A Metric Definitions and Instantiation Guidelines

A.1 Key Node Dependency (KND)

Definition KND measures the distributional distance of node pair dependencies between the synthetic and original data. For a key node pair (O_i, O_j), let $C_{i,j}$ be the cosine similarity between their embeddings, and let $\omega_{C_{i,j}}$ and $\omega'_{C_{i,j}}$ be the distributions of these similarities in the original and synthetic data, respectively. Then, KND is defined as:

$$\text{KND}(O_i, O_j) = Dis(\omega_{C_{i,j}}, \omega'_{C_{i,j}}),$$

where Dis is the Wasserstein-2 distance.

Instantiation Guideline We allow the user to specify key nodes. If not specified, all nodes parsed by CFG are treated as key nodes by default. To instantiate key nodes, we recommend users ask the question *"Which nodes are central to our downstream tasks, and which nodes are semantically related to them?"*. For example, key node pairs could be a query and response in a conversation dataset, or a review and its rating in a product review dataset. We've specified the key nodes of our datasets in Table 1.

A.2 Attribute Match (AM)

Definition AM calculates the distributional distance of a given attribute between the synthetic and original data. For attribute a, let ω_a and ω'_a denote its distributions in the original and synthetic data, respectively. Then, AM is defined as:

$$\text{AM}(a) = Dis(\omega_a, \omega'_a).$$

For distributional distance Dis, we use Wasserstein-2 distance for numeric attributes and total variation distance for categorical attributes.

Instantiation Guideline Users can specify semantic or statistical attributes. A guiding question is: *"Which data properties matter for our downstream tasks?"* Common semantic attributes include topic, intent, and sentiment; statistical attributes include token length (overall or per node). Original categorical/numerical values are also often relevant. The selected attributes for our datasets are detailed in §B.4.

A.3 Downstream Evaluations (DE)

Label Generation Guideline To instantiate labels for downstream evaluation, we recommend that users first determine whether the label should be extracted directly from the sample or generated. Extracted labels typically include original categorical or numerical values—such as income level in our Adult dataset or paper decision in the ICLR dataset. If the label is to be generated, we suggest considering the question: *"What is a data property that is present and can be inferred from each node in a sample and is relevant to our downstream tasks?"* Generated labels are usually semantic attributes of the samples or nodes, such as topic, intent, or sentiment. To discourage prompt gaming by data uploaders, we provide a suggested prompt template for labeling: *"For each sample in ____ scenarios, label the ____ property. Possible answers include {____, ____, ____, ...}."* The specific labels used for our datasets are detailed in §B.4.

B Data Modeling and Evaluation Items of Each Dataset

B.1 Examples on ShareGPT

To support the running example of data modeling and representation discussed in §2, we first show a sample of the ShareGPT dataset, the data modeling of it, and the context-free grammar (CFG) according to its data structure.

A sample of ShareGPT

```
HUMAN: Pretend you are a successful and very thorough real estate
    investor and financial analyst
GPT: Sure, I can help you with that. As a successful real estate
    investor and financial analyst, I have a deep understanding of the
     real estate market and the financial factors that drive its
    performance. Here are some key things that I would keep in mind
    when analyzing investment opportunities:...
HUMAN: Assume the debt down payment is 50%
GPT: Sure, I can adjust the calculations to assume a 50% down payment.
     Here is how the projections would change:...
HUMAN: What is the IRR from the perspective of the equity
GPT: To calculate the IRR from the perspective of the equity, we need
    to adjust the cash flows to reflect the equity portion of the
    profits. Here is how we can calculate the equity IRR:...
```

Illustration of the data modeling of ShareGPT

We illustrate the data modeling of ShareGPT in Fig. 10.

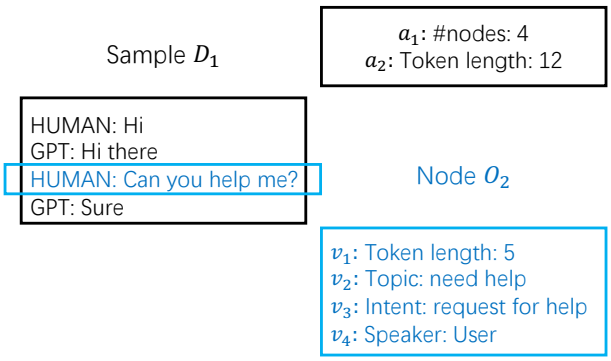


Figure 10: Illustration of the data modeling of ShareGPT.

CFG of ShareGPT

```
ShareGPT: conversation (conversation)*
    // ShareGPT contains one or more conversation rounds
conversation: query response
    // Each conversation round contains a query and a response
query: "HUMAN:␣" query_text
    // The query starts with "HUMAN:␣"
response: "GPT:␣" response_text
    // The response starts with "GPT:␣"
query_text: /(?s).+?(?=(?:GPT: |$))/
    // The query text ends before "GPT:␣" or the end of the string
response_text: /(?s).+?(?=(?:HUMAN: |$))/
    // The response text ends before "HUMAN:␣" or the end of the
        string
```

B.2 Dataset Descriptions

ShareGPT [1] The ShareGPT dataset contains multi-round conversations between users and GPT. We structure each conversation such that each user's query starts with 'HUMAN: ' and each GPT's response starts with 'GPT: '. The downstream task we conduct is to predict the user's intent and conversation topic based on user queries.

ICLR [2] The ICLR dataset contains the reviews, author rebuttals, follow-up discussions, and final decisions of the papers submitted to ICLR 2024 [2]. Each review or reviewer's comment starts with

'Reviewer n' where n represents the reviewer's identity, and each author rebuttal or discussion starts with 'Response'. The downstream task is to predict the research area of the paper based on the review and rebuttals.

Water [52] The Water dataset contains reviews of water bottles. The columns are product_name, overall_rating, title, cleaned_review and the goal is to predict the current rating (column "rating") of the bottle, which takes values $1, 2, 3, 4, 5$.

Arena [67] The Arena dataset contains pairs of human-model conversations. The columns are conversation_a, conversation_b and the goal is to predict which of the conversations are better (column "winner"), which takes values model_a, model_b, tie, "tie (bothbad)".

Adult [6] The Adult dataset contains census data. The columns are age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country and the goal is to predict income (column "income"), which takes values $<= 50k$ or $> 50k$.

Synthetic Datasets with Controllable Data Attributes We include two synthetic datasets[6] named **Synthetic Reviews** and the **Synthetic Grounding Dataset**. The reviews dataset has 4 fields, namely text, sentiment, emotion, and rating. The grounding dataset has 4 fields including two source documents, a query, and a response. We generate these datasets through a multi-step synthetic data generation process with GPT-4o wherein we verify whether the fields satisfy certain conditions, e.g., the reviews dataset is a 1:1 split of extreme negative and extreme positive reviews about products and the grounding dataset is a 1:1:1:1 split of relevant/irrelevant queries and consistent/inconsistent source documents. In particular, the reviews dataset is composed on only extreme reviews, either very positive or very negative. This differs from a typical review distribution and is unique to this particular dataset. Similarly, for the grounding dataset, we vary the samples along two axes, first on the consistency of the information between the sources and second, on the relevancy of the query to the sources. Each of the synthetic datasets is balanced in both their training and (downstream) test sets on these variations.

B.3 Data Modeling of Each Dataset

Tables 3 and 4 shows the data modeling and structure rules of each dataset.

Table 3: Data modeling of each dataset

Dataset	**Sample** D	**Sample Attributes**	**Node** O	**Node Attributes**
ShareGPT	a conversation	a_1: number of nodes a_2: token length	a query/response	v_1: token length v_2: topic v_3: intent v_4: speaker
ICLR	reviews & rebuttals of a paper	a_1: number of nodes a_2: token length a_3: topic a_4: final decision	a post from the reviewer/author	v_1: token length v_2: writer v_3: review score
Water	water bottle review	a_1: number of nodes a_2: attitude	a column of the tabular data	v_1: token length v_2: review score
Arena	2 conversations to compare	a_1: number of nodes a_2: winner	a column of the tabular data	v_1: token length v_2: winner
Adult	census information of an adult	a_1: age a_2: workclass	a column of the tabular data	v_1: token length v_2: income
Reviews	annotated product review	a_1: number of nodes a_2: rating	a review text	v_1: token length v_2: rating
Grounding	2 sources and a QA pair	a_1: number of nodes a_2: answer	a grounded response	v_1: token length v_2: answer

[6] https://www.kaggle.com/datasets/structpedataset/structpe-synthetic-datasets

Table 4: Structure rules of each dataset

Dataset	Rules
ShareGPT	**[Alternate Speakers]** $\forall O_i, O_{i+1} : O_i[\texttt{Speaker}] \neq O_{i+1}[\texttt{Speaker}]$. **[Format]** $O[\texttt{Speaker}] \in \{\texttt{User}, \texttt{AI Agent}\}$. If $O[\texttt{Speaker}] = \texttt{User}$, the text starts with 'HUMAN: '; If $O[\texttt{Speaker}] = \texttt{AI Agent}$, the text starts with 'GPT: '.
ICLR	**[Format]** $O[\texttt{Writer}] \in \{\texttt{Author}, \texttt{Reviewer 1-9}, \texttt{Meta Reviewer}\}$. If $O[\texttt{Writer}] = \texttt{Author}$, the text starts with 'Response:'; if $O[\texttt{Writer}] = \texttt{Reviewer n}$, the text starts with 'Reviewer n:' $(1 \leq n \leq 9)$. **[Format]** $O[\texttt{Review Score}] \in \{1, 3, 5, 6, 8, 10\}$. **[Format]** $D[\texttt{Final Decision}] \in \{\texttt{Reject}, \texttt{Accept:poster}, \texttt{Accept:top5\%}, \texttt{Accept:top25\%}, \}$.
Water	**[Format]**$O[\texttt{Overall\_rating}] \in \{\texttt{1.0}, \texttt{1.1}, \texttt{1.2}, \ldots, \texttt{4.9}, \texttt{5.0}\}$. **[Format]**$O[\texttt{Rating}] \in \{1, 2, 3, 4, 5\}$
Arena	**[Format]**$O[\texttt{Winner}] \in \{\texttt{model\_a}, \texttt{model\_b}, \texttt{tie}, \texttt{tie (bothbad)}\}$. O[Conversation_a] starts with "Question:" and has "Answer:" before somewhere in the following text. O[Conversation_b] starts with "Question:" and has "Answer:" before somewhere in the following text.
Adult	**[Format]**$O[\texttt{income}] \in \{<= 50k, > 50k\}$. **[Format]** Some of the columns are categorical (e.g. `workclass`, `native-country`). **[Format]** Some of the columns are numerical (e.g. `age`, `capital-gain`).
Reviews	**[Format]**$O[\texttt{Rating}] \in \{1, 2, 3, 4, 5\}$
Grounding	**[Format]**$O[\texttt{Consistency}] \in \{1, 2, 3, 4, 5\}$. **[Format]**$O[\texttt{relevancy}] \in \{1, 2, 3, 4, 5\}$

B.4 Evaluation Metrics of Each Dataset

The evaluation items of each dataset are summarized in Table 5.

For ShareGPT, in our experimental results, we show the semantic similarity of the node pair (query, response) as KND, show the distributional distance of the queries' token lengths as AM, and present the prediction accuracy of the conversation topics in downstream task performance.

For ICLR, we show the semantic similarity of the node pair (review, rebuttal) as KND, show the distributional distance of the reviews' token lengths as AM, and present the prediction accuracy of the paper's research area in downstream task performance.

C Additional Results on Struct-Bench

Resource Costs All baselines are implemented and performed on a server with eight H100 GPUs. Running experiments took approximately 400 GPU hours.

Implementation Details on Instruction Fine-tuning For both Instruct DP-FT and Instruct FT, we use the same instructions as those in the Random API of PE. We prepend the instructions to each training sample and fine-tune the foundation model for 20 epochs with batch size 32, weight decay 0.01, and learning rate 10^{-4}. The fine-tuned model then generates new samples conditioned on the given instructions.

C.1 Benchmarking DP Synthetic Data Generation Across Datasets

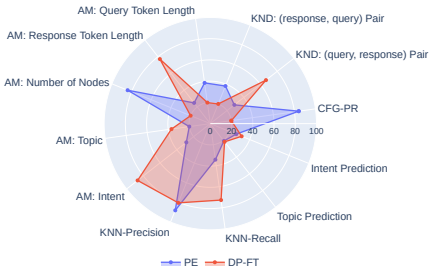
We present the results of benchmarking the DP synthetic data generation methods under different datasets with $\epsilon = 4$ in Table 6. We use GPT-2 for FT and DP-FT, and use GPT-4o for IF and PE.

C.2 Benchmarking DP Synthetic Data Generation with Varying Privacy Budget

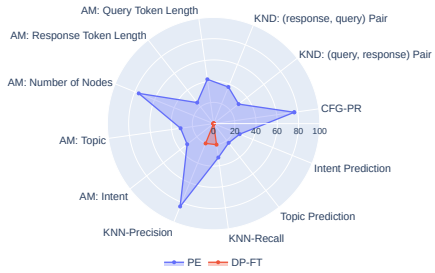
We illustrate the performance of PE and DP-FT on all metrics under different privacy budgets $\epsilon \in \{1, 2, 4, \infty\}$ on ShareGPT and ICLR datasets by radar plots in Figs. 11 and 12. Similar to §C.1, we use GPT-2 for FT and DP-FT, and use GPT-4o for IF and PE.

Table 5: Metrics for Different Datasets.

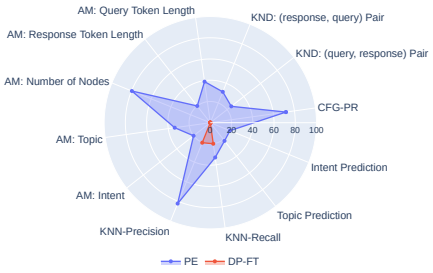
Dataset	Structural Metrics	Non-structural Metrics	Downstream Task
ShareGPT	**CFG-PR** **KND** 1. (query, response) pair 2. (response, query) pair **AM** 1. number of nodes 2.query token length 3. response token length 4. topic 5. intent	1. KNN-Precision 2. KNN-Recall	1. topic prediction 2. intent prediction
ICLR	**CFG-PR** **KND** 1. (review, rebuttal) pair 2. (rebuttal, comment) pair 3. (review, review) pair from different reviewers **AM** 1. number of nodes 2. review token length 3. rebuttal token length 4. Recommendation 5. final decision 6. topic	1. KNN-Precision 2. KNN-Recall	topic prediction
Arena	**CFG-PR** **KND** 1. (conversation_a, conversation_b) pair **AM** 1. winner	1. KNN-Precision 2. KNN-Recall	winner prediction
Water	**CFG-PR** **KND** 1. (title, cleaned_review) pair **AM** 1. attitude	1. KNN-Precision 2. KNN-Recall	rating prediction
Adult	**CFG-PR** **KND** 1. (native country, workclass) pair **AM** 1. income	1. KNN-Precision 2. KNN-Recall	income prediction
Reviews	**CFG-PR** **KND** 1. (text, sentiment) pair **AM** 1. review token length	1. KNN-Precision 2. KNN-Recall	review label prediction
Grounding	**CFG-PR** **KND** 1. (source1, source2) pair **AM** 1. query relevancy	1. KNN-Precision 2. KNN-Recall	query relevancy prediction



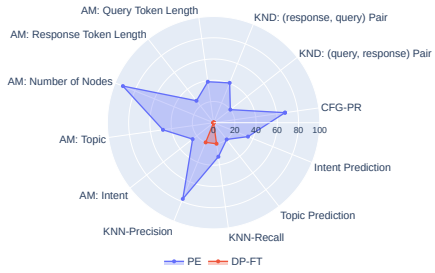
(a) $\epsilon = \infty$



(b) $\epsilon = 4$

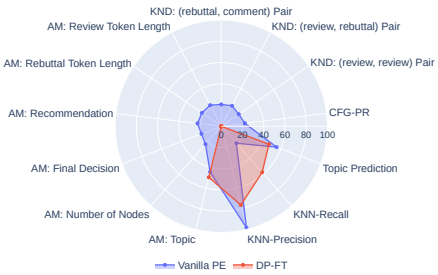


(c) $\epsilon = 2$

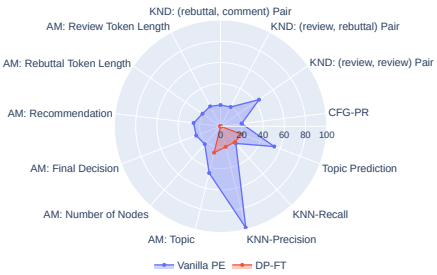


(d) $\epsilon = 1$

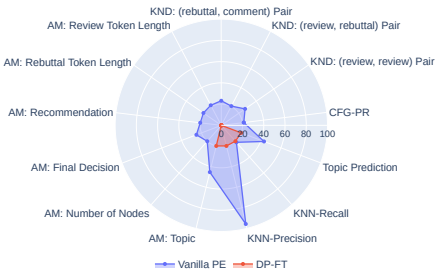
Figure 11: Performance of PE and DP-FT on all metrics under different privacy budgets on ShareGPT.



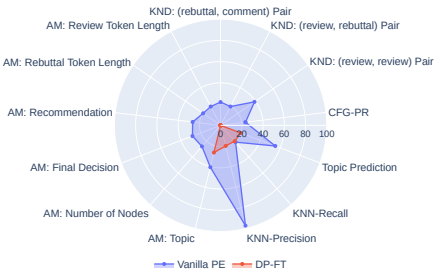
(a) $\epsilon = \infty$



(b) $\epsilon = 4$



(c) $\epsilon = 2$



(d) $\epsilon = 1$

Figure 12: Performance of PE and DP-FT on all metrics under different privacy budgets on ICLR.

Table 6: DP synthetic data generation benchmarking results on Struct-Bench with $\epsilon = 4$

		Structural Metrics			Non-Structural Metrics		DE
Dataset	Baseline	CFG-PR ↑	KND ↓	AM ↓	KNN-Precision ↑	KNN-Recall ↑	Acc↑
ShareGPT	IF ($\epsilon = 0$)	0.8700	0.0635	43.8514	0.7217	0.2627	0.3754
	FT ($\epsilon = \infty$)	0.5378	0.0315	52.6984	0.7594	0.6588	0.3718
	DP-FT	0	-	-	0.0161	0.0000	-
	PE	0.8633	0.0660	38.1678	0.8050	0.1528	0.3816
ICLR	IF ($\epsilon = 0$)	0.1733	0.2582	204.7997	0.8400	0.0257	0.4715
	FT ($\epsilon = \infty$)	0	-	-	0.7056	0.4747	0.4584
	DP-FT	0	-	-	0.0000	0.0000	0.1806
	PE	0.1900	0.2599	240.9434	0.9800	0.0207	0.5218
Water	IF ($\epsilon = 0$)	1.0000	0.4222	0.1574	0.0000	0.0060	0.5485
	FT ($\epsilon = \infty$)	0	-	-	0.0000	0.0060	-
	DP-FT	0	-	-	0.0000	0.0060	-
	PE	1.0000	0.2877	0.0236	0.0000	0.0070	0.6130
Arena	IF ($\epsilon = 0$)	1.0000	0.1257	0.9395	0.0000	0.0090	0.3607
	FT ($\epsilon = \infty$)	0	-	-	0.0000	0.0060	-
	DP-FT	0	-	-	0.0000	0.0060	-
	PE	1.0000	0.1054	0.9193	0.0000	0.0070	0.3510
Adult	IF ($\epsilon = 0$)	1.0000	0.0290	0.0332	0.0030	0.0030	0.7920
	FT ($\epsilon = \infty$)	0	-	-	0.0030	0.0030	-
	DP-FT	0	-	-	0.0030	0.0030	-
	PE	1.0000	0.0042	0.0000	0.0030	0.0060	0.8017
Reviews	IF ($\epsilon = 0$)	1.0000	0.3510	0.4010	0.0334	0.0344	0.6000
	FT ($\epsilon = \infty$)	0	-	-	0.0020	0.0900	0.5400
	DP-FT	0	0.0020	0.0060	0.0020	0.0900	0.5600
	PE	1.0000	0.2495	0.0770	0.0290	0.0900	0.5400
Grounding	IF ($\epsilon = 0$)	1.0000	0.5800	0.6006	0.0500	0.0600	0.6400
	FT ($\epsilon = \infty$)	0	-	-	0.0290	0.0900	0.4000
	DP-FT	0	-	-	0.0430	0.0900	0.4000
	PE	1.0000	0.1435	0.4710	0.0300	0.0600	0.6000

C.3 Benchmarking DP Synthetic Data Generation on ShareGPT using Llama2-7b

We illustrate the performance of PE, DP-FT, and Instruct DP-FT in Fig. 13 , where each dimension corresponds to a different metric from Struct-Bench. To better visualize differences in the performance of different methods, we scale the metrics in these radar plots as follows: We assign a score of 0 if CFG-PR=0 or a structure-related metric is not applicable for the dataset, and rescale the values of other metrics from 20 to 100, where 20 indicates the worst performance among all methods, and 100 indicates the performance upper bound the synthetic data can achieve (e.g., CFG-PR=1 or AM=0).

Fig. 13 shows that (1) DP-FT does not learn any structural information (CFG-PR); and (2) with instruction-guided conditional generation, Instruct DP-FT achieves similar performance to PE on most metrics and has a slight edge in terms of structure learning CFG-PR.

D Detailed analysis of the case study on PE

Resource Costs All baselines are implemented and performed on a server with eight H100 GPUs. Running experiments took approximately 1000 GPU hours.

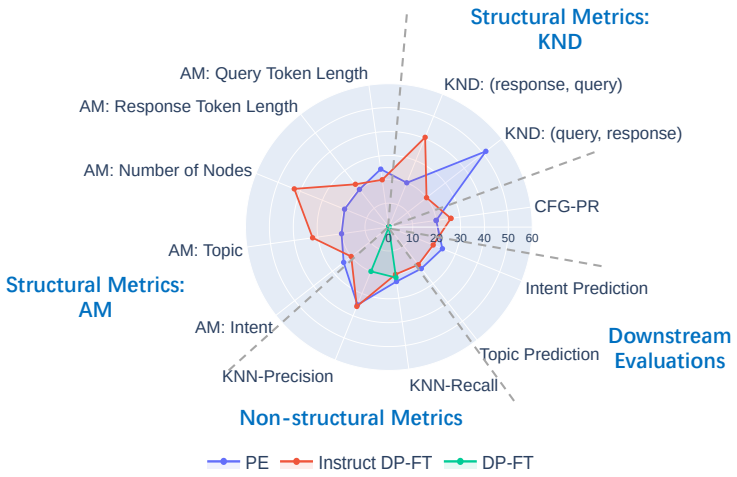


Figure 13: Performance of different baselines on ShareGPT using Llama2-7b with $\epsilon = 4$. With instruction-guided conditional generation, Instruct DP-FT achieves similar performance to PE on most metrics and has a slight edge in terms of CFG-PR.

D.1 Analyzing Vanilla PE on ShareGPT Dataset

In this section, we analyze the performance of PE under the ShareGPT dataset according to our proposed benchmark. We further divide the metrics into semantic and statistic metrics, and the evaluation items for ShareGPT can be categorized in Table 7.

Table 7: Metrics for ShareGPT.

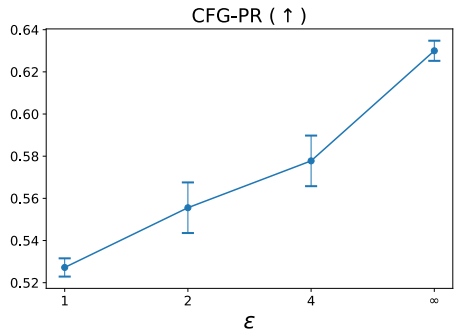
	Statistic Metrics	**Semantic Metrics**	**CFG-PR**
Structural Metrics	AM: 1. number of statements 2. query token length 3. response token length	KND: 1. (query, response) pair 2. (response, query) pair AM: 1. topic 2. intent	CFG-PR
Non-structural Metrics	-	1. KNN-Precision 2. KNN-Recall	-
Downstream Tasks	-	1. topic prediction 2. intent prediction	-

We illustrate and compare the performance of PE with privacy parameter $\epsilon \in \{1, 2, 4, \infty\}$ under structural semantic and statistic metrics in Figs. 14c and 14d respectively, and plot the CFG-PR and KNN-Precision & KNN-Recall in Figs. 14a and 14b. We do not include PE with $\epsilon = 0$ (that is, IF) as its CFG-PR is only 2% and thus its performance under structural metrics is unreliable.

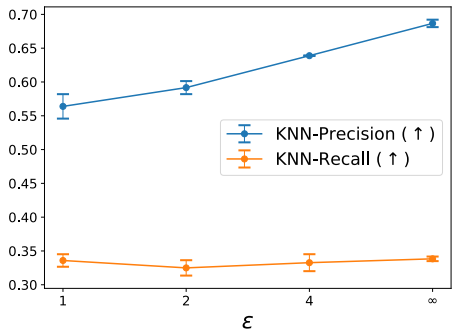
As we can observe, only CFG-PR and KNN-Precision improve with the increase of ϵ, while the value of KNN-Recall always keep around 0.35 and the performance under other semantic metrics and all statistic metrics does not necessarily increase with more relaxed privacy constraints. Additionally, CFG-PR drops below 60% when $\epsilon \leq 4$. Since downstream tasks also depend on structural information, we can conclude that PE mainly focuses on non-structural semantic quality of the synthetic samples, while suffers from poor performance on semantic diversity and structure-based properties.

D.2 CFG Reformat Prompt

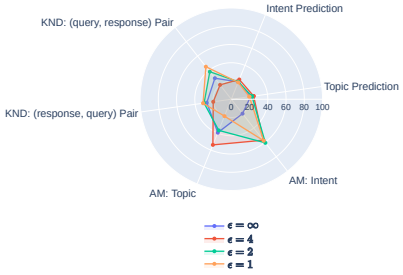
```
You are required to REFORMAT the provided conversation between a user
    and an AI agent in ChatGPT.  The format should be:
    -User prompt must start with "HUMAN:␣", and ChatGPT response must
        start with "GPT:␣".
```



(a) CFG-PR



(b) KNN-Precision & KNN-Recall



(c) Performance on structural semantic metrics

(d) Performance on statistic metrics

Figure 14: Performance of Vanilla PE with different privacy guarantees under ShareGPT dataset

```
    -The conversation may contain one or multiple rounds. Each round
        includes ONE user prompt and ONE ChatGPT response.
    -User prompts and ChatGPT responses appear alternately.
    -The conversation begins with a user prompts.
The reformatted conversation follows the following context-free
    grammar:
    sharegpt: round (round)*
    round: request response
    request: "HUMAN:␣" user_string
    response: "GPT:␣" gpt_string
    user_string: /(?s).+?(?=(?:GPT: |HUMAN: |$))/
    gpt_string: /(?s).+?(?=(?:GPT: |HUMAN: |$))/
    %import common.WS
    %ignore WS
Do NOT change the content of the conversation.
For example: If the input conversation is: "How␣are␣you?␣I'm␣fine."
    You should reformat it as "HUMAN:␣How␣are␣you?␣GPT:␣I'm␣fine."
```

D.3 Further Analysis on CFG Reformat as Self-debugging

We compare the performance of vanilla PE and PE with CFG reformat on all evaluation items in Fig. 15. Self-debugging after voting directly reformats voted samples, which are taken as output or utilized as seeds in the next PE iteration without further selection, resulting in higher CFG-PR while lower performance on semantic and statistic properties.

D.4 Improving Node Dependency (KND): Fix Format Token in Variation API

Key node dependency (i.e., KND) is an important semantic metric that measures the similarity of the node pair dependencies between the private and synthetic datasets. To improve KND, we fix the format tokens during blank-filling in variation API. Since nodes are recognized and separated by

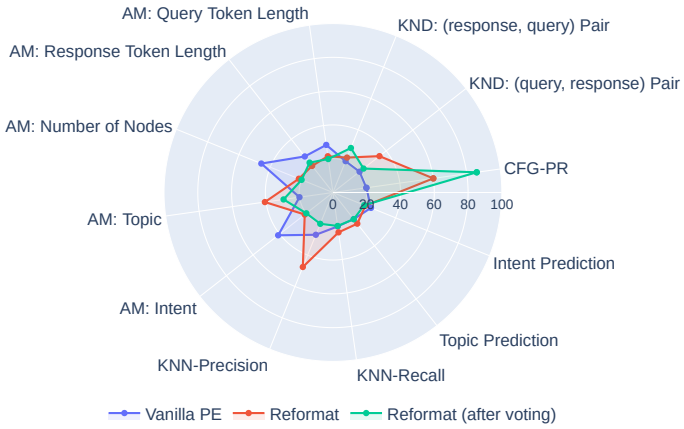


Figure 15: Performance of PE with CFG Reformat on ShareGPT with $\epsilon = 4$

format tokens in textual datasets, fixing the format tokens ensures that multiple nodes will not be mistakenly merged into one and thus helps to remain the original semantic meaning of each node, and therefore the node semantic dependencies. We compare the performance of vanilla PE and PE with fixed format token on KND on (query, response) and (response, query) pairs and CFG-PR in Fig. 16, where we consider two variants of our method: fix all the format tokens (shown as Fixed Token) and randomly fix 65% of the format tokens (shown as Fixed Selected Token). We can observe that our methods achieve better semantic performance on KND compared to vanilla PE, and Fixed Token outperforms since it keeps more node structures than Fixed Selected Token. Additionally, as fixing format tokens avoids node merging, it also improves the structural validity, i.e., CFG-PR.

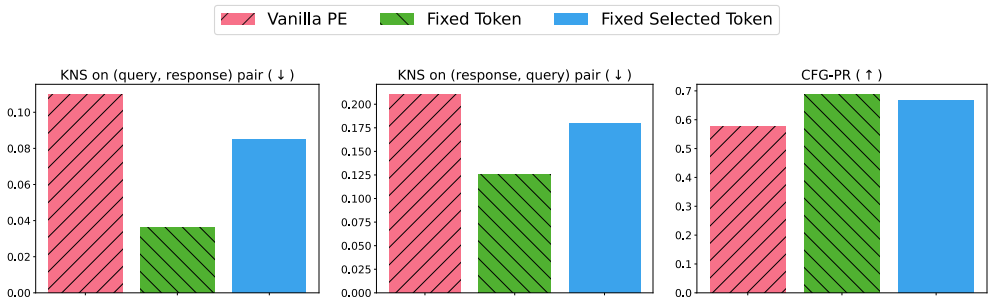


Figure 16: Performance of Vanilla PE and PE with fix token on CFG-PR and KND

We then compare the performance of vanilla PE and our methods on all metrics in Fig. 17. Since Fixed Token fixes all format tokens, the blank-filling process becomes less flexible, e.g., the number of nodes after blank-filling will never decrease, which is ensured by existing format tokens. Therefore, its performance in most statistic properties is worse than that of vanilla PE and Fixed Selected Token.

D.5 Further Analysis on Node extraction & Auto-generation

To further examine the semantic diversity of the dataset, we adopt another metric Type to Token Ratio (TTR) [41] to provide auxiliary information. TTR measures diversity in the tokens used in the dataset by dividing the number of unique tokens by the total number of tokens in the dataset. A higher TTR suggests a more diverse vocabulary. Fig. 18 shows that Extract Query has a higher TTR than vanilla PE.

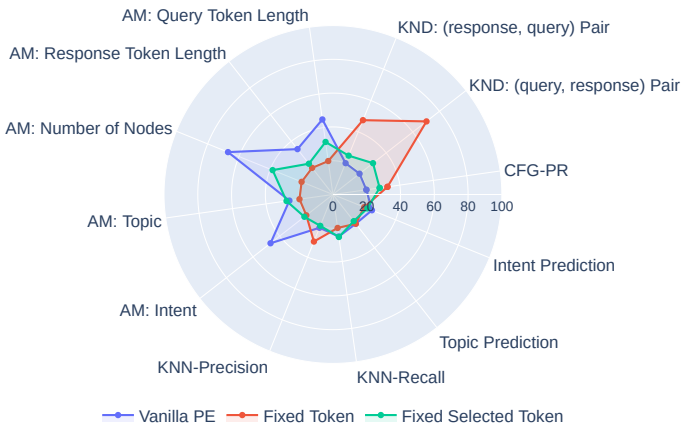


Figure 17: Performance of PE with Fix Format Token on ShareGPT with $\epsilon = 4$

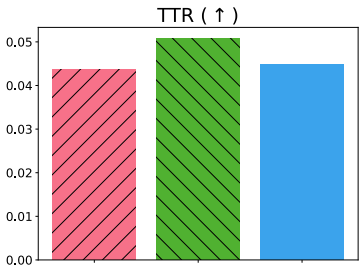
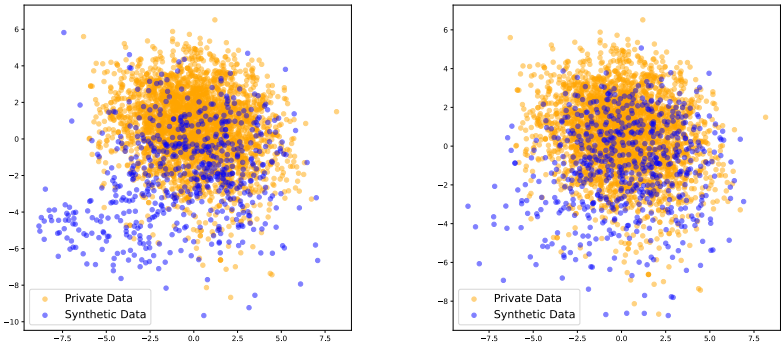


Figure 18: Performance of vanilla PE and PE with node extraction on Type to Token Ratio (TTR).

To illustrate the semantic quality and diversity of the synthetic dataset, we then focus on the embeddings of the generated sample, and draw them in a 2-dimensional plot after principal component analysis (PCA). As shown in Figs. 19a and 19b, the embeddings of vanilla PE and PE with query node extraction (blue dots) are drawn together with the embeddings of private data (yellow dots). We can easily observe that the embeddings of PE with query node extraction have more overlaps with the private data embeddings, indicating a higher sample semantic quality and diversity.



(a) Embeddings of Vanilla PE (b) Embeddings of PE with node extraction

Figure 19: Embedding distributions of Vanilla PE and PE with node extraction.

We them compare the performance of vanilla PE and PE with node extraction on all metrics in Fig. 20, where we consider several variants of our method: extract all query nodes and auto-generate all response nodes (shown as Extract Query); combination of query node extraction, reformat before voting, and fix format token (Extract Query & Reformat & Fixed Token); combination of query node extraction, reformat before voting, and fix 65% format token (Extract Query & Reformat & Fixed Selected Token); combination of response node extraction, reformat before voting, and fix 65% format token (Extract Response & Reformat & Fixed Selected Token). We can observe that (1) Extract Query outperforms vanilla PE across most statistic properties, CFG-PR, and semantic properties including KNN-Precision, KNN-Recall, and KND on (response, query) pair. (2) Extract Query & Reformat & Fixed Selected Token outperforms or achieves similar performance to other node extraction variants on CFG-PR, most statistic and semantic metrics. This indicates that the combination of reformat and fix selected format tokens to node extraction improves CFG-PR and structural semantic performance without degrading statistic performance. (3) Extracting query nodes outperforms extracting response nodes, indicating that the type of nodes extracted significantly influences the synthetic data performance.

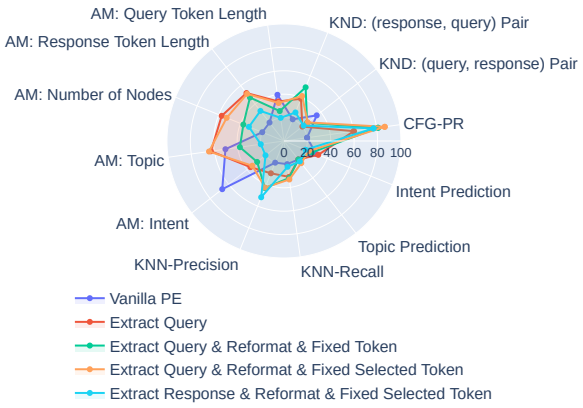


Figure 20: Performance of PE with Node Extraction on ShareGPT with $\epsilon = 4$

D.6 Performance Comparison between Different Methods

We compare the performance of our proposed methods and some combinations of them according to our benchmark. Specifically, in Fig. 21, we illustrate the performance of vanilla PE; PE with CFG reformat; PE with fixed format token; combination of CFG reformat and fix format token (Fixed Token & Reformat); and combination of CFG reformat, fix partial format token, and query node extraction (Extract Query & Reformat & Fixed Selected Token). As we can observe, Extract Query & Reformat & Fixed Selected Token outperforms on structural validity CFG-PR, semantic properties KNN-Precision and KNN-Recall, and statistic properties AM on conversation round and response token length; while Fixed Token & Reformat outperforms mainly on semantic properties KND on (query, response) and (response, query) pair. As different methods focus on different aspects of the synthetic data, users can choose the method according to their practical needs. The algorithm design and analysis based on our benchmark also pave the way to propose a method that outperforms on all evaluation metrics, which we leave as a future work.

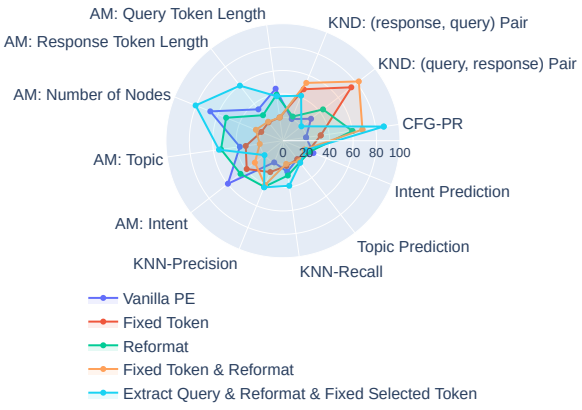


Figure 21: Performance of Different Methods on ShareGPT with $\epsilon = 4$