Privacy-Preserving Retrieval-Augmented Generation with Differential Privacy

***Abstract*—With the recent remarkable advancement of large language models (LLMs), there has been a growing interest in utilizing them in the domains with highly sensitive data that lies outside their training data. For this purpose, retrieval-augmented generation (RAG) is particularly effective—it assists LLMs by directly providing relevant information from the external knowledge sources. However, without extra privacy safeguards, RAG outputs risk leaking sensitive information from the external data source. In this work, we explore RAG under differential privacy (DP), a formal guarantee of data privacy. The main challenge with differentially private RAG is how to generate long accurate answers within a moderate privacy budget. We address this by proposing an algorithm that smartly spends privacy budget only for the tokens that require the sensitive information and uses the non-private LLM for other tokens. Our extensive empirical evaluations reveal that our algorithm outperforms the non-RAG baseline under a reasonable privacy budget of $\epsilon \approx 10$ across different models and datasets.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

Large language models (LLMs) have shown a great deal of promise in a variety of applications. In particular, a major application of LLMs is in question-answering (QA). The practical adoption of these systems often involves domains whose data is highly sensitive. For instance, healthcare institutions might want to utilize their internal medical records to provide precise medical information and personal feedback, while legal firms can leverage their case archives to assist clients with legal research and documentation. One way to achieve such domain-specific QA is through retrieval-augmented generation (RAG) [6, 18, 25]. Here, we have a set of domain-specific documents; while answering a question, RAG retrieves a list of relevant documents and inputs them to LLMs as the context. However, even though this is effective for QA, RAG on a sensitive corpus can leak private information about individual documents in the corpus [5, 34, 35, 45]. This is particularly problematic when end users are outside the data-holding entity, e.g., patients interacting with a healthcare institution's RAG system.

Our goal in this paper is to prevent the information leakage of the sensitive external corpus by designing a privacy-preserving RAG system. For this purpose, we use differential privacy (DP) [12, 13] as a notion of privacy. Differential privacy guarantees privacy by ensuring that the participation of a single person's data does not make much difference to the probability of any output. In our system, we assume that each RAG document comes from a single individual, and our goal is to ensure differential privacy on the eventual answer of the LLM.

There are two aspects of the challenges with designing an effective RAG algorithm under DP. The first is how to fit differential privacy into the RAG framework, and the second is how to manage the privacy-utility tradeoffs. We address the first challenge by proposing an algorithm, DPVoteRAG, based on the sample-and-aggregate framework in DP [30]. Our algorithm prepares multiple LLM instances, or voters, feeds disjoint partitions of the sensitive corpus to them, and produces output tokens one by one each through the majority vote of the voters' token outputs. Note, however, that LLMs often output many tokens in response to a question. This is detrimental to privacy—the composition property of differential privacy states that multiple calculations based on the same dataset lead to greater privacy degradation. To resolve this challenge, we design another algorithm, DPSparseVoteRAG, that spends a privacy budget only when we need to. More specifically, we take advantage of the fact in RAG that LLMs require the sensitive corpus *only* when generating tokens related to the knowledge. When not, outputs from LLMs without any context suffice. We formalize this idea with the sparse vector technique in DP [11, 14]—when voters agree with the non-private output of the LLM without contexts, we will simply output the non-private one without incurring a privacy budget. Consequently, our algorithm successfully generates sufficiently long, accurate responses under a reasonable privacy budget.

We conduct extensive experiments with a series of LLMs on multiple benchmarking datasets to evaluate our algorithms. The results demonstrate that our algorithms are able to enhance the LLMs by RAG while ensuring privacy for the external corpus. We further show that DPSparseVoteRAG improves DPVoteRAG by only spending a privacy budget when necessary and enabling us to generate longer answers within a reasonable privacy budget of $\epsilon \approx 10$.

II. Preliminaries & Problem Setting

A. Retrieval-Augmented Generation with Large Language Model

Retrieval-augmented generation (RAG) is a technique to improve the performance of large language models (LLMs) on knowledge-intensive tasks by providing external knowledge. Given a question prompt, a retriever finds relevant documents from the external data source. Then, the relevant documents are added to the prompt as the contexts. An LLM (or generator) takes the augmented prompt as input and outputs the answer.

More formally, let $x \in \bigcup_{t=1}^{\infty} \mathcal{V}^t$ be a prompt, where $\mathcal{V}$ is some vocabulary. We further let D be a dataset of documents as an external corpus with size $|D| = n$. A retriever R finds

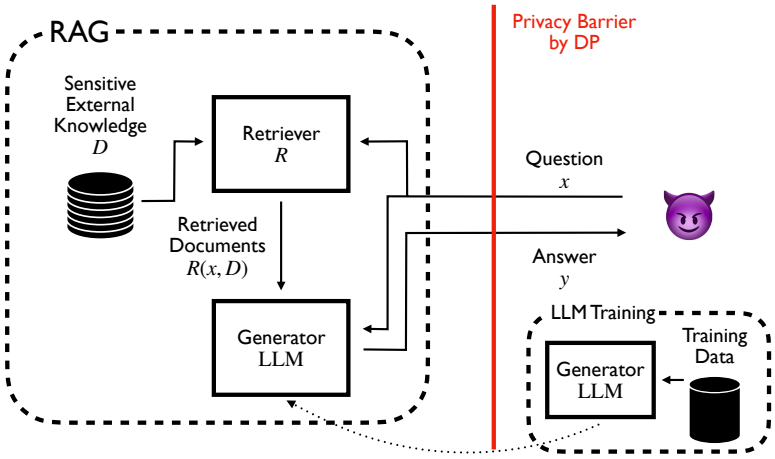

Fig. 1: Overview of our problem setting. Note that the LLM in RAG is trained outside the privacy barrier by DP.

a subset of D, $D_x \subset D$, with size k that is relevant to x, i.e., $D_x = R(x, D; k)$. Finally, an LLM generates an answer $y = \mathrm{LLM}(x, D_x) \in \bigcup_{t=1}^{\infty} \mathcal{V}^t$. The answer generation can further be decomposed into next-token generation. In particular, for each t, the t-th token $y_t \in \mathcal{V}$ is generated by LLM_t, which takes x, D_x, and previously generated tokens $y_{<t}$ as inputs: $y_t = \mathrm{LLM}_t(x, D_x, y_{<t})$.

B. Differential Privacy

Differential privacy (DP) is a strong cryptographically motivated definition of individual-level privacy. It guarantees that the participation of a single individual in a dataset does not change the probability of any outcome by much. In particular, suppose we have two datasets D and D', each consisting of private data from n individuals. We say that D and D' are neighboring if they differ in a single individual's private data. A randomized algorithm satisfies differential privacy if the output distributions on any pair of neighboring datasets are close. The formal definition is given as follows.

Definition 1 ((ϵ, δ)-Differential Privacy [12])**.** *A randomized algorithm M satisfies (ϵ, δ)-differential privacy if for any two neighboring datasets D, D' and for any $S \subseteq \mathrm{range}(M)$,*

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.$$

One of the key properties of DP is composition—sequential runs of differentially private algorithms also satisfy differential privacy. The composition property quantitatively captures the intuition that the more we release the information about the sensitive data, the worse the privacy guarantee becomes. More specifically, suppose $M_1, \ldots, M_T$ are (ϵ_0, δ_0)-differentially private algorithms, which can be chosen adaptively based on previous outputs. Sequential composition theorem [12] states that the composed sequence of such algorithms guarantee $(T\epsilon_0, T\delta_0)$-differential privacy. Furthermore, advanced composition theorem [15, 21] states that the total privacy guarantee has $\epsilon = \mathcal{O}(\sqrt{T}\epsilon_0)$.

1) Sparse Vector Technique: The sparse vector technique [11, 14] has originally emerged as the alternative of the composition in DP when we have such a large number of numerical queries that the composition theorem cannot provide a reasonable privacy guarantee but we are only interested in answers above some threshold. In such a case, the sparse vector technique algorithm, Sparse, reports whether each (noisy) query answer exceeds the threshold. It is shown that the privacy guarantee degrades by the number of queries above the threshold, instead of the total number of queries. Therefore, we save privacy budget by much when we expect only a few queries will be above the threshold.

We state the AboveThreshold in Algorithm 1 and state their guarantee as below.

Algorithm 1: AboveThreshold [11]

Require: A private database D, an adaptively chosen stream of sensitivity 1 queries $f_1, \cdots$, and a threshold τ.
Ensure: A stream of responses $a_1, \cdots$.

```
Let τ̂ = τ + Lap(2/ε).
for Each query i do
    Let v_i = Lap(4/ε)
    if f_i(D) + v_i ≥ τ̂ then
        Output a_i = 1, Halt.
    else
        Output a_i = 0.
    end if
end for
```

Theorem 1. *ALgorithm 1 is $(\varepsilon, 0)$-DP.*

2) Differentially Private Generation via Sample-and-Aggregate: There has been a body of work on generating a token sequence by LLM with DP. The most common way is to borrow the idea of the sample-and-aggregate framework in DP [30]. To generate a single token, a set of LLMs, each depending on a disjoint subset of the sensitive dataset D, generates a token respectively. The generated tokens form

an aggregate histogram of tokens, which is then carefully randomized with noise and only the most frequent token in the noisy histogram is published as the final output. The repetition of this process along with the composition theorem of DP yields the differentially private token sequence generation.

C. Problem Setting

Our goal is to generate an LLM answer to a prompt x with retrieved external knowledge, $D_x = R(x, D; k)$, from a *sensitive* data source D with a differential privacy guarantee. More specifically, let the sensitive data source D be a collection of individuals' records—one record corresponds to one individual's sensitive data[1]. We consider a realistic adversary who does not have direct access to the data source D but has a capability of querying to the RAG system with any prompt x. We further assume that the LLM used in the RAG system is a copy of publicly available LLMs and is already pre-trained (and fine-tuned) with data *disjoint* from the sensitive data source D. That is, having access to the LLM parameters and/or pre-training (and fine-tuning) data does not provide any information on the sensitive data source D. To this end, we aim to formally guarantee that given any question x, a randomized LLM generation algorithm with RAG, $\mathrm{LLM}_{\mathrm{priv}}(x, R(x, D; k))$ satisfies (ϵ, δ)-differential privacy w.r.t the external knowledge data source D. We present the figure for this problem setting in Figure 1.

III. Differentially Private Retrieval-Augmented Generation with Sparse Vector Technique

Our differentially private RAG algorithm consists of two main components—DP voting for the single-token generation and efficient privacy budget spending by leveraging the sparse vector technique combined with the utilization of LLMs without any relevant documents provided. These two components enable us to generate answers that incorporate external knowledge while guaranteeing a reasonable level of differential privacy. We start from our algorithm with the first component alone, and then extend it to include the second component. The graphical overview of our algorithm is presented in Figure 2.

A. DPVoteRAG: Differentially Private Voting Algorithm for RAG

By the nature of retrieval in RAG—retrieving relevant documents for a question, the LLM outputs can depend on a sensitive individual's document. Therefore, our algorithmic design needs to relax the dependency of a single individual's document on the output, while exploiting the external data source, to achieve a reasonable privacy-utility tradeoff. Inspired by the differentially private generation via sample-and-aggregate framework, we present a differentially private voting algorithm for RAG—**DPVoteRAG**.

Given a prompt x and external data source D, DPVoteRAG first retrieves mk documents as D_x. Then it makes uniformly randomly partitions D_x into m disjoint datasets $D_x^1, \ldots, D_x^m$ and each subset has exact size k. Then, for each $i = 1, \ldots, m$, it feeds k documents D_x^i into the LLM along with the original prompt x, and generates a next token. It collects these tokens to form a histogram over the vocabulary. It remains to privately choose the most frequent element from the histogram. While it is generally hard to do so when the histogram dimension is large as in our setting, e.g., the vocabulary size of OPT [46] is 50272, there is a line of work in the community to overcome this difficulty. Following the work by Hong et al. [17], we integrate the LimitedDomain mechanism [10] into our algorithm. The mechanism enables us to reduce the histogram dimension significantly with some cost in a privacy budget and thus achieve a better privacy-utility tradeoff. By its design, the LimitedDomain mechanism possibly outputs the designed null token. In such a case, we halt the algorithm, or equivalently, regard that it outputs the end of sequence token. [2] Finally, we append the chosen token to the next input to the LLM. We repeat this process until we see the end of sequence token chosen or reach the maximum number of token generation, which is computed in advance from the per-token and total privacy budget [3]. We present the concrete algorithm in Algorithm 2. The formal privacy analysis is as follows.

Theorem 2. *For any question x, DPVoteRAG satisfies $(\epsilon_{\mathrm{total}}, \delta_{\mathrm{total}})$-DP w.r.t. the external data source D.*

The guarantee simply follows from the property of uniformly random partition, the privacy guarantee of the LimitedDomain mechanism and the composition theorem used to compute $T_{\max}$.

Proof. Let's first consider steps 3 and 4 in Algorithm 2. Suppose L_x is the list of documents in D_x ranked by the relevance. One way to uniformly randomly split D_x into m disjoint subsets $D_x^1, \cdots, D_x^m$ is that: given a ranked list of documents $L_x = (d_1, \cdots, d_{mk})$, we randomly permute this list by π to $L_x^\pi = (d_{\pi(1)}, \cdots, d_{\pi(mk)})$ and let $D_x^i := \{d_{\pi((i-1)k+1)}, \cdots, d_{\pi(ik)}\}$. The process from D_x to L_x is deterministic, and the remaining of the algorithm is independent of D given L_x^π. Therefore, we can equivalently denote the outcome of Algorithm 2 as $\mathcal{A}(L_x^\pi)$.

For any two neighboring datasets D and D', the retrieved datasets are $D_x = R(x, D; mk)$ and $D'_x = R(x, D'; mk)$ and we denote $L_x = (d_1, \cdots, d_{mk})$ and $L'_x = (d'_1, \cdots, d'_{mk})$. We only need to show for any set of outcomes S, $\Pr_{A,\pi}[A(L_x^\pi) \in S] \leq \exp(\epsilon) \Pr_{A,\pi}[A((L'_x)^\pi) \in S] + \delta$. First of all, D_x and D'_x have at most one different document (without considering the order). Therefore we can define another list of documents $L''_x = (d''_1, \cdots, d''_{mk})$, such that L''_x is some ranking of D'_x and

[1] It is straightforward to extend the setting to where multiple records correspond to one individual's data by modifying the granularity of neighboring datasets in DP possibly with overhead in privacy-utility tradeoff.

[2] We find that by choosing the appropriate size of reduced dimension, the LimitedDomain mechanism in our experiment rarely outputs the null token.

[3] The maximum number of token generation is computed as follows. We first calculate the maximum numbers of composition with the sequential and advanced composition theorem [11] under the per-token privacy budget $(\epsilon_{\mathrm{token}}, \delta_{\mathrm{token}})$ and total privacy budget $(\epsilon_{\mathrm{total}}, \delta_{\mathrm{total}})$. Then, we take the maximum of two numbers of possible composition.

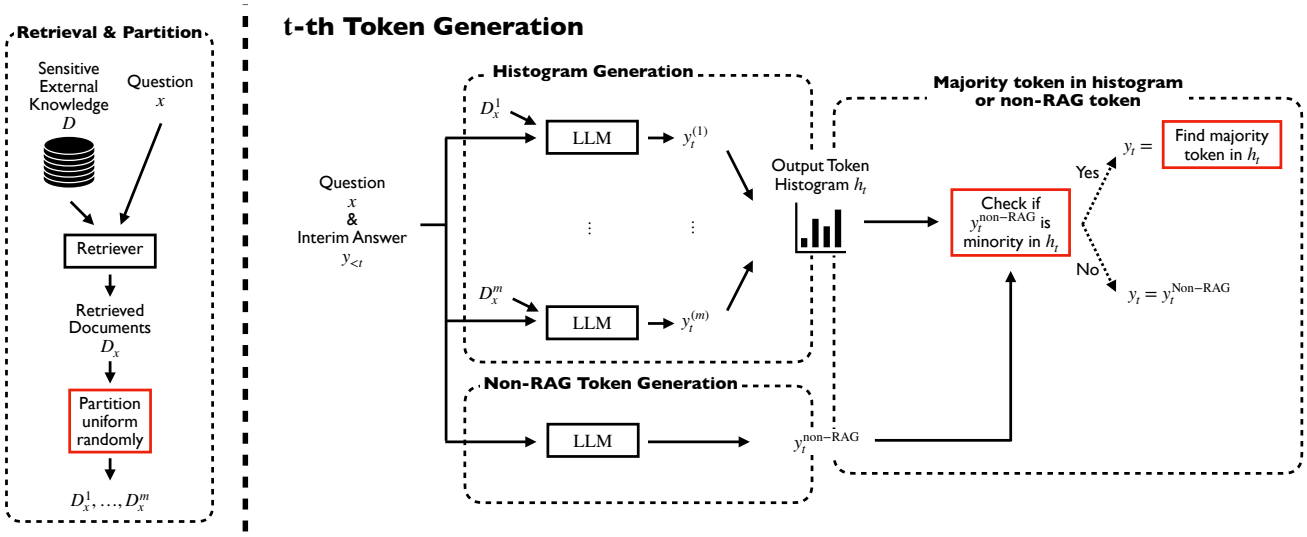

Fig. 2: Overview of DPSparseVoteRAG. It first retrieves the relevant documents to a question x from sensitive external knowledge D and partitions them uniform randomly (left). When generating t-th token (right), it takes a question x and an interim answer up to $t-1$-th tokens, $y_{<t}$, as inputs and outputs a new token y_t. Components enclosed in red indicate that operations involve randomness for the privacy guarantee. DPVoteRAG works the same until computing the output token histogram h_t, but it immediately finds the majority token in h_t afterwards.

it differs at most one position from L. Notice that L'_x and L''_x are only different at orders. Therefore $(L'_x)^\pi$ and $(L''_x)^\pi$ have same distributions and as a consequence $A((L'_x)^\pi)$ and $A((L''_x)^\pi)$ have same distributions.

Thus, the remaining is to prove for any set of outcomes S, $\Pr_{A,\pi}[A(L_x^\pi) \in S] \leq \exp(\epsilon) \Pr_{A,\pi}[A((L''_x)^\pi) \in S] + \delta$. We can actually prove a stronger conclusion $\Pr_A[A(L_x^\pi) \in S] \leq \exp(\epsilon) \Pr_A[A((L''_x)^\pi) \in S] + \delta$. It is because L_x^π and $(L''_x)^\pi$ differ at most one position and therefore at most one subset in step 4 is different given L_x^π or $(L''_x)^\pi$. This means that the histogram in step 10 differs at most one token. The guarantees of LimitedDomain and the composition theorem of DP together imply $\Pr_A[A(L_x^\pi) \in S] \leq \exp(\epsilon) \Pr_A[A((L''_x)^\pi) \in S] + \delta$. □

B. DPSparseVoteRAG: Differentially Private Voting Algorithm for RAG with Sparse Vector Technique

The main drawback of the aforementioned algorithm is that we need to spend a non-negligible amount of privacy budget for each token to guarantee its quality. This prevents our algorithm from generating longer answers—sometimes it can halt before it generates the actual answers due to privacy budget shortage. More concretely, consider the following question-answering example:

> **Question**: what type of literature is the great gatsby
> **Ground Truth Answer**: novel

Here are possible outputs from (non-private) RAG and our DPVoteRAG given the retrieved documents.

> **RAG Output:** The Great Gatsby is a novel written by American author F. Scott Fitzgerald.
> **DPVoteRAG Output:** The Great Gatsby is a

While non-private RAG correctly answers the question, due to the pre-fixed total privacy budget, DPVoteRAG can only output 5 words and thus it fails to output the ground truth answer, *novel*.

However, having a closer look at our voting algorithm, we observe that there is room for improvement. When generating the 3rd word, *Gatsby*, every input of the LLM contains *The Great*, 1st and 2nd previously output words, and *the great gatsby*, a part of the question, even though the provided retrieved documents are different. Thus, the LLM should successfully generate *Gatsby* without access to the sensitive information. Ideally, we should not spend a privacy budget for such a word.

We address this by incorporating the sparse vector technique into our voting algorithm, yielding our improved algorithm **DPSparseVoteRAG**. In particular, before we apply private voting among generated tokens, we check if the generated tokens coincide with the token generated by the LLM without retrieved documents appended, i.e., the input is composed of the prompt and previously generated tokens only. We continue to the voting only when they do not coincide. Otherwise, we use the LLM output without retrieved documents. It is shown from the analysis of the sparse vector technique that the consumed privacy budget scales with the number of times that it uses the private voting, not with the total number of generated tokens. Consequently, the resulting algorithm, shown

Algorithm 2: DPVoteRAG

Require: Prompt x, External data source D, Generator LLM, Retriever R, # of voters m, # of retrieval per voter k, Per-token privacy budget $(\epsilon_{\text{token}}, \delta_{\text{token}})$, Total privacy budget $(\epsilon_{\text{total}}, \delta_{\text{total}})$

Ensure: Private answer y

1: $T_{\max} \leftarrow$ maximum # of tokens to generate based on $(\epsilon_{\text{token}}, \delta_{\text{token}})$ and $(\epsilon_{\text{total}}, \delta_{\text{total}})$
2: {Retrieval and random partition of the relevant documents}
3: $D_x \leftarrow$ Retrieve mk most relevant documents $R(x, D; mk)$.
4: $D_x^1, \cdots, D_x^m \leftarrow$ Uniformly randomly partition D_x into m disjoint subsets.
5: **for** $t \leftarrow 1$ **to** $T_{\max}$ **do**
6: {Generating the token histogram with the sensitive documents}
7: **for** $i \leftarrow 1$ **to** m **do**
8: $y_t^{(i)} \leftarrow \text{LLM}_t(x, D_x^i, y_{<t})$
9: **end for**
10: $h_t \leftarrow$ Build a histogram of tokens from $y_t^{(1)}, \ldots, y_t^{(m)}$
11: {Producing a token from the histogram privately}
12: $y_t \leftarrow \text{LimitedDomain}(h_t, \epsilon_{\text{token}}, \delta_{\text{token}})$
13: {Halting if end of sequence}
14: **if** $y_t =$ <EOS> **then**
15: **return** $(y_1, \ldots, y_t)$
16: **end if**
17: **end for**
18: **return** $(y_1, \ldots, y_{T_{\max}})$

in Algorithm 3, enables us to spend a privacy budget only when it needs sensitive information. We note that as a result of this change in our algorithm, we compute the maximum number of tokens to generate *with private voting*, $c_{\max}$, from the per-token privacy budget $(\epsilon_{\text{token}}, \delta_{\text{token}})$ and total privacy budget $(\epsilon_{\text{total}}, \delta_{\text{total}})$, instead of the maximum number of token generation, $T_{\max}$, as in DPVoteRAG but in the same way to compute $T_{\max}$. The formal privacy analysis of this algorithm is as follows. The guarantee holds due to the privacy guarantee of the LimitedDomain mechanism and the AboveThreshold algorithm [11].

Theorem 3. *For any question x, DPSparseVoteRAG satisfies $(\epsilon_{\text{total}}, \delta_{\text{total}})$-DP w.r.t. the external data source D.*

Proof. Similar to the proof of Theorem 2, we only need to prove that if $\mathcal{D}_x = (D_x^1, \cdots, D_x^m)$ at step 7 and $\mathcal{D}'_x = (D_x^{1'}, \cdots, D_x^{m'})$ differ at most one set D_x^i, $\Pr(y_1, \cdots, y_{T_{\max}}) \leq \varepsilon \cdot \Pr(y'_1, \cdots, y'_{T_{\max}}) + \delta$.

Denote t_γ as the first time step that holds $c = \gamma$ at the beginning of this time step, e.g. $t_{c_{\max}} = 1$. We can first prove that $\Pr(y_{t_\gamma \leq t < t_{\gamma-1}} | \mathcal{D}_x, y_{t<t_\gamma}) \leq \varepsilon_{\text{token}} \cdot \Pr(y'_{t'_\gamma \leq t < t'_{\gamma-1}} | \mathcal{D}'_x, y'_{t<t'_\gamma}) + \delta_{\text{token}}$ if (1) $t_\gamma = t'_\gamma$ and (2) $y_{t<t_\gamma} = y'_{t<t_\gamma}$.

This can be proved by two parts. First, because $t_\gamma = t'_\gamma$,

$$\Pr(t_{\gamma-1} | \mathcal{D}_x, y_{t<t_\gamma}) \leq \varepsilon_{\text{token}}/2 \cdot \Pr(t_{\gamma-1} | \mathcal{D}'_x, y'_{t<t'_\gamma}).$$

This is implied by applying the sparse vector technique presented in Algorithm 1 and Theorem 1 to analyze our algorithm (step 17-20). Furthermore, if $t_{\gamma-1} = t'_{\gamma-1}$,

$$\begin{aligned}
&\Pr(y_{t_\gamma \leq t < t_{\gamma-1}} | \mathcal{D}_x, y_{t<t_\gamma}) = \Pr(y^{non-RAG}_{t_\gamma \leq t < t_{\gamma-1}}, y^{DP}_{t_{\gamma-1}} | \mathcal{D}_x, y_{t<t_\gamma}) \\
&= \Pr(y^{non-RAG}_{t'_\gamma \leq t < t'_{\gamma-1}}, y^{DP}_{t'_{\gamma-1}} | \mathcal{D}_x, y'_{t<t'_\gamma}) \\
&\leq \varepsilon_{\text{token}}/2 \cdot \Pr(y^{non-RAG}_{t'_\gamma \leq t < t'_{\gamma-1}}, y^{DP'}_{t_{\gamma-1}} | \mathcal{D}'_x, y'_{t<t'_\gamma}) + \delta_{\text{token}}/2 \\
&= \Pr(y'_{t'_\gamma \leq t < t'_{\gamma-1}} | \mathcal{D}'_x, y'_{t<t_\gamma})
\end{aligned}$$

where the first and the last equality come from the definition of the algorithm (step 17-23), the second equality holds because we assume $t_\gamma = t'_\gamma$, $y_{t<t_\gamma} = y'_{t<t_\gamma}$ and $t_{\gamma-1} = t'_{\gamma-1}$, and the inequality comes from the DP guarantee by the LimitedDomain mechanism.

Lastly, our algorithm must stop before $c = 0$, means that our algorithm is a composition of at most $c_{\max}$ steps of $(\varepsilon_{\text{token}}, \delta_{\text{token}})$-DP. As shown in step 3 in our algorithm, $c_{\max}$ is picked to guarantee that the composition of $c_{\max}$ steps of $(\varepsilon_{\text{token}}, \delta_{\text{token}})$-DP is $(\varepsilon_{\text{total}}, \delta_{\text{total}})$-DP. Therefore, our algorithm is $(\varepsilon_{\text{total}}, \delta_{\text{total}})$-DP. □

IV. Experiment

We investigate how our differentially private voting RAG algorithms (Algorithms 2 and 3) work. Specifically, we ask the following questions:

1) How do our algorithms improve the accuracy of question-answering over non-RAG LLM while ensuring a formal privacy guarantee?
2) Is DPSparseVoteRAG (Algorithm 3) always a better choice than DPVoteRAG (Algorithm 2)?
3) Is there any useful guidance of choosing hyperparameters m (the number of voters) and ϵ_{token}?
4) How do our algorithm protect against empirical privacy attack?

We study each question through extensive evaluations on the well-used benchmarking datasets with multiple LLMs.

Algorithm 3: DPSparseVoteRAG

Require: Prompt x, External data source D, Generator LLM, Retriever R, # of voters m, # of retrieval per voter k, Per-token privacy budget $(\epsilon_{\text{token}}, \delta_{\text{token}})$, Total privacy budget $(\epsilon_{\text{total}}, \delta_{\text{total}})$, Threshold τ, Maximum # of output tokens (regardless of privacy) $T_{\max}$

Ensure: Private answer y

{Privacy budget setup}
$(\epsilon_{\text{token-RAG}}, \delta_{\text{token-RAG}}) \leftarrow (\epsilon_{\text{token}}/2, \delta_{\text{token}})$, $\epsilon_{\text{token-Lap}} \leftarrow \epsilon_{\text{token}}/2$
$c_{\max} \leftarrow$ maximum # of tokens to generate *privately* based on $(\epsilon_{\text{token}}, \delta_{\text{token}})$ and $(\epsilon_{\text{total}}, \delta_{\text{total}})$
$c \leftarrow c_{\max}$, $\hat{\tau} \leftarrow \tau + \text{Lap}(2/\epsilon_{\text{token-Lap}})$
{Retrieval and random partition of the relevant documents}
$D_x \leftarrow$ Retrieve mk most relevant documents $R(x, D; mk)$.
$D_x^1, \cdots, D_x^m \leftarrow$ Uniformly randomly partition D_x into m disjoint subsets.
for $t \leftarrow 1$ **to** $T_{\max}$ **do**
  {Generating the non-private token and token histogram with the sensitive documents}
  $y_t^{\text{non-RAG}} \leftarrow \text{LLM}_t(x, \text{“”}, y_{<t})$
  for $i \leftarrow 1$ **to** m **do**
    $y_t^{(i)} \leftarrow \text{LLM}_t(x, D_x^i, y_{<t})$
  end for
  $h_t \leftarrow$ Build a histogram of tokens from $y_t^{(1)}, \ldots, y_t^{(m)}$
  {Producing a token from the histogram privately only when $y_t^{\text{non-RAG}}$ is uncommon in h_t}
  $a_t \leftarrow$ Extract a count of h_t at $y_t^{\text{non-RAG}}$
  if $a_t + \text{Lap}(4/\epsilon_{\text{token-Lap}}) \leq \hat{\tau}$ **then**
    $y_t \leftarrow \text{LimitedDomain}(h_t, \epsilon_{\text{token-RAG}}, \delta_{\text{token-RAG}})$
    {The privacy budget is only consumed when y_t is from the histogram}
    $c \leftarrow c - 1$, $\hat{\tau} \leftarrow \tau + \text{Lap}(2/\epsilon_{\text{token-Lap}})$
  else
    $y_t \leftarrow y_t^{\text{non-RAG}}$
  end if
  {Halting if end of sequence or the privacy budget has been exhausted}
  if $y_t =$ <EOS> or $c = 0$ **then**
    return $(y_1, \ldots, y_t)$
  end if
end for
return $(y_1, \ldots, y_{T_{\max}})$

A. Methodology

a) Datasets.: We use two question-answering benchmarking datasets for RAG: **Trivia** [20] and **Natural Question (NQ)** [23]. Each dataset consists of a list of pairs of question and answer lists, i.e., every question can have multiple answers. By following the standard evaluations in RAG [6, 18, 25], we use the Wikipedia dataset as the external data source from which a retriever finds relevant documents. For each dataset, we use a subset of 100 questions to manage the computational overhead [4].

In addition, we experiment with a realistic privacy-sensitive application, where the external corpus contains inherently private information. Chatdoctor Questions [27] consist of QA interactions between patients and doctors in the healthcare domain. We sample 100 patient questions from the original dataset as our test set. The external dataset consists of the remaining QA pairs from the original ChatDoctor dataset, excluding the 100 patient questions used for testing. Here is a document example:

> *Patient's description*: My son fell from bed heads on, and didn t vomit or pass out. However, we put him to sleep as this was his sleep time. After two hours he woke up, and we felt he had a fever. We gave him brufen... ### *Doctor's answer*: Hi, according to me, I think since the kid fell and did not have vomiting, indicates there is no concussion. The fever is incidental, which can occur after a fall. This is normal. Give the kid paracetamol, probably every 6 hours once. If even after two days fever does not subside, kindly visit your pediatrician.... Take care.

[4]We first filter out questions to less than 20 ground truth documents in the Wikipedia dataset. If a question relates with only a few documents, DP algorithms will likely fail since replacing a document would change the output a lot. Then, we split the remaining questions into 4 bins with 20–29, 30–39, 40–49 and 50–59 relevant documents and sample 25 questions from each bin.

b) Models.: The retriever we use is the Dense Passage Retriever (DPR) [22] which is built on top of BERT [7]. It finds relevant documents that are close to the question in the embedding space produced by BERT. We compare the following generator LLMs: OPT (1.3B) [46], Llama 3.1 (8B) [9], and Pythia (1.4B) [4]. We additionally report the result of OPT (2.7B), Llama 3.2 (1B), and GPT2-XL [36] in Appendix A.

c) Algorithms.: We compare our algorithms, **DPVoteRAG** (Algorithm 2) and **DPSparseVoteRAG** (Algorithm 3), with two baseline algorithms. One baseline algorithm is **Non-RAG** where we only provide a question to the LLM without any relevant documents appended as a prompt. In order for our algorithms to be useful, they have to outperform this baseline. The other is **VoteRAG** where we carry out the same voting procedure as our algorithms but choose the most frequent token across voters non-privately—the most frequent token is always chosen as the next token to generate. For each number of voters, the result of this baseline serves as the upper bound of our DP algorithms.

d) Experimental Setup.: We observe the results under multiple total privacy budgets, $(\epsilon_{\text{total}}, \delta_{\text{total}})$. More specifically, we sweep $\epsilon_{\text{total}} = 2$ to 40 and set $\delta_{\text{total}} = 10^{-4}$. Furthermore, we consider different per-token privacy budgets for our private algorithms: $\epsilon_{\text{token}} = 1, 2, 5$ and $\delta_{\text{token}} = 10^{-5}$. We consider the number of voters m of 10, 20, 30, 40, and 50 for VoteRAG, and 30, 40, and 50 for DPVoteRAG and DPSparseVoteRAG so as to ensure reasonable privacy-utility tradeoff and computational overhead. For DPSparseVoteRAG, we set the threshold τ to be half of the number of voters, i.e., $\tau = m/2$. When we use the LimitedDomain mechanism to privately choose the most frequent token, we set their parameter $\bar{k}$ to be the number of voters, where $\bar{k}$ is the limited size of the domain to which we add the Gumbel noise. For voting algorithms, each voter receives 1 relevant document, i.e., $k = 1$. The utility evaluation metric is the match accuracy [3, 29, 38, 47] which measures if the prediction to a question contains any of its answers.

B. Main Results

a) Our RAG algorithms boost the QA accuracy even under a formal privacy guarantee.: Figure 3 shows the average match accuracy of baseline algorithms and our private algorithms under different total privacy guarantees (ϵ_{total}). Across different datasets and LLMs, we observe that DPSparseVoteRAG outperforms Non-RAG mostly under $\epsilon_{\text{total}} \geq 10$ and approaches the upper bound of VoteRAG as we allow a larger privacy budget. This demonstrates that our algorithms enable us to exploit the external knowledge through RAG to improve the utility of QA tasks *while* ensuring a reasonable level of privacy.

b) DPSparseVoteRAG is strictly better than DPVoteRAG.: In Figure 3, we find that DPSparseVoteRAG consistently outperforms DPVoteRAG across different LLMs and datasets. DPSparseVoteRAG augments DPVoteRAG by utilizing the non-RAG LLM and the sparse vector technique so that it only spends a privacy budget for an output token requiring sensitive external knowledge. The consistently better performances of DPSparseVoteRAG suggest the importance of separately treating token generations for meaningful tokens, i.e., tokens requiring external knowledge, and for other general tokens in the privacy-constraint setting.

c) ϵ_{token} should allow medium-length outputs. m should balance the DP noise and # of well-informed voters.: We take a closer look at the effects of the hyperparameters in Table I with OPT (1.3B) on Trivia dataset under different total privacy budgets ϵ_{total}. We provide the detailed results, as in Table I, with other LLMs in Appendix A.

Commonly between our private algorithms, we observe that the optimal ϵ_{token} increases as we allow more total privacy budgets. Under a tight total privacy budget, large ϵ_{token} allows our algorithms to only output a few meaningful tokens; thus, smaller ϵ_{token} is preferable. Conversely, under a large total privacy budget, accurate token generation with large ϵ_{token} is more important than having more tokens generated with small ϵ_{token}. Therefore, it is advised that we set ϵ_{token} to be as large as possible to enable accurate token generations *as long as* it is small enough to allow the algorithms to generate a reasonably large number of tokens (≈ 10). Notice that DPSparseVoteRAG generally allows us to set larger ϵ_{token} than DPVoteRAG under a fixed total privacy budget. This implies the benefit of DPSparseVoteRAG to save and spend a privacy budget cleverly—it can spend the saved privacy budget for generating important tokens for answering questions correctly.

With regard to the number of voters m, we generally see that more voters yield better utility with $\epsilon_{\text{token}} = 1$, but the number of voters has less effect on the utility with larger ϵ_{token}. This is due to the two distinct consequences of having more voters. More voters alleviate the effect of DP noise on the token histograms constructed in the algorithms. However, depending on the number of relevant documents to the question, there is a risk of having voters with irrelevant documents who can vote for the wrong tokens. The first consequence is more dominant particularly under small ϵ_{token} while the second is more dominant under larger ϵ_{token}. Hence, m should be set to balance these two consequences for achieving better per-token generation quality.

C. Empirical Privacy Evaluation

To assess the degree of privacy protection offered by our proposed method, we evaluate the vulnerability of both a non-private RAG system and our privacy-preserving RAG system on the privacy-sensitive ChatDoctor dataset using membership inference attacks (MIA). Given a target document x and a system f_D, an MIA computes a score $s(x, f)$ that reflects the likelihood of $x \in D$. Without loss of generality, we assume higher scores indicate a greater probability of membership. By applying the attack to two sets of documents (an in-distribution set $D_{\text{in}} \subset D$ and an out-of-distribution set D_{out} with no overlap with D), we can derive a TPR–FPR curve and compute its AUC.

We adopt the membership score design from S^2MIA [28]. In the ChatDoctor dataset, each document corresponds to a

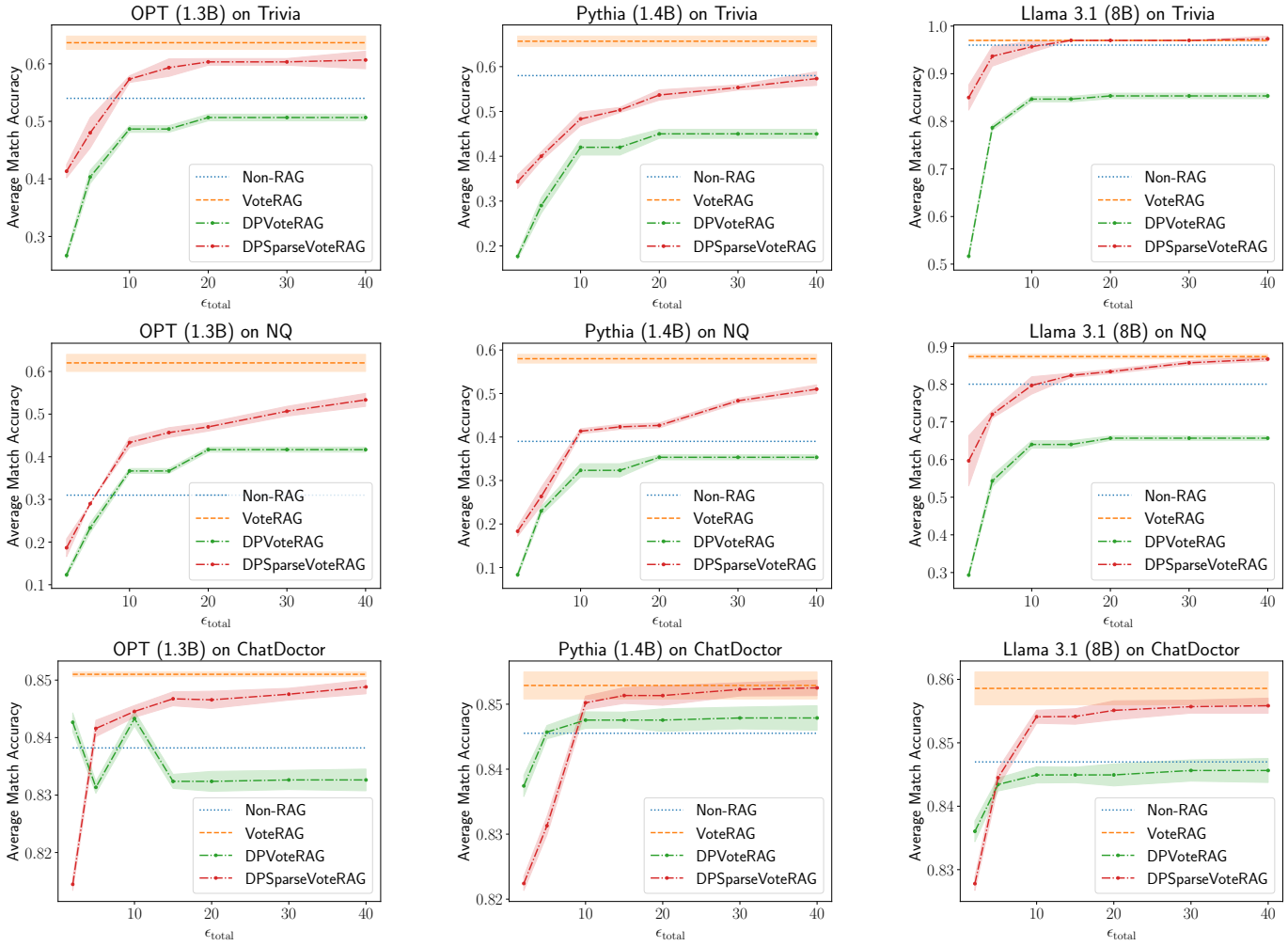

Fig. 3: Average match accuracy comparison across algorithms on Trivia (upper row) and NQ (lower row) datasets with different generator LLMs: OPT (1.3B) (left column), Pythia (1.4B) (middle column), and Llama 3.1 (8B) (right column). The reported results are the means and standard deviations of average match accuracy over three runs. We report the best results over hyperparameters for each ϵ_{total}.

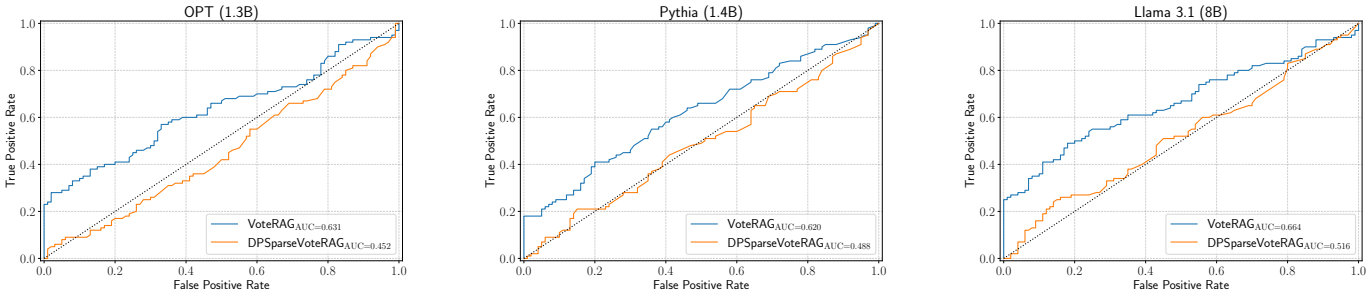

Fig. 4: TPR-FPR curve of S^{2}MIA for VoteRAG and DPSparseVoteRAG (ε=10) when the base LLMs are OPT (1.3B), Pythia (1.4B) and Llama 3.1 (8B).

patient–doctor conversation. For a target document x, we extract the patient's query x_t^q and measure the similarity between the response x_t^r generated by the tested RAG system and the doctor's ground-truth answer x_t^g in x. The similarity is quantified using the average precision score defined in BLEU [33], which serves directly as the membership score in S^{2}MIA.

Figure 4 presents the TPR–FPR curves and the corresponding AUC values. Without any privacy protection (VoteRAG), the attack is highly effective, yielding AUC values well above

TABLE I: Average match accuracy comparison of our algorithms on Trivia dataset with OPT (1.3B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}} = 5$			$\epsilon_{\text{total}} = 10$			$\epsilon_{\text{total}} = 20$			$\epsilon_{\text{total}} = 40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}} = 1$		0.37	**0.40**	**0.40**	0.45	**0.49**	**0.49**	0.45	0.49	0.49	0.45	0.49	0.49
$\epsilon_{\text{token}} = 2$		0.25	0.27	0.27	0.42	0.41	0.41	**0.51**	0.50	0.49	**0.51**	0.50	0.49
$\epsilon_{\text{token}} = 5$		0.12	0.12	0.12	0.25	0.27	0.27	0.38	0.38	0.38	0.46	0.46	0.44
DPSparseVoteRAG													
$\epsilon_{\text{token}} = 1$		0.28	0.37	0.45	0.28	0.38	0.46	0.28	0.38	0.46	0.28	0.38	0.46
$\epsilon_{\text{token}} = 2$		0.42	0.45	**0.48**	0.48	0.55	**0.57**	0.49	0.57	**0.60**	0.49	0.57	0.60
$\epsilon_{\text{token}} = 5$		0.33	0.33	0.34	0.47	0.48	0.49	0.56	0.55	0.55	0.59	0.60	**0.61**

the diagonal baseline (0.5). In contrast, when querying our privacy-preserving system DPSparseVoteRAG with $\varepsilon = 10$, the attack performance collapses to the naive baseline (AUC ≈ 0.5). This demonstrates that our method effectively mitigates empirical privacy attacks while maintaining strong utility on the QA task, as shown in Figure 3.

D. More Analysis of DPSparseVoteRAG and DPVoteRAG

a) The length of generation.: The design of DPSparseVoteRAG might allow longer generation than DPVoteRAG because of the tighter composition from SVT. We empirically validated this intuition. Figure 5 shows the numbers of tokens generated by DPVoteRAG and DPSparseVoteRAG. As we expect by the design of DPSparseVoteRAG, we see DPSparseVoteRAG generates much more tokens than DPVoteRAG. This implies the effectiveness of the sparse vector technique in DPSparseVoteRAG to smartly spend privacy budget enabling long enough token sequences.

b) Effects of number of ground truth relevant documents.: Figure 6 shows the performances for different numbers of ground truth relevant documents. We see questions with more relevant documents tend to be answered correctly by our algorithm.

V. Related Work

a) Privacy-preserving algorithms in large language models.: Zeng et al. [44] proposed an empirical privacy-preserving algorithm for RAG through the synthetic data generation, while our work studies privacy-preserving RAG in the framework differential privacy, which protects the privacy of each individual document with the theoretical guarantee. Differential privacy has been studied in many other tasks in large language models too. *Prompt tuning* helps tailor the LLM to new tasks from a (private) test-domain dataset. Hong et al. [17] and Duan et al. [8] study the DP mechanism on two different prompt tuning frameworks: prompt optimization and offsite prompt tuning [39]. *In-context learning* adapts to different tasks by illustrating some examples in the context as the task description. DP in-context learning considers the situation when the examples are picked from any private set. Tang et al. [40] tackles this problem by generating synthetic examples with DP and Wu et al. [42] solves the DP test query by generating the answers, both in a sample-and-aggregate fashion. Amin et al. [2] proposes the aggregation based method to generate synthetic texts with DP, which applies the similar SVT idea of our methodology to save the budget for some tokens. The differentially private *pretraining and finetuning* of LLMs has been studied to address the privacy concern in the training data and memory is a large bottleneck when naively deploying DP-SGD [1]. Li et al. [26] focuses on the pretraining stage which introduces ghost clipping to make DP-SGD more memory efficient. Yu et al. [43] explores finetuning in the parameter-efficient framework LoRA [19]. Notice that DP voting plays a crucial role in these sample-and-aggregate algorithms, including ours. A basic approach is to apply the Laplacian or Gaussian mechanism [11]. Papernot et al. [31, 32] proposed a data-dependent privacy analysis, which can be tighter when the majority vote has a large margin over other options. We integrate the LimitedDomain mechanism for our algorithm, which addresses challenges when the voting domain is large [10]; the large vocabulary size in token voting is our main bottleneck.

b) Composition in differential privacy.: Our algorithms generate the answers token by token, where each token needs a query to the private dataset and consumes some privacy budget. In this paper, we set up the privacy parameters before the start of the algorithm and have a pre-set maximum number of tokens to generate. However, the number of tokens to generate is different per question and is unknown before the algorithm starts – it is possible that the number of generated tokens is much smaller than the pre-set number but we still need to pay the full pre-defined privacy cost. A line of work [16, 24, 37, 41] tries to measure the privacy budget in fully adaptive composition where the budget consuming can interact with the data. Especially, Whitehouse et al. [41] gives an analysis for this fully adaptive setting which matches the tightness of advanced composition. The idea of fully adaptive composition sounds a fit to our problem, which allows us to "pay as we go", rather than predefining the ϵ_{total} before the generation process. We found the analysis for fully adaptive setting is effective for large number of steps and small budget per step, while in our algorithm the number of generated tokens would not be very large and each token generation needs a relatively

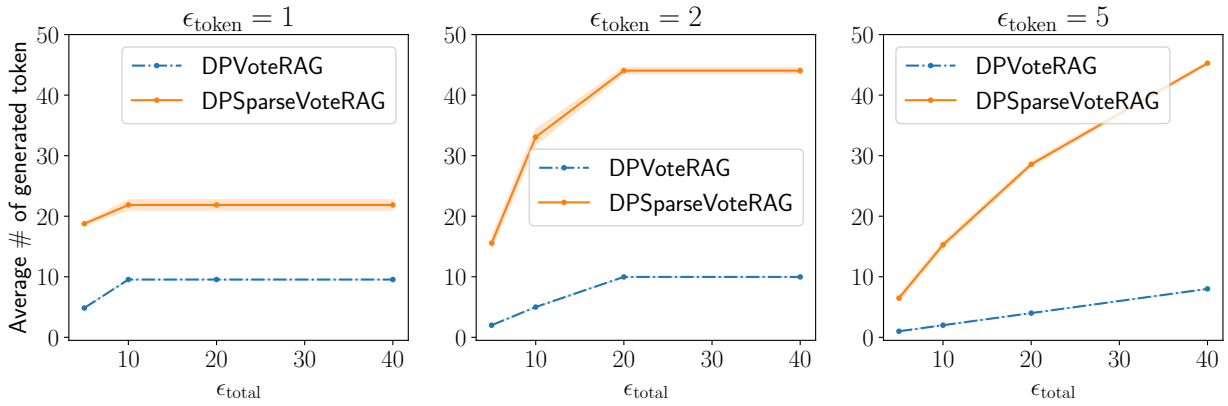

Fig. 5: Average numbers of generated tokens for each ϵ_{token} and ϵ_{total} with OPT (1.3B) on Trivia dataset. We fix $m = 50$. We report the means and standard deviations over three runs. We see DPSparseVoteRAG generates much more tokens than DPVoteRAG.

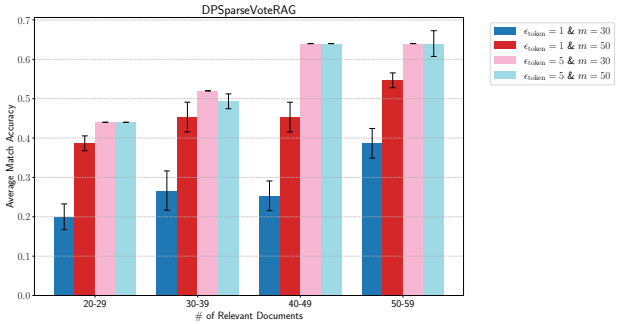

Fig. 6: Average match accuracy for questions with different numbers of ground truth relevant documents across baseline algorithms and our algorithms with OPT (1.3B) on Trivia dataset. We report the means and standard deviations over three runs. We see questions with more relevant documents tend to be answered correctly by our algorithm.

large budget to guarantee the utility. This mismatch makes us stick with the advanced composition.

VI. Conclusion and Future Work

We introduce the first differentially private algorithms for RAG, enabling us to enhance LLMs by domain-specific but sensitive external corpus. With our novel combination of the DP voting algorithm and sparse vector technique along with the non-private LLM, we succeed in spending privacy budget only when the LLM needs sensitive information to generate a new token. Consequently, DPSparseVoteRAG generates a sufficiently long and accurate response under a reasonable privacy budget. Our experiments demonstrate that our algorithms outperform the non-RAG baseline across different datasets and models, showing their effectiveness.

One of our future directions is to conduct more practical empirical evaluations. The Wikipedia dataset, which we use as the external data source, is typically included in the training data of recent LLMs. RAG is particularly effective when the external knowledge is *truly sensitive* and thus outside the LLM training data. It is essential for us to conduct evaluations that are as close to the real situation as possible and see how effective our algorithms are over non-RAG LLMs. Since our usage of the sparse vector technique is applicable to any DP token generation algorithm through voting, another future direction would be to examine how it improves DP token generation across different tasks, e.g., in-context learning and prompt tuning.

LLM Usage Considerations

We primarily use LLMs to refine the grammar and clarity of our writing, while the core ideas and research progress are developed independently through our own study and investigation.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Kareem Amin, Alex Bie, Weiwei Kong, Alexey Kurakin, Natalia Ponomareva, Umar Syed, Andreas Terzis, and Sergei Vassilvitskii. Private prediction for large-scale synthetic text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7244–7262, 2024.

[3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*, October 2023.

[4] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2397–2430. PMLR, July 2023.

[5] Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A. Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. Phantom: General Trigger Attacks on Retrieval Augmented Language Generation, October 2024.

[6] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi:10.18653/v1/P17-1171.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

[8] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[10] David Durfee and Ryan M Rogers. Practical Differentially Private Top-k Selection with Pay-what-you-get Composition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi:10.1561/0400000042.

[12] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, Lecture Notes in Computer Science, pages 486–503, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-34547-3. doi:10.1007/11761679_29.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 265–284, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-32732-5. doi:10.1007/11681878_14.

[14] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 381–390, New York, NY, USA, May 2009. Association for Computing Machinery. ISBN 978-1-60558-506-2. doi:10.1145/1536414.1536467.

[15] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and Differential Privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, October 2010. doi:10.1109/FOCS.2010.12.

[16] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems*, 34:28080–28091, 2021.

[17] Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. DP-OPT: Make large language model your privacy-preserving prompt engineer. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Ifz3IgsEPX.

[18] Jennifer Hsia, Afreen Shaikh, Zhiruo Wang, and Graham Neubig. Ragged: Towards informed design of retrieval augmented generation systems. *arXiv preprint arXiv:2403.09040*, 2024.

[19] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[20] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi:10.18653/v1/P17-1147.

[21] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The Composition Theorem for Differential Privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, June 2017. ISSN 0018-9448, 1557-9654. doi:10.1109/TIT.2017.2685505.

[22] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, January 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.emnlp-main.550.

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi:10.1162/tacl_a_00276.

[24] Mathias Lécuyer. Practical privacy filters and odometers with r\'enyi differential privacy and applications to differentially private deep learning. *arXiv preprint arXiv:2103.01379*, 2021.

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[26] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=bVuP3ltATMz.

[27] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

[28] Yuying Li, Gaoyang Liu, Chen Wang, and Yang Yang. Generating is believing: Membership inference attacks against retrieval-augmented generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[29] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.acl-long.546.

[30] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, June 2007. Association for Computing Machinery. ISBN 978-1-59593-631-8. doi:10.1145/1250790.1250803.

[31] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *5th International Conference on Learning Representations (ICLR), 2017*. arXiv, March 2017. doi:10.48550/arXiv.1610.05755.

[32] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable Private Learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[34] Yuefeng Peng, Junda Wang, Hong Yu, and Amir Houmansadr. Data Extraction Attacks in Retrieval-Augmented Generation via Backdoors, November 2024.

[35] Zhenting Qi, Hanlin Zhang, Eric Xing, Sham Kakade, and Himabindu Lakkaraju. Follow My Instruction and Spill the Beans: Scalable Data Extraction from Retrieval-Augmented Generation Systems, October 2024.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners.

[37] Ryan M Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. *Advances in Neural Information Processing Systems*, 29, 2016.

[38] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

[39] Alessandro Sordoni, Xingdi Yuan, Marc-Alexandre Côté,

Matheus Pereira, Adam Trischler, Ziang Xiao, Arian Hosseini, Friederike Niedtner, and Nicolas Le Roux. Joint prompt optimization of stacked LLMs using variational inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=iImnbUVhok.

[40] Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oZtt0pRnOl.

[41] Justin Whitehouse, Aaditya Ramdas, Ryan Rogers, and Steven Wu. Fully-adaptive composition in differential privacy. In *International Conference on Machine Learning*, pages 36990–37007. PMLR, 2023.

[42] Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=x4OPJ7lHVU.

[43] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q42f0dfjECO.

[44] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4505–4524, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.findings-acl.267. URL https://aclanthology.org/2024.findings-acl.267.

[45] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4505–4524, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.findings-acl.267.

[46] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open Pre-trained Transformer Language Models, June 2022.

[47] Zihan Zhang, Meng Fang, and Ling Chen. RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6963–6975, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.findings-acl.415.

APPENDIX

A. Additional Experimental Results

In Figures 7–9, we present the average match accuracy of baseline algorithms and our algorithms for different total privacy guarantees (ϵ_{total}) with OPT (2.7B), Llama 3.2 (1B), and GPT2-XL. We see the similar trend observed in Figure 3.

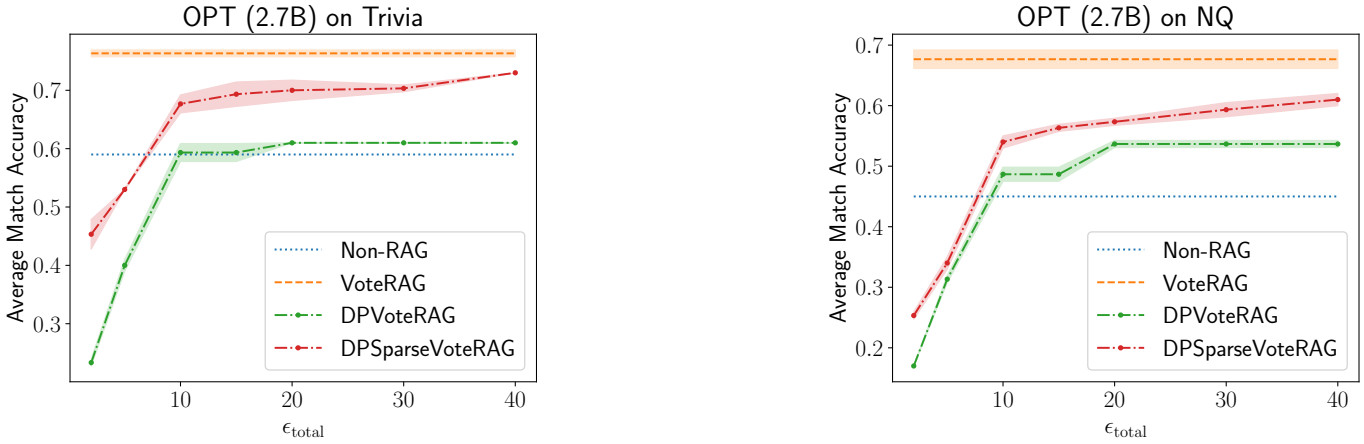

Fig. 7: Average match accuracy comparison across algorithms on Trivia (left) and NQ (right) datasets with OPT (2.7B). The reported results are the means and standard deviations of average match accuracy over three runs. We report the best results over hyperparameters for each ϵ_{total}.

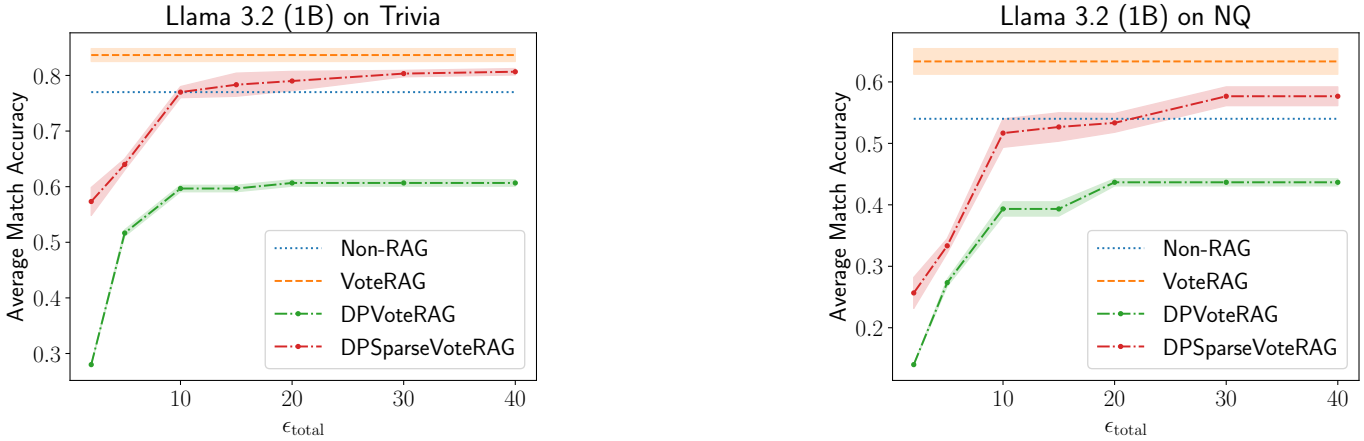

Fig. 8: Average match accuracy comparison across algorithms on Trivia (left) and NQ (right) datasets with Llama 3.2 (1B). The reported results are the means and standard deviations of average match accuracy over three runs. We report the best results over hyperparameters for each ϵ_{total}.

For completeness, we further present the detailed results, in the same form of the one provided in Table I, with OPT (1.3B and 2.7B), Pythia (1.4B), Llama 3.1 (8B), Llama 3.2 (1B), and GPT2-XL on Trivia and NQ datasets in Tables II–XV.

TABLE II: Average match accuracy comparison of our algorithms on NQ dataset with OPT (1.3B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.17	**0.23**	**0.23**	0.28	0.35	**0.37**	0.28	0.35	0.37	0.28	0.35	0.37
$\epsilon_{\text{token}}=2$		0.12	0.14	0.13	0.24	0.25	0.26	0.38	**0.42**	**0.42**	0.38	**0.42**	**0.42**
$\epsilon_{\text{token}}=5$		0.04	0.06	0.05	0.12	0.14	0.13	0.22	0.22	0.21	0.36	0.36	0.35
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.14	0.18	0.28	0.14	0.18	0.29	0.14	0.18	0.29	0.14	0.18	0.29
$\epsilon_{\text{token}}=2$		0.23	0.25	**0.29**	0.37	0.41	**0.43**	0.39	0.44	**0.47**	0.39	0.44	0.47
$\epsilon_{\text{token}}=5$		0.11	0.14	0.13	0.28	0.31	0.29	0.43	0.43	0.42	**0.53**	**0.53**	**0.53**

TABLE III: Average match accuracy comparison of our algorithms on Trivia dataset with OPT (2.7B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.34	**0.40**	0.39	0.48	0.58	**0.59**	0.48	0.58	0.59	0.48	0.58	0.59
$\epsilon_{\text{token}}=2$		0.24	0.23	0.23	0.41	0.41	0.40	0.59	**0.61**	0.60	0.59	**0.61**	0.60
$\epsilon_{\text{token}}=5$		0.09	0.10	0.10	0.23	0.24	0.23	0.38	0.39	0.38	0.49	0.51	0.51
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.31	0.48	**0.53**	0.31	0.49	0.53	0.31	0.49	0.53	0.31	0.49	0.53
$\epsilon_{\text{token}}=2$		0.50	0.52	**0.53**	0.57	0.66	**0.68**	0.58	0.69	**0.70**	0.58	0.69	0.70
$\epsilon_{\text{token}}=5$		0.35	0.38	0.37	0.54	0.54	0.54	0.66	0.66	0.66	**0.73**	0.71	0.70

TABLE IV: Average match accuracy comparison of our algorithms on NQ dataset with OPT (2.7B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.26	0.30	**0.31**	0.39	0.48	**0.49**	0.39	0.48	0.49	0.39	0.48	0.49
$\epsilon_{\text{token}}=2$		0.20	0.18	0.17	0.36	0.34	0.33	**0.54**	0.51	0.51	**0.54**	0.51	0.51
$\epsilon_{\text{token}}=5$		0.06	0.05	0.05	0.20	0.17	0.17	0.34	0.31	0.30	0.46	0.43	0.42
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.16	0.24	0.32	0.16	0.24	0.32	0.16	0.24	0.32	0.16	0.24	0.32
$\epsilon_{\text{token}}=2$		0.31	0.32	**0.34**	0.46	0.52	**0.54**	0.47	0.53	**0.57**	0.47	0.53	0.57
$\epsilon_{\text{token}}=5$		0.12	0.12	0.11	0.38	0.35	0.35	**0.57**	0.53	0.53	**0.61**	0.58	0.60

TABLE V: Average match accuracy comparison of our algorithms on Trivia dataset with Pythia (1.4B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.24	0.28	**0.29**	0.34	0.41	**0.42**	0.34	0.41	0.42	0.34	0.41	0.42
$\epsilon_{\text{token}}=2$		0.18	0.18	0.19	0.30	0.31	0.32	0.43	**0.45**	0.44	0.43	**0.45**	0.44
$\epsilon_{\text{token}}=5$		0.06	0.05	0.07	0.18	0.18	0.19	0.29	0.30	0.30	0.38	0.39	0.37
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.25	0.32	0.38	0.25	0.32	0.39	0.25	0.32	0.39	0.25	0.32	0.39
$\epsilon_{\text{token}}=2$		0.37	**0.40**	**0.40**	0.42	0.47	**0.48**	0.43	0.53	**0.54**	0.43	0.53	0.54
$\epsilon_{\text{token}}=5$		0.25	0.25	0.26	0.43	0.44	0.41	0.51	0.53	0.51	**0.57**	**0.57**	**0.57**

TABLE VI: Average match accuracy comparison of our algorithms on NQ dataset with Pythia (1.4B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm	$\epsilon_{\text{total}} = 5$			$\epsilon_{\text{total}} = 10$			$\epsilon_{\text{total}} = 20$			$\epsilon_{\text{total}} = 40$		
m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG												
$\epsilon_{\text{token}} = 1$	0.16	**0.23**	**0.23**	0.21	0.30	**0.32**	0.21	0.30	0.32	0.21	0.30	0.32
$\epsilon_{\text{token}} = 2$	0.09	0.08	0.09	0.23	0.26	0.26	0.32	**0.35**	**0.35**	0.32	**0.35**	**0.35**
$\epsilon_{\text{token}} = 5$	0.05	0.04	0.04	0.10	0.09	0.09	0.23	0.24	0.23	0.31	0.32	0.30
DPSparseVoteRAG												
$\epsilon_{\text{token}} = 1$	0.09	0.17	0.24	0.09	0.17	0.24	0.09	0.17	0.24	0.09	0.17	0.24
$\epsilon_{\text{token}} = 2$	0.20	0.25	**0.26**	0.25	0.36	**0.41**	0.26	0.37	**0.43**	0.26	0.37	0.43
$\epsilon_{\text{token}} = 5$	0.14	0.14	0.15	0.27	0.28	0.27	0.39	0.41	**0.43**	0.46	**0.51**	0.49

TABLE VII: Average match accuracy comparison of our algorithms on Trivia dataset with Llama 3.1 (8B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm	$\epsilon_{\text{total}} = 5$			$\epsilon_{\text{total}} = 10$			$\epsilon_{\text{total}} = 20$			$\epsilon_{\text{total}} = 40$		
m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG												
$\epsilon_{\text{token}} = 1$	0.75	0.77	**0.79**	0.81	0.84	**0.85**	0.81	0.84	**0.85**	0.81	0.84	**0.85**
$\epsilon_{\text{token}} = 2$	0.51	0.51	0.51	0.78	0.78	0.78	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**
$\epsilon_{\text{token}} = 5$	0.20	0.21	0.21	0.51	0.51	0.51	0.73	0.74	0.74	0.83	0.82	0.82
DPSparseVoteRAG												
$\epsilon_{\text{token}} = 1$	0.72	0.83	0.88	0.72	0.83	0.88	0.72	0.83	0.88	0.72	0.83	0.88
$\epsilon_{\text{token}} = 2$	0.89	**0.94**	**0.94**	0.93	**0.96**	**0.96**	0.93	0.96	0.96	0.93	0.96	0.96
$\epsilon_{\text{token}} = 5$	0.76	0.78	0.76	0.93	0.93	0.93	0.95	**0.97**	**0.97**	0.95	**0.97**	**0.97**

TABLE VIII: Average match accuracy comparison of our algorithms on NQ dataset with Llama 3.1 (8B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm	$\epsilon_{\text{total}} = 5$			$\epsilon_{\text{total}} = 10$			$\epsilon_{\text{total}} = 20$			$\epsilon_{\text{total}} = 40$		
m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG												
$\epsilon_{\text{token}} = 1$	0.49	**0.54**	**0.54**	0.57	0.63	**0.64**	0.57	0.63	0.64	0.57	0.63	0.64
$\epsilon_{\text{token}} = 2$	0.30	0.29	0.30	0.55	0.56	0.55	0.64	**0.66**	0.65	0.64	**0.66**	0.65
$\epsilon_{\text{token}} = 5$	0.11	0.11	0.11	0.29	0.29	0.30	0.52	0.52	0.51	0.62	0.63	0.62
DPSparseVoteRAG												
$\epsilon_{\text{token}} = 1$	0.38	0.55	0.65	0.38	0.55	0.66	0.38	0.55	0.66	0.38	0.55	0.66
$\epsilon_{\text{token}} = 2$	0.64	**0.72**	**0.72**	0.71	0.78	**0.80**	0.71	0.79	**0.83**	0.71	0.79	0.83
$\epsilon_{\text{token}} = 5$	0.48	0.49	0.48	0.74	0.75	0.74	0.79	0.80	0.79	0.85	**0.87**	0.85

TABLE IX: Average match accuracy comparison of our algorithms on Trivia dataset with Llama 3.2 (1B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm	$\epsilon_{\text{total}} = 5$			$\epsilon_{\text{total}} = 10$			$\epsilon_{\text{total}} = 20$			$\epsilon_{\text{total}} = 40$		
m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG												
$\epsilon_{\text{token}} = 1$	0.46	0.49	**0.52**	0.54	0.57	**0.60**	0.54	0.57	0.60	0.54	0.57	0.60
$\epsilon_{\text{token}} = 2$	0.27	0.27	0.28	0.51	0.52	0.52	0.60	**0.61**	0.60	0.60	**0.61**	0.60
$\epsilon_{\text{token}} = 5$	0.10	0.10	0.10	0.27	0.27	0.28	0.47	0.47	0.47	0.56	0.57	0.56
DPSparseVoteRAG												
$\epsilon_{\text{token}} = 1$	0.38	0.52	0.63	0.38	0.52	0.63	0.38	0.52	0.63	0.38	0.52	0.63
$\epsilon_{\text{token}} = 2$	0.61	0.62	**0.64**	0.70	0.76	**0.77**	0.71	0.78	**0.79**	0.71	0.78	0.79
$\epsilon_{\text{token}} = 5$	0.43	0.43	0.45	0.66	0.64	0.66	0.76	0.75	0.76	**0.81**	0.80	0.79

TABLE X: Average match accuracy comparison of our algorithms on NQ dataset with Llama 3.2 (1B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.24	**0.27**	**0.27**	0.31	0.38	**0.39**	0.31	0.38	0.39	0.31	0.38	0.39
$\epsilon_{\text{token}}=2$		0.15	0.15	0.12	0.30	0.31	0.27	0.42	**0.44**	0.42	0.42	**0.44**	0.42
$\epsilon_{\text{token}}=5$		0.05	0.05	0.05	0.15	0.15	0.13	0.30	0.30	0.27	0.38	0.39	0.35
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.14	0.25	0.32	0.14	0.25	0.34	0.14	0.25	0.34	0.14	0.25	0.34
$\epsilon_{\text{token}}=2$		0.31	**0.33**	**0.33**	0.41	0.49	**0.52**	0.42	0.50	**0.53**	0.42	0.50	0.53
$\epsilon_{\text{token}}=5$		0.15	0.14	0.13	0.40	0.36	0.32	0.53	0.51	**0.53**	0.57	0.56	**0.58**

TABLE XI: Average match accuracy comparison of our algorithms on Trivia dataset with GPT2-XL under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.31	0.32	**0.34**	0.36	0.38	**0.42**	0.36	0.38	0.42	0.36	0.38	0.42
$\epsilon_{\text{token}}=2$		0.22	0.23	0.23	0.37	0.38	0.37	0.44	**0.46**	**0.46**	0.44	**0.46**	**0.46**
$\epsilon_{\text{token}}=5$		0.08	0.08	0.08	0.23	0.23	0.23	0.34	0.34	0.33	0.44	0.43	0.42
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.25	0.30	0.38	0.25	0.30	0.38	0.25	0.30	0.38	0.25	0.30	0.38
$\epsilon_{\text{token}}=2$		0.37	**0.39**	0.38	0.41	0.46	**0.47**	0.42	0.47	0.48	0.42	0.47	0.48
$\epsilon_{\text{token}}=5$		0.28	0.27	0.27	0.43	0.43	0.43	0.52	**0.53**	**0.53**	**0.59**	0.57	0.56

TABLE XII: Average match accuracy comparison of our algorithms on NQ dataset with GPT2-XL under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.21	0.24	**0.27**	0.25	0.28	**0.33**	0.25	0.28	0.33	0.25	0.28	0.33
$\epsilon_{\text{token}}=2$		0.18	0.19	0.19	0.30	0.29	0.29	0.35	0.35	**0.36**	0.35	0.35	**0.36**
$\epsilon_{\text{token}}=5$		0.08	0.08	0.07	0.19	0.19	0.19	0.30	0.29	0.29	**0.36**	0.34	0.34
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.14	0.19	0.24	0.14	0.19	0.24	0.14	0.19	0.24	0.14	0.19	0.24
$\epsilon_{\text{token}}=2$		0.24	0.28	**0.31**	0.28	0.33	**0.36**	0.29	0.35	0.40	0.29	0.35	0.40
$\epsilon_{\text{token}}=5$		0.18	0.18	0.16	0.34	0.33	0.35	**0.41**	0.39	**0.41**	0.42	0.41	**0.43**

TABLE XIII: Average match accuracy comparison of our algorithms on ChatDoctor dataset with OPT (1.3B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.82	**0.83**	**0.83**	0.82	0.83	0.83	0.82	**0.83**	**0.83**	0.82	**0.83**	**0.83**
$\epsilon_{\text{token}}=2$		0.82	0.82	0.81	0.83	0.83	0.83	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**	**0.83**
$\epsilon_{\text{token}}=5$		0.82	0.82	0.82	**0.84**	**0.84**	**0.84**	0.82	0.82	0.82	**0.83**	**0.83**	**0.83**
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.79	0.80	0.81	0.79	0.80	0.81	0.79	0.80	0.81	0.79	0.80	0.81
$\epsilon_{\text{token}}=2$		0.83	**0.84**	**0.84**	0.82	**0.84**	**0.84**	0.82	0.84	**0.85**	0.82	0.84	**0.85**
$\epsilon_{\text{token}}=5$		0.13	0.13	0.13	**0.84**	**0.84**	**0.84**	0.84	0.84	0.84	**0.85**	**0.85**	**0.85**

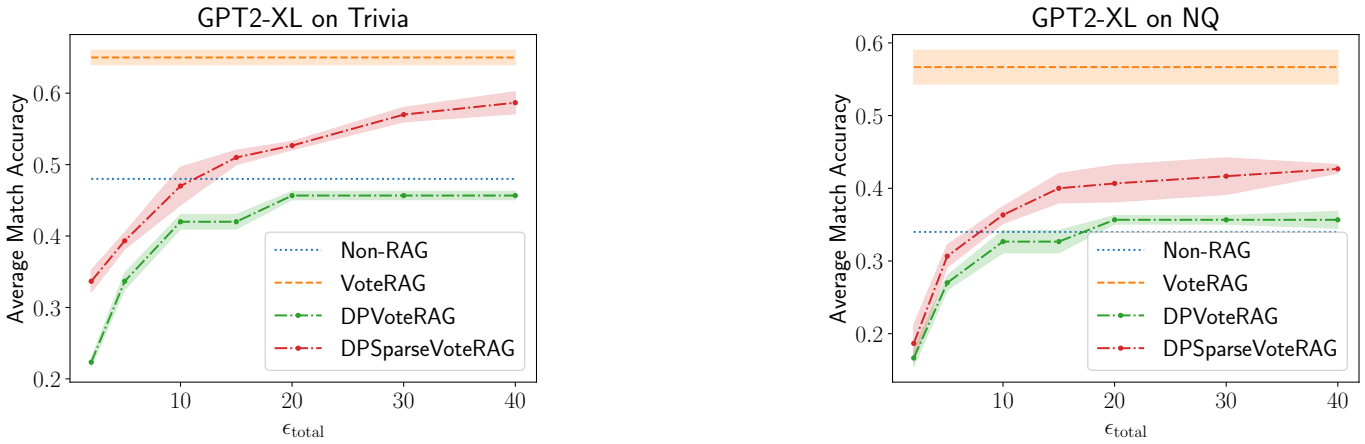

Fig. 9: Average match accuracy comparison across algorithms on Trivia (left) and NQ (right) datasets with GPT2-XL. The reported results are the means and standard deviations of average match accuracy over three runs. We report the best results over hyperparameters for each ϵ_{total}.

TABLE XIV: Average match accuracy comparison of our algorithms on ChatDoctor dataset with Pythia (1.4B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.82	0.84	**0.85**	0.82	0.84	**0.85**	0.82	0.84	**0.85**	0.82	0.84	**0.85**
$\epsilon_{\text{token}}=2$		0.82	0.82	0.82	0.84	**0.85**	**0.85**	0.84	**0.85**	**0.85**	0.84	**0.85**	**0.85**
$\epsilon_{\text{token}}=5$		0.84	0.84	0.84	0.81	0.82	0.81	0.84	0.84	0.84	**0.85**	**0.85**	**0.85**
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.79	0.80	0.81	0.79	0.80	0.81	0.79	0.80	0.81	0.79	0.80	0.81
$\epsilon_{\text{token}}=2$		0.82	0.82	**0.83**	0.83	0.84	**0.85**	0.83	**0.85**	**0.85**	0.83	**0.85**	**0.85**
$\epsilon_{\text{token}}=5$		0.82	0.82	0.82	0.83	0.83	0.83	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**

B. *Revisions after the Last Submission*

We addressed the reviews raised from the last submission. Here are the details.

a) Evaluation on realistic private dataset.: We additionally evaluate the methods with ChatDoctor dataset, which is supposed to be unseen in the pre-train stage and is practically private and sensitive. Please check the details of our experimental results.

TABLE XV: Average match accuracy comparison of our algorithms on ChatDoctor dataset with Llama 3.1 (8B) under varying values of total privacy budget ϵ_{total} with different hyperparameters, the number of voters m and ϵ_{token}. Bold font values represent the best performance of each algorithm under fixed ϵ_{total}. We report the means of average match accuracy over three runs.

Algorithm		$\epsilon_{\text{total}}=5$			$\epsilon_{\text{total}}=10$			$\epsilon_{\text{total}}=20$			$\epsilon_{\text{total}}=40$		
	m	30	40	50	30	40	50	30	40	50	30	40	50
DPVoteRAG													
$\epsilon_{\text{token}}=1$		0.83	**0.84**	**0.84**	0.83	**0.84**	**0.84**	0.83	**0.84**	**0.84**	0.83	0.84	0.84
$\epsilon_{\text{token}}=2$		0.83	0.83	0.83	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	0.84	0.84	0.84
$\epsilon_{\text{token}}=5$		0.82	0.82	0.82	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.85**	0.84	0.84
DPSparseVoteRAG													
$\epsilon_{\text{token}}=1$		0.82	0.83	0.82	0.82	0.83	0.82	0.82	0.83	0.82	0.82	0.83	0.82
$\epsilon_{\text{token}}=2$		**0.84**	**0.84**	**0.84**	**0.85**	**0.85**	**0.85**	0.85	0.85	0.85	0.85	0.85	0.85
$\epsilon_{\text{token}}=5$		0.46	0.45	0.35	0.84	**0.85**	0.84	0.85	**0.86**	0.85	**0.86**	**0.86**	**0.86**

b) Our method only beats nonrag baseline when $\varepsilon = 10$, which seems impractical.: We additionally evaluate our method against with empirical privacy attack and the results show that $\varepsilon = 10$ is sufficient to perfectly defend against the existing attack.

c) Complete proof of theorems.: We provided the full proofs for both theorems.

d) Discussion on more related work.: We distinguish our work with more related work in the literature in our related work section.

Reviewer 1

Summary:
The paper studies the problem of private retrieval augmented generation: given a database of sensitive texts, we want to be able to answer user questions using these texts while preserving privacy (of the users that contributed documents to the database).

The mechanism described in the paper uses subsample and aggregate idea and generates tokens of the response one by one.

1. First, it retrieves relevant documents.
2. It splits the documents into parts.
3. It assembles each part into prompt,
4. It predicts the next token for each prompt.
5. Using DP voting mechanism it selects the most common token and appends it to the prompts.
6. If the token is not "END OF RESPONSE" it goes to step 4.

It is easy to see why the mechanism is DP; however, utility of the method is non-trivial and depends on the agreement between the tokens produced on different parts of the retrieved collection.

Strengths And Weaknesses:
The topic discussed in the paper is very important for the field and the paper is written really clearly. However, the paper is not comparing the algorithm they propose to the other solutions even the algorithms that are using almost the same idea (e.g., https://arxiv.org/abs/2312.02132 and https://arxiv.org/abs/2407.12108).

Quality: 3: good
Clarity: 3: good
Significance: 3: good
Originality: 2: fair
Questions:

1. The main thing to address is to extend the experiments and compare with the other solutions for the problem.
2. Privacy parameters used in the experiments are not great: with all due respect, epsilon=10 is not a "reasonable budget", also delta=10^-4 is pretty big.
3. In section 2.0.1 a reference for NumericSparse is missing.
4. In addition, in your comparisons, you only compare to non-RAG solution, while it would be good to compare to non-private RAG to see the "loss".
5. Finally, the algorithm in the paper has an improvement over non-RAG solution only when there are a lot of documents in the dataset that would help answering the question. It would be good to design an experiment that would allow measuring this effect.

Limitations:
Yes.

Rating: 3: Borderline reject: Technically solid paper where reasons to reject, e.g., limited evaluation, outweigh reasons to accept, e.g., good evaluation. Please use sparingly.
Confidence: 5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.
Ethical Concerns: NO or VERY MINOR ethics concerns only
Paper Formatting Concerns:
No Concerns

Code Of Conduct Acknowledgement: Yes
Responsible Reviewing Acknowledgement: Yes

Reviewer 2

Summary:

This paper studies retrieval-augmented generation (RAG) with differential privacy, by proposing a method that spends privacy budget only for tokens that require sensitive information and uses the non-private LLM for other tokens. The authors propose an algorithm, DPVoteRAG, which prepares multiple LLM voters, feeds disjoint partitions of the sensitive corpus to them, and produces output tokens by taking the majority vote. They then design another algorithm, DPSparseVoteRAG, that spends privacy budget only when the vectors do not agree with the non-private LLM output without context. They do experiments with LLMs on two datasets and multiple models, and show improvements over the non-RAG baseline for privacy budgets of .

Strengths And Weaknesses:
Strengths

- Privacy with LLMs is an important and timely topic.
- The algorithm is quite general and can be applied to different models or datasets.
- The experiments cover multiple models and hyperparameters.

Weaknesses

- The private RAG algorithms only outperform the non-RAG baseline for a privacy budget of epsilon=10, which may not be small enough for practitioners.
- The proposed algorithms involved taking the majority votes from multiple LLM voters, which might be computationally costly to implement.
- In the experiments, the authors use the Wikipedia dataset as the external data source for retrieving documents. However, the models may be trained on it since it is a common public dataset, which may be problematic since the external dataset should be distinct from the dataset used for training, in order for the privacy guarantees to hold.

Quality: 2: fair
Clarity: 3: good
Significance: 2: fair
Originality: 2: fair
Questions:

- What is the computational overhead of implementing the private algorithm? Is there any data on the computation-utility tradeoff?
- It would significantly improve the results if there is an evaluation on a dataset where privacy is more important, rather than using the Wikipedia dataset.

Limitations:
yes

Rating: 2: Reject: For instance, a paper with technical flaws, weak evaluation, inadequate reproducibility and incompletely addressed ethical considerations.
Confidence: 4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
Ethical Concerns: NO or VERY MINOR ethics concerns only
Paper Formatting Concerns:
None.

Code Of Conduct Acknowledgement: Yes
Responsible Reviewing Acknowledgement: Yes

Reviewer 3

Summary:
This paper studies the problem of using RAG with differential privacy (DP). Long-form answers might deplete the privacy budget all too quickly, and the authors propose a method to spend the privacy budget only for a subset of tokens. The proposed method makes use of a 'sparse vector' technique to spend the privacy budget only for the tokens that require the sensitive information. This sparsity then leads to a more economical use of the privacy budget. Experimental results are reported for two datasets and three different LLM variations.

Strengths And Weaknesses:

- RAG is an important method to scale LLMs to external datasets. In most of those applications, the data is sensitive and differential privacy is a common choice to protect the privacy of the data.
- The method requires the computational overhead of running multiple LLMs as voters. (However, that computational cost can be worth it to obtain the DP guarantee.)
- Theorem 1 is ill-posed. Differential privacy is defined with respect to an adjacency relationship on two datasets. The theorem does not mention the adjacency relationship [1]. Although one can retrieve it from the proof in the appendix, I would encourage making the theorem statement more explicit. The main confusion is that the adjacency relationship seems to be with respect to documents, yet the method talks about the privacy loss of each consecutive token.

Quality: 2: fair
Clarity: 1: poor
Significance: 3: good
Originality: 2: fair
Questions:

- The proof for theorem 2 is not included. Could that please be added or provided? Moreover, I do not think that the current statement of Theorem 2 can be correct at all. Each access to the data, either for direct analysis, or for adaptivity, incurs a (small) loss of privacy. An example of this was established in Dwork and Lei (2009) [5] and a further generalization was given in Redberg et al. (2023) [6].
- On the 'Llama 3.1 (8B) on Trivia' experiment, the non-RAG solution is almost as good as the RAG solution. This is surprising, as the non-RAG solution does not see the data at all. Does this mean that any DP solution could benefit from Privacy Amplification by Subsampling? (or, alternatively, could this imply that the wikipedia dataset was part of the pre-training data and the method should be evaluated on a private RAG dataset?)
- The results do not include for epsilon_total=1.0. Although many papers exist that evaluate many values of epsilon, the value of epsilon=1.0 is still the recommended setting [2][3][4]. What is the motivation to start the parameter sweep at epsilon_total=2.0?
- Why is delta set at 1e-4? I assume that the wikipedia dataset is much larger than 10.000 entries. Whenever the delta is larger than 1 divided by the sample size, the privacy loss could be boundless for at least one datapoint, rendering the differential privacy guarantee meaningless [2].

[1] Dwork, Roth. "The algorithmic foundations of differential privacy." Foundations and Trends® in Theoretical Computer Science (2014)

[2] Blanco-Justicia et al. "A critical review on the use (and misuse) of differential privacy in machine learning." ACM Computing Surveys 55.8 (2022): 1-16.

[3] Hsu et al. "Differential privacy: An economic method for choosing epsilon." 2014 IEEE 27th Computer Security Foundations Symposium. IEEE, 2014.

[4] van Dijk, Nguyen. "Considerations on the theory of training models with differential privacy." Federated Learning. Academic Press, 2024. 29-55.

[5] Dwork and Lei. "Differential privacy and robust statistics." ACM symposium on Theory of computing, 2009.

[6] Redberg, Zhu, and Wang. "Generalized ptr: User-friendly recipes for data-adaptive algorithms with differential privacy." International Conference on Artificial Intelligence and Statistics. PMLR, 2023.

Limitations:
The proof for Theorem 2 is not provided.

Rating: 2: Reject: For instance, a paper with technical flaws, weak evaluation, inadequate reproducibility and incompletely addressed ethical considerations.
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Ethical Concerns: NO or VERY MINOR ethics concerns only
Paper Formatting Concerns:

None

Small observations that are not part of the review

From the limitations section, there are two different ways of phrasing the answer. Line 504: 'Yes we discuss the limitations and the future work in the last section.' Line 536: 'Yes, we provide the full proof in the main proof'

I would choose one of the two styles and use that throughout the paper.

Code Of Conduct Acknowledgement: Yes
Responsible Reviewing Acknowledgement: Yes

Reviewer 4

Summary:
This paper proposes differentially private retrieval-augmented generation (RAG) algorithms to enable LLMs to utilize sensitive external knowledge while preserving privacy. The authors introduce two algorithms: DPVoteRAG, which uses a sample-and-aggregate framework with multiple LLM voters operating on disjoint document partitions, and DPSparseVoteRAG, which incorporates the sparse vector technique to only consume privacy budget when sensitive information is actually needed for token generation. Experiments on Trivia and Natural Questions datasets with various LLMs show that DPSparseVoteRAG outperforms non-RAG baselines under reasonable privacy budgets ($\varepsilon \approx 10$) while generating longer, more accurate responses than DPVoteRAG.

Strengths And Weaknesses:
Strengths:

1.This work tackles the important intersection of privacy and knowledge-augmented AI systems, which is increasingly relevant as organizations seek to leverage LLMs with proprietary or sensitive data while maintaining compliance with privacy regulations.

2: The paper provides a principled solution grounded in differential privacy theory, with formal privacy guarantees and clever application of the sparse vector technique to optimize privacy budget allocation.

Weaknesses

1. The evaluation on only 100 questions per dataset due to "computational overhead" is insufficient for drawing robust conclusions. This small scale undermines the statistical significance of results and raises concerns about generalizability to real-world applications.

2: Using Wikipedia as the external corpus is problematic since it's likely included in LLM training data, making the privacy scenario artificial. Real sensitive data would have different characteristics and retrieval patterns that aren't captured in this evaluation.

3. The requirement of $\varepsilon \approx 10$ for reasonable performance is quite high for privacy-sensitive applications. Many real-world scenarios would require much stricter privacy budgets ($\varepsilon < 1$), making the practical applicability questionable.

4 The voting mechanism with 30-50 LLM instances creates significant computational and economic barriers to deployment. The paper lacks analysis of inference costs, latency impacts, and scalability considerations that would be critical for practical adoption.

Quality: 2: fair
Clarity: 3: good
Significance: 3: good
Originality: 3: good
Questions:
How does the privacy guarantee degrade when the retrieval mechanism itself leaks information? The paper assumes retrieval queries don't reveal sensitive information, but in practice, the pattern of which documents are retrieved for specific questions could leak substantial information about the corpus content. How would the authors extend their framework to handle differentially private retrieval?

What happens when the assumption of "one document per individual" is violated? Real-world scenarios often involve multiple documents per individual or documents that contain information about multiple individuals. How would the privacy analysis change, and what modifications to the algorithms would be needed to handle these more complex data ownership patterns?

Limitations:
Yes

Rating: 3: Borderline reject: Technically solid paper where reasons to reject, e.g., limited evaluation, outweigh reasons to accept, e.g., good evaluation. Please use sparingly.
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Ethical Concerns: NO or VERY MINOR ethics concerns only
Paper Formatting Concerns:
No

Code Of Conduct Acknowledgement: Yes
Responsible Reviewing Acknowledgement: Yes