outputs or datasets.

2. Declarative Optimization Challenges:

(C2-1) Not all information in a dataset is sensitive. It is important to allow data owners to easily specify sensitive attributes and tuples in a fine-grained way (i.e., to specify *what to protect*), and allow the declarative optimizer to automatically figure out *how to protect* the fine-grained sensitive information.

(C2-2) There are multiple ways to model data for range aggregation queries. For instance, models may map arbitrary ranges to aggregated values or compute cumulative aggregations across fixed dimensions. Determining the optimal modeling approach automatically is nontrivial.

(C2-3) Selecting the optimal model architecture (e.g., through neural architecture search) depends on factors like data modality, distribution, and the availability of pre-trained models, posing significant challenges. Position: Advances in AI/ML hold great promise for balancing privacy, accuracy, and efficiency in DP-enabled range aggregation query processing. For example, the growing accuracy of AI/ML architectures (e.g., foundation models) can enable high-accuracy results on private data with minimal privacy budgets. These methods can extend to queries over structured, semi-structured, and unstructured data, including images and text. However, realizing this potential requires overcoming significant hurdles, including the lack of declarative optimization frameworks, neural architecture search capabilities, support for updates, and scalability to complex, large-scale datasets.

The remainder of this paper is organized as follows: we discuss modeling challenges in Section II and declarative optimization challenges in Section III. In Section IV, we explore extending the framework to a new class of queries combining SQL with AI/ML model inferences, termed inference queries. Related works are reviewed in Section V, and we conclude in Section VI.

II. Modeling Challenges and Opportunities

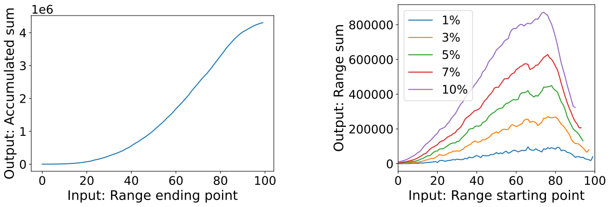
In this section, we first discuss promising model strategies based on ranges, density distribution, and cumulative distribution respectively. Then, we will discuss the challenges and opportunities in improving the accuracy and spatial efficiency of such modeling.

A. Overall Modeling Strategies

Given a one-dimensional range sum query q in the template of `SELECT sum(y) FROM T WHERE x >= sp AND x < ep`, with y being the dependent variable and x being the independent variable, following an underlying function $y = f(x)$, we can learn three different functions as detailed below:

Histogram Distribution. An intuitive idea is to learn a histogram sum function $hs(start\_point, range\_size) = \int_{start\_point}^{start\_point+range\_size} f(x)\ dx$, or $hs(start\_point, range\_size) = \sum_{start\_point}^{start\_point+range\_size} f(x)\ dx$, if x is discrete. In this case, we may generate training data for each different histogram bin size, so that every training sample has features being the $start\_point$ and the $range\_size$ of the query range (i.e., bin size), with the label being the sum of data points falling in the query range (i.e., bin). This approach is an extension of Spatial Noisy Histogram [20], which focuses on privacy-preserving range count queries on spatial-temporal data. Then, the sum aggregation result falling in the range $[sp, ep)$ will be $hs(sp, ep - sp)$.

Cumulative Distribution. To answer the range sum queries, we can also learn a cumulative sum function $cs(end\_point) = \int_{origin\_point}^{end\_point} f(x)\,dx$. Here, $origin\_point$ is usually chosen as the minimal value of the independent variable x. Then, the answer to the query q can be represented as $cs(ep) - cs(sp)$. It also holds if x is discrete, where the cumulative sum is represented as $cs(end\_point) = \sum_{x=origin\_point}^{end\_point} f(x)$. In this case, we will train a model, for which each training sample has its feature represented as the `end_point` of a one-dimensional range, while the label is the cumulative sum of the dependent variable from the `origin_point` (e.g., the minimum value of the independent variable) to the `end_point`. This approach is first proposed by us, inspired by the observation that the cumulative distribution is smoother and easier to learn than the histogram distribution, as illustrated in Fig. 1.



(a) Cumulative Sum

(b) Histogram Sum

Fig. 1: Comparison of cumulative sum distribution and range/histogram sum distribution on the combined cycle power plant dataset [] that consists of four attributes: AT for ambient temperature, AP for ambient pressure, RH for relative humidity, and PE for net hourly energy output. The x-axis shows one of the 1-D AT range boundaries, the y-axis in (a) shows the sum of PE events when AT's value starts from 0 to the point specified by the x-axis, and the y-axis in (b) shows the sum of PE events when AT's value starts from the point specified by the x-axis and ends at x*(1+query_range).

Density Distribution and Regression Function. An alternative is to learn a density function $D(x)$ and a regression function $R(x)$. $R(x)$ is trained from $(x, f(x))$ pairs. Then, to answer a sum query with range $[sp, ep)$, we have $s(sp, ep) = \int_{sp}^{ep} D(x)R(x)\ dx$. This approach is proposed for approximate query processing without considering DP [21].

There are two strategies of adding noises to the above modeling strategies to protect the personal information in the underlying data with DP. One approach is to add noises to the training data and the other approach is to leverage the Differential Privacy Stochastic Gradient Descent (DP-SGD) [22] to train the models. However, it is usually considered DP-SGD to increase the sensitivity for aggregation queries since adding or removing a sample to the batch of training samples may affect the value of every gradient while adding noises to the underlying data will avoid such issues [18].

Research Opportunity 1. Comparison and Improvement of Modeling Techniques

The first research question is to better understand the privacy-accuracy trade-offs, advantages, and disadvantages of each modeling strategy. It requires more theoretical analysis, benchmark design, and evaluations. There also exist research opportunities in developing new modeling approaches (e.g., new model architectures) leveraging the advancement in AI/ML.

B. C1-1: Data Insertion and Deletion

A significant challenge posed by the idea of modeling data and using the AI/ML model(s) for answering range aggregation queries is how to manage data manipulations such as insertion, deletion, and updates. Supporting such operations will enable more application scenarios that require maintaining the states of individuals, such as aggregation queries over healthcare records, financial records, student records, etc.

However, inserting, changing, and deleting a training sample from the training dataset is extremely challenging. Inserting training samples through incremental training will suffer from the catastrophic forgetting issue [23], while deleting a training sample, i.e., unlearning, is also challenging [24].

Retraining the models for each data update is expensive. One potential approach is to develop an auxiliary data store to cache such data changes before running a periodic retaining process [25]. Therefore, to answer each range aggregation query, we will not only run the model predictions but also query the auxiliary structure and then combine the results. Although such an approach has been adopted for modeling data for look-up queries, it is non-trivial to extend the idea for aggregation queries. First, one newly inserted/updated/deleted data record may change multiple training samples, if histogram distribution (with different bin sizes), cumulative distribution, and density distribution are to be learned. Second, noises should be added to the auxiliary data to ensure privacy guarantee. Third, for multi-dimensional continuous independent variables, the volume of the auxiliary data structure could be overwhelming, leading to long latency in querying the auxiliary data structure.

Another potential approach is to leverage new AI/ML advances that address the catastrophic forgetting and unlearning issues. For example, to overcome catastrophic forgetting issue during incremental training, replay-based approaches will select and store a subset of features that best approximate the class mean and they are used to supervise incremental learning [26], [27]. Another approach is regularization based, which applies regularization terms to constrain the parameter updates in order to preserve previous knowledge during the incremental learning process [23], [28]. Dynamic architectures are also designed to alleviate the catastrophic forgetting issue [29], [30].

In addition, we may also leverage AI/ML unlearning techniques [31], [32] for deleting data from the model. For example, the gradient ascent approach inversely minimizes the likely hood to diverge the model's predictions from the correct information for the examples that need to be forgotten [31].

Research Opportunity 2. Data Insertion and Deletion

To support flexible data manipulation, it is promising to develop a new novel auxiliary data structure to cache update operations. It is also promising to leverage new AI/ML advances that address the catastrophic forgetting and unlearning issues.

C. C1-2: Scaling to Large-Scale Complex Data

We observed that it could be challenging to learn a single model to learn the models discussed in Sec. II-A over large-scale datasets that may have diverse patterns and multiple dimensions. To address the challenge, one approach is to partition the dataset leveraging the sum-product network (SPN) [33], [34] so that each leaf node of a sum-product tree corresponds to a partition of the dataset, and then we learn a model on each partition. In an SPN network, a dataset can be horizontally partitioned through sum nodes and vertically partitioned through product nodes, forming a general tree structure, while each leaf node represents a fragment of the dataset that has similar patterns so that the data distributions existing in the fragment can be easily captured by an AI/ML model. At each level, the splitting point should be carefully selected to minimize the mutual information across the partitions. The tree structure of SPN will also bring the interpretability for the learned models.

Another approach is to leverage the Mixture-of-Experts (MoE) [35] to learn a network of multiple experts of which decisions are mitigated through a gateway model.

Research Opportunity 3. Modeling Large-Scale Complex Data

To scale the modeling approach to large complex data, it is promising to adopt a divide-and-conquer approach by partitioning the underlying data into many fragments and learning a model for each fragment, which may require new research on more sophisticated partitioning strategies. It may also trigger more research to apply cutting-edge AI/ML techniques such as MoE, attention mechanisms, and foundation models, to improve the modeling accuracy.

III. Declarative Optimization Challenges and Opportunities

A. Overview of Declarative Workflow

Significant challenges lie in the complexity of identifying the sensitive part of the query to be replaced, the model architecture for replacing the sensitive sub-query, and the hyper-parameters for training the model. It would require a lot of human resources to manually figure out the optimal scheme and configurations for each query. The lack of declarative and automated tooling poses significant challenges for our proposed AI-range aggregation query framework with DP guarantees. To address this problem, we further advocate for a novel privacy-preserving workflow. The workflow uses taint analysis to automatically identify sensitive sub-queries (Sec. III-B). After that, the query will be automatically analyzed and transformed into a model-assisted plan for balancing privacy and accuracy (Sec. III-C). Then, an AutoML model search process will be invoked to efficiently identify the optimal model search architecture (Sec. III-D) that would strike a desired balance among privacy budget, utility (i.e., accuracy), and efficiency. The workflow further allows human experts to review and refine the selected plans.

(1) Privacy Information Specification. The data owners are allowed to label the attributes and tuples to specify private information that needs to be protected. In addition, the view-based access control policies as supported in most relational database management systems (RDBMSs), can also be used as input for the proposed system to understand the privacy requirements of the relations.

(2) Query Intermediate Representation (IR) and Taint Analysis. The data scientists will issue queries over sensitive datasets, which will be automatically lowered to an IR based on relational algebra to efficiently identify sub-queries that involve sensitive attributes or tuples through taint analysis [36].

(3) Privacy-Preserving Transformation Enumeration and Optimization. The system will automatically enumerate possible privacy-preserving schemes based on the proposed model-query transformation technique. Those schemes can be applied to the sensitive sub-query as a transformation to the IR. A cost-based optimizer will evaluate the computational costs of each candidate (logical) plan and select the optimal plan using a cost model and an optimization algorithm.

(4) Efficient Model Search based on Transfer Learning and Neural Architecture Search. Given a query-model transformation request (e.g., as recommended by the optimizer), the system will first search for a pre-trained model that can be directly reused or be fine-tuned to implement the transformation. Then, it will search for the adaptive layer(s) [37], [38] to be inserted into the pre-trained model to meet the accuracy, privacy, and latency requirements. If there are no existing pre-trained models that match the task, then we use neural architecture search (NAS) [39] to obtain a deep learning architecture and then train the model from scratch.

(5) Feedback. Finally, the system administrator or a system program can review and modify the selected privacy-preserving mechanism for audit/compliance and optimization purposes. The final query execution latency and accuracy will also be recorded and used to refine the cost model. Next, we will discuss the research challenges and opportunities related to key components of the worklfow.

B. *Taint Analysis*

A key component of the proposed declarative workflow is taint analysis, where sensitive attributes and tuples are directly specified by the data owner. Then once a data scientist issues a query, the query will be lowered to a graph-based Intermediate Representation (IR) based on the nested relational algebra, where each node represents a relational operator, and each edge represents a relation. The tainted data sources will propagate the taint through the IR graph so that the sensitive sub-queries that access private information are identified. While the tainting and tracking of sensitive attributes could be easily supported, a research challenge lies in fine-grained tainting of sensitive tuples and propagating tuple-level taints across the query graph. To achieve fine-grained tainting and taint propagation, one straightforward approach is to materialize the results at every operator and check whether they contain sensitive tuples. However, this approach is not only expensive but also may incur additional privacy costs since it requires scanning these personal tuples to check whether they exist in the intermediate results. A more promising approach is to create a view that only includes sensitive tuples, termed sensitive view, and a view of the dataset that excludes those sensitive tuples, termed insensitive view. Then, the scan of the dataset in all user queries will be rewritten to the scan of a union of the sensitive view and the insensitive view. In this way, all the operators that access the tainted sensitive attributes and tuples from the sensitive view will be tainted.

Research Opportunity 4. Fine-grained Taint Analysis

By partitioning each source dataset and separating sensitive tuples from insensitive tuples, we may map the user query to the relational algebra graph as an IR, and leverage the IR graph analysis to detect the sensitive range aggregation queries.

C. *Query Transformation Rules*

When the query parser detects a range aggregation query that accesses sensitive information, such as `SELECT sum(y) FROM T WHERE x >= sp AND x < ep`, we will enumerate all transformations that may convert the query to a form that leverages the modeling techniques as described in Sec. II-A to better balance privacy and utility.

For example, the above query can be converted into `SELECT s(ep)-s(sp)`, where $s(p)$ represents an ML model that predicts the cumulative sum of attribute y of each data point $t \in T$, where $t.x > o$ and $t.x < p$. Here o represents the origin point (e.g., the minimal value of attribute x. Such a transformation rule can be formalized as Eq. 1

$$\Sigma(\pi_y(\sigma_{(x>=sp)\wedge(x<ep)}(T))) \overset{\Delta auc,\epsilon^*}{=} cs(sp) - cs(ep) \quad (1)$$

Using this transformation rule, the query $\Sigma(\pi_y(\sigma_{(x>=sp)\wedge(x<ep)}(T)))$ is transformed into $s(sp) - s(ep)$ with the maximal privacy budget caused by this transformation must be constrained by ϵ^*, and the accuracy drop must be constrained by Δauc. The ϵ^* and Δauc could be determined according to the contract, or service-level agreement (SLA), between the user (i.e., query issuer) and the platform, which defined the price the user would like to pay to compensate a certain privacy loss (i.e., ϵ^*) at the data owner side, and the amount of accuracy (Δauc) the user would like to give up for the service. Later, the query optimization and the neural architecture search process will attempt to search for a model to satisfy these constraints.

Similarly, we can have other transformation rules for different modeling techniques as discussed in Sec. II-A, as shown in Eq. 2 and Eq. 3.

$$\Sigma(\pi_y(\sigma_{(x>=sp)\wedge(x<ep)}(T))) \overset{\Delta auc,\epsilon^*}{=} hs(sp, ep - sp) \quad (2)$$

$$\begin{aligned}&\Sigma(\pi_y(\sigma_{(x>=sp)\wedge(x<ep)}(T))) \overset{\Delta auc,\epsilon^*}{=} \\ &\Sigma\pi_{D(x)R(x)}\sigma_{(x>=sp)\wedge(x<ep)}T\end{aligned} \quad (3)$$

It is challenging to estimate the accuracy, privacy budget, and latency if we apply a transformation, and decide whether it can meet the SLA. Usually, we can only obtain these metrics through evaluation after the model training component returns the model(s) for each transformation, which is time-consuming. It requires an approach to predict the costs of a transformation without applying the model search and training process.

To address the challenge, we train a surrogate model for each type of workload, which predict the costs given a transformation plan. Then we may use Bayesion optimization to search for the optimal plan that minimizes the costs through an acquisition function while keeping updating the surrogate models.

Given the costs predicted by each surrogate model for each objective, it is possible to formulate a multi-objective cost representation for each query transformation plan to search for Pareto-optimal plans using an evolutionary algorithm [40]. Motivated by the observation that a lot of Pareto-optimal plans are like each other in the costs for all objectives, we fruther reduce the query optimization time by only returning representative Pareto-optimal logical plans. Then for SLA provided by the user, the optimizer will efficiently return a Pareto-optimal plan that represents the best trade-off.

> Research Opportunity 5. Automatic Optimization
>
> The optimal transformation plan depends on the aggregation function, the number dimensions in defining the range, the continuity of the range, the underlying data size and patterns, etc. It would be valuable to explore how all these factors affect the privacy, utility, and efficiency. It is important to automatically search for the optimal transformation plan.

D. *Neural Architecture Search*

Given a selected transformation plan, it is important to configure the neural architecture(s) and hyper-parameters required for implementing the transformation plan. We first consider reusing models pre-trained on public datasets to reduce the training time [41]. Particularly, to improve the trade-off between accuracy and the privacy budget, it is important to finetune a foundation model [42] that is trained on large-scale public datasets, e.g. DeepFace [43] and FaceNet [44] for facial analysis, GPT-3 [45] , BERT [46] for natural language processing, TabTrasformer [47] for tabular data, ALIGN [48] , CLIP [49], and DALL-E-2 [50] for multi-modal data. The benefits include (1) we can leverage the foundation model to achieve good accuracy while using only a small private training dataset; (2) the fewer number of training epochs further leads to smaller privacy budget [22].

There are multiple strategies to finetune a foundation model on private datasets. One example is to dynamically train some layers and freeze other layers at every epoch [51], or train a small adapter network while freezing the parameters in the foundation model, such as LORA [38].

To find the optimal architecture of the neural network model with minimal human effort, leveraging our prior work [19], we may consider a Differentiable Neural Architecture Search (DNAS) algorithm [52]. When a foundation model is found [41], DNAS will be used to search for layers to be trainable and/or the size of the adaptive layers; otherwise, DNAS will search for the architecture of a new model based on the modality of the data. In DNAS, the choices of various neural operations are characterized by a set of architecture parameters defined in a graph that consists of all possible layers. DNAS searches for the optimal network architecture through gradient descent by optimizing both the prediction loss and the computation cost of the neural networks with respect to the architecture parameters and the network weights.

One significant challenge is that the model search process will also introduce private information leaking. To address the issue, once the search is complete, we define the architecture of the neural network based on the optimized architecture parameters. Next, we *reset the weights of the searched neural network to the initial random state* and train it on the private data using DP-SGD. Note that although DNAS is conducted on private data, it does not increase the privacy budget as the network weights of the supergraph will not be released.

> Research Opportunity 6. Neural Architecture Search for Modeling Data to Answer Aggregation Queries
>
> Given a transformation plan, it is important to search for optimal neural architecture(s) and hyper-parameters to implement the transformation plan leveraging state-of-the-art neural architecture search techniques.

IV. Extension to Inference Queries

With more database systems supporting AI/ML, queries increasingly involve AI/ML model inferences. However, limited research has addressed declarative support for differential privacy (DP) in inference queries, particularly when the underlying dataset contains sensitive information. Ideally, with declarative DP support, a data owner will declare sensitive information that needs to be protected by tainting the data attributes and tuples. Database users could then issue arbitrary inference queries without needing to define the specifics of the DP mechanism or model. Instead, the system would automatically identify sensitive subqueries and apply DP safeguards based on the user's privacy budget. We use an example to illustrate the idea as follows.

Motivating Example. Online social media posts may disclose personal information such as use patterns, life habits, and social status. Linkage attacks can relate the information in the online post to private information. Consider the following query, "*SELECT * FROM IMDB_MOVIE_REVIEW R WHERE sentiment_classifier(R.Review) = Positive* ". Instead of directly sharing private reviews with business analysts or data scientists, the social media company can use Differential Privacy-Stochastic Gradient Descent (DP-SGD) [22] to train a sentiment classification model that takes a review as input and outputs its sentiment. DP-SGD ensures, with high probability, that business analysts cannot reconstruct any private training data from the model. As a result, the system will use the privacy-preserving model to answer the query, which often achieves better privacy-utility trade-offs than alternatives, such as adding perturbations to the aggregation results. The privacy cost of the query is constrained by the remaining privacy budget of both the dataset and the user.

We argue that the research questions we discussed for range aggregation queries can also extend to inference queries. Taking neural architecture search as an example, for the above movie review count query, we may fine-tune a pre-trained BERT model on the private social media posts using DP-SGD, while we can also train a bi-LSTM model with pretrained Word2Vec embeddings from scratch using DP-SGD. Alternatively, a pre-trained deep learning model can be used to encode each data point into an embedding vector, with noise added to ensure differential privacy. In this case, inference is performed using approximate nearest neighbor search. Using the IMDB dataset, we found that the privacy-utility trade-offs of the first two approaches outperformed the third approach. The first approach (i.e., fine-tuned a pre-trained BERT model using DP-SGD) achieved better accuracy than the second approach (i.e., training a bi-LSTM model from scratch using DP-SGD) for small privacy budgets (i.e., $\epsilon < 6$), while the second approach outperformed the first approach for other cases.

We abstract the process of providing DP support for a sensitive inference query as a special type of query transformation rules, described as follows (using the classical relational algebra notations): (1) $\lambda_f R \overset{\Delta acc,\epsilon}{=} \lambda_{f'} R$. λ_f represents the prediction operator that takes each tuple in the relation (or a collection of arbitrary objects) R as input features, and outputs prediction results. The DP-SGD-trained model, denoted as $\lambda_{f'} R$, outputs the prediction results with privacy guarantee ϵ. In addition, such transformation may results in an accuracy drop, represented as Δacc. (2) $\lambda_f(\pi_A(\sigma_p(R))) \overset{R,\Delta acc,\epsilon}{=} \lambda_{f'}(A,p)$. λ_f represents the prediction operator taking the query output $\pi_A(\sigma_p(R))$ as input features. The model trained with DP-SGD, denoted as $\lambda_{f'}(A,p)$, which takes the projection attributes A and selection predicate p as inputs, and outputs the final prediction result for $\lambda_f(\pi_A(\sigma_p(R)))$. We have similar rules extended for aggregation queries, which are omitted due to space limitation. A query optimizer searches for the most promising transformation plan using a learned cost model.

Research Opportunity 7. Extension to Other Types of Queries

We argue that the proposed declarative workflow can extend to other types of queries that require DP guarantees over sensitive underlying data. Taking inference queries as an example, the proposed declarative workflow may automatically detect model inferences over sensitive data (which may be extracted from a collection of insensitive and sensitive data sources through SQL queries) and automatically search for neural architectures to be trained using the DP-SGD algorithm, and thus alleviate the human efforts and improve the trade-offs between privacy and utility.

V. Related Works

SNH [20] learns a deep neural network model on noisy histograms with varying granularities to map the query range coordinates and range size to a noisy count in the range. However, SNH is limited to range count queries in spatial-temporal workloads, and it does not consider other types of range aggregation queries such as sum, for which the histograms are not as helpful as cumulative distributions.

Modeling techniques have also been applied to approximate query processing (AQP) for range aggregation queries that do not consider DP. DBest [21] uses kernel density model for density estimation and an ensemble of XGBoost and GBoost as the regression model. Ma et al. [53] assumes the underlying data is drawn from a mixture Gaussian distribution and models it by a neural network named Mixture Density Network (MDN). These models are combined with catelog information to answer approximate queries. DeepDB [34] learns a Relational Sum Product Network (RSPN) to model the joint distribution of multiple variables. Unlike many other neural networks, RSPN is a tree-structure network whose leaves correspond to a table column, making the network more interpretable. Thirumuruganathan et al. [54] learns the underlying data distribution with a Variational Auto-Encoder (VAE) and generates sample to approximate query results. However, none of these works have considered differential privacy (DP).

DeepMapping [25] has applied modeling techniques to compress data and accelerate look-up queries. It integrates a parameter-efficient algorithm to search for a hybrid structure that combines neural architecture models and auxiliary data structures. However, DeepMapping did not consider DP.

VI. Conclusion

Range aggregation queries play a critical role in enabling data analysis while preserving privacy, especially in domains like healthcare, finance, and social sciences, where sensitive data is prevalent. Efficiently answering these queries with DP guarantees ensures that insights can be derived without compromising individual privacy.

In this work, we present our vision of two trends in range aggregation queries over sensitive data: (1) Applying AI/ML techniques to model noisy histogram distributions, noisy cumulative distributions, noisy density distributions, and regression models of the noisy underlying data and use these models to answer range aggregation queries with DP guarantees. (2) Providing a declarative workflow so that we only need data owners to label the sensitive attributes and tuples in their dataset, and a query issued by the data consumers will be automatically analyzed to identify the sensitive subqueries, and automatically transformed into a new execution plan with optimized modeling techniques applied.

In conclusion, our vision aims to bridge the gap between the growing demand for privacy-preserving data analysis and the complexity of implementing such techniques. By leveraging AI/ML models and automation, we discuss various techniques for scalable, secure, and efficient handling of range aggregation queries, fostering trust and enabling broader adoption of privacy-preserving data analytics across various industries. Furthermore, we argue for the integration of automated workflows to reduce the burden on data owners and data sharing/query platform operators and simplify the process of implementing privacy-preserving analytics.

References

[1] C.-T. Ho, R. Agrawal, N. Megiddo, and R. Srikant, "Range queries in olap data cubes," *ACM SIGMOD Record*, vol. 26, no. 2, pp. 73–88, 1997.

[2] L. Wang, G. Wang, and C. A. Alexander, "Big data and visualization: methods, challenges and technology progress," *Digital Technologies*, vol. 1, no. 1, pp. 33–38, 2015.

[3] Z. Cai, X. Zheng, J. Wang, and Z. He, "Private data trading towards range counting queries in internet of things," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4881–4897, 2022.

[4] A. Dobra, M. Garofalakis, J. Gehrke, and R. Rastogi, "Processing complex aggregate queries over data streams," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pp. 61–72, 2002.

[5] J. N. S. Rubí and P. R. L. Gondim, "Iomt platform for pervasive healthcare data aggregation, processing, and sharing based on onem2m and openehr," *Sensors*, vol. 19, no. 19, p. 4283, 2019.

[6] S. Han, S. Zhao, Q. Li, C.-H. Ju, and W. Zhou, "Ppm-hda: privacy-preserving and multifunctional health data aggregation with fault tolerance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1940–1955, 2015.

[7] T. Kulkarni, "Answering range queries under local differential privacy," in *Proceedings of the 2019 International Conference on Management of Data*, pp. 1832–1834, 2019.

[8] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*, pp. 1–12, Springer, 2006.

[9] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[10] S. L. Pardau, "The california consumer privacy act: Towards a european-style privacy regime in the united states," *J. Tech. L. & Pol'y*, vol. 23, p. 68, 2018.

[11] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, no. 110, pp. 1–108, 1998.

[12] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[13] W. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in *2013 IEEE 29th international conference on data engineering (ICDE)*, pp. 757–768, IEEE, 2013.

[14] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially-private histograms through consistency," *arXiv preprint arXiv:0904.0942*, 2009.

[15] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *2012 IEEE 28th International Conference on Data Engineering*, pp. 20–31, IEEE, 2012.

[16] J. Zhang, X. Xiao, and X. Xie, "Privtree: A differentially private algorithm for hierarchical decompositions," in *Proceedings of the 2016 international conference on management of data*, pp. 155–170, 2016.

[17] 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, IEEE, 2012.

[18] S. Zeighami, R. Ahuja, G. Ghinita, and C. Shahabi, "A neural database for differentially private spatial range queries," *arXiv preprint arXiv:2108.01496*, 2021.

[19] L. Zhou, K. S. Candan, and J. Zou, "Deepmapping: The case for learned data mapping for compression and efficient query processing," *arXiv preprint arXiv:2307.05861*, 2023. To appear in ICDE 2024.

[20] S. Zeighami, R. Ahuja, G. Ghinita, and C. Shahabi, "A neural database for differentially private spatial range queries," *Proceedings of the VLDB Endowment*, vol. 15, no. 5, pp. 1066–1078, 2022.

[21] Q. Ma and P. Triantafillou, "Dbest: Revisiting approximate query processing engines with machine learning models," in *Proceedings of the 2019 International Conference on Management of Data*, pp. 1553–1570, 2019.

[22] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[24] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, IEEE, 2021.

[25] L. Zhou, K. S. Candan, and J. Zou, "Deepmapping: Learned data mapping for lossless compression and efficient lookup," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1–14, IEEE, 2024.

[26] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[27] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Machine Intelligence*, vol. 1, no. 8, pp. 364–372, 2019.

[28] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[29] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

[30] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International conference on machine learning*, pp. 4548–4557, PMLR, 2018.

[31] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *arXiv preprint arXiv:2310.10683*, 2023.

[32] B. Liu, Q. Liu, and P. Stone, "Continual learning and private unlearning," in *Conference on Lifelong Learning Agents*, pp. 243–254, PMLR, 2022.

[33] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 689–690, IEEE, 2011.

[34] B. Hilprecht, A. Schmidt, M. Kulessa, A. Molina, K. Kersting, and C. Binnig, "Deepdb: Learn from data, not from queries!," *arXiv preprint arXiv:1909.00607*, 2019.

[35] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[36] J. Clause, W. Li, and A. Orso, "Dytan: a generic dynamic taint analysis framework," in *Proceedings of the 2007 international symposium on Software testing and analysis*, pp. 196–206, 2007.

[37] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, pp. 2790–2799, PMLR, 2019.

[38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[39] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.

[40] C. A. C. Coello and G. B. Lamont, *Applications of multi-objective evolutionary algorithms*, vol. 1. World Scientific, 2004.

[41] L. Zhou, A. Jain, Z. Wang, A. Das, Y. Yang, and J. Zou, "Benchmark of dnn model search at deployment time," in *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*, 2022.

[42] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[43] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[44] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[45] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[47] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, “Tabtransformer: Tabular data modeling using contextual embeddings,” *arXiv preprint arXiv:2012.06678*, 2020.

[48] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” in *International Conference on Machine Learning*, pp. 4904–4916, PMLR, 2021.

[49] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[50] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *arXiv preprint arXiv:2204.06125*, 2022.

[51] R. Pan, X. Liu, S. Diao, R. Pi, J. Zhang, C. Han, and T. Zhang, “Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning,” *arXiv preprint arXiv:2403.17919*, 2024.

[52] A. Wan, X. Dai, P. Zhang, Z. He, Y. Tian, S. Xie, B. Wu, M. Yu, T. Xu, K. Chen, *et al.*, “Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions,” in *CVPR*, 2020.

[53] Q. Ma, A. M. Shanghooshabad, M. Almasi, M. Kurmanji, and P. Triantafillou, “Learned approximate query processing: Make it light, accurate and fast,” in *Conference on Innovative Data Systems,(CIDR21)*, 2021.

[54] S. Thirumuruganathan, S. Hasan, N. Koudas, and G. Das, “Approximate query processing for data exploration using deep generative models,” in *2020 IEEE 36th international conference on data engineering (ICDE)*, pp. 1309–1320, IEEE, 2020.