# A Code for Reproducibility

See `https://anonymous.4open.science/r/checkin-B2B1/` for the code of our implementation.

# B Additional Privacy Analyses

## B.1 Distributed check-in under removal DP

As before, we assume that the server aggregates the received messages by outputting the average, $\frac{1}{B}\sum_{i=1}^{B} x_i + \mathcal{N}(0, \sigma^2/B)$. The mechanism is closely related to Poisson subsampling (Zhu & Wang, 2019).

**Theorem 5** (Distributed check-in Gaussian RDP (removal DP))**.** *Consider the mechanism in Equation 9 where agg is a mean operation on collected values. Here it is assumed that the adversary does not know the number of user checking in in each round.*

$$
\begin{aligned}
\epsilon(\lambda) \leq \frac{1}{1-\lambda} & \log\{(1-\gamma)^{\lambda-1}(\lambda\gamma - \gamma + 1) \\
& + \sum_{k=1}^{n-1} \binom{\lambda}{2}\binom{n-1}{k}\gamma^{k+2}(1-\gamma)^{n+\lambda-3-k} e^{\epsilon_k^{\mathrm{CCI}}(2)} \qquad (12) \\
& + \binom{\lambda}{2}(1-\gamma)^{n+\lambda-3}\gamma^2 e^{1/\sigma^2} + 3\sum_{l=3}^{\lambda}\binom{\lambda}{l}\gamma^{\lambda-l}(1-\gamma)^{n+l-1}e^{l(l-1)/2\sigma^2} \\
& + 3\sum_{k=1}^{n-1}\sum_{l=3}^{\lambda}\binom{n-1}{k}\binom{\lambda}{l}\gamma^{k+\lambda-l}(1-\gamma)^{n+l-1-k}e^{(l-1)\epsilon_k^{\mathrm{CCI}}(l)}\} \qquad (13)
\end{aligned}
$$

*where* $\exp((l-1)(\epsilon_k^{\mathrm{CCI}}(l)) = \sqrt{\frac{(k+1)^l}{(k+l)k^{l-1}}}\exp\left(-\frac{l+1}{\sigma^2(k+1)}\right)$ *for* $k \geq 1$*. Line 12 can be further approximated as*

$$
\binom{\lambda}{2}\gamma^2(1-\gamma)^{\lambda-2}\left(e^{\epsilon_1^{\mathrm{CCI}}(2)-\Delta^2(n-1)\gamma/2} + e^{\epsilon_{(1-\Delta)(n-1)\gamma+1}^{\mathrm{CCI}}(2)}\right) \qquad (14)
$$

*and Line 13 can be further approximated as*

$$
3\sum_{l=2}^{\lambda}\binom{\lambda}{l}(1-\gamma)^l\gamma^{\lambda-l}(e^{(l-1)\epsilon_1^{\mathrm{CCI}}(l)-\Delta^2(n-1)\gamma/2} + e^{(l-1)\epsilon_{(1-\Delta)(n-1)\gamma+1}^{\mathrm{CCI}}(l)}) \qquad (15)
$$

*where* $\Delta \in [0, 1]$.

*Proof.* The neighboring databases are $D$, $D' = D \cup \{x\}$, and the mechanisms acting on them are $\mathcal{M}, \mathcal{M}'$ respectively. Let $J$ be the index set indicating whether a user participates in training, $J = (\sigma_1, \sigma_2, \cdots, \sigma_n) \in \{0, 1\}^n$. Let $\mathbb{P}(J)$ be the probability distribution of $J$.

We denote $q(J)$ by the underlying randomization mechanism, which is a Gaussian mechanism with variance $\sigma^2/k$, $k$ being the number of non-zero elements in $J$. Following Zhu & Wang (2019), we define $q' = \sum_J q'(J), q'(J) = q(\sigma_1, \cdots, \sigma_{n-1}, 1)$, and similarly $p(J) = p'(J) = q(\sigma_1, \cdots, \sigma_{n-1}, 0)$. We have $q = (1-\gamma)p + \gamma q'$ or $p = q + \gamma p' - \gamma q'$.

we want to calculate the RDP of neighboring databases $\mathcal{M} \sim p$, $\mathcal{M}' \sim q$:

$$
\mathbb{E}_{\mathcal{M}'}\left[(\mathcal{M}/\mathcal{M}')^\lambda\right] = \mathbb{E}_q\left[\left(\frac{q + \gamma p' - \gamma q'}{q}\right)^\lambda\right]
$$

Table 1: Our approaches to bounding shuffled check-in mechanism with a generic $(\epsilon_0, \delta_0)$-LDP randomizer. Here, A, B, C, D refer to techniques used in Feldman et al. (2022), Wang et al. (2019), Balle et al. (2018), Balle et al. (2018), respectively. See text for details. The conversion from DP to RDP (using Lemma 3) can be performed in two different steps leading to different time complexities as shown in the Table. Dependence of the time complexity on other factors is suppressed.

| | Shuffling | Subsampling | Convert to RDP from | Time complexity |
|---|---|---|---|---|
| Shuff. conv. | Use A (DP) | Use B (RDP) | A | $O(n\lambda^2)$ |
| Subs. shuff. conv. | Use A (DP) | Use C (DP) | A+D | $O(n\lambda)$ |

$$\leq \mathbb{E}_{\sigma_1,\ldots,\sigma_{n-1}}\{\gamma\mathbb{E}_{q'(J)}\left(\frac{(1-\gamma)q'(J)+\gamma p'(J)}{q'(J)}\right)^{\lambda} + (1-\gamma)\mathbb{E}_{p'(J)}\left(\frac{(1+\gamma)p'(J)-\gamma q'(J)}{p'(J)}\right)^{\lambda}\}$$

$$= \sum_{k=0}^{n-1}\sum_{l=0}^{\lambda}\binom{\lambda}{l}\binom{n-1}{k}\gamma^k(\gamma)^{n-1-k}(1-\gamma)^{\lambda-l}\gamma^l\left\{\gamma\mathbb{E}_{q'(k)}\left(\frac{p'(k)}{q'(k)}\right)^l + (1-\gamma)\mathbb{E}_{p'(k)}\left(2-\frac{q'(k)}{p'(k)}\right)^l\right\}$$

Here, $q'(k) \sim \mathcal{N}(1/(k+1), \sigma^2/(k+1))$, $p'(k) \sim \mathcal{N}(0, \sigma^2/k)$ when $k \geq 1$. By direct computation $\mathbb{E}_{q'(k)}(p'(k)/q'(k))^\lambda = \sqrt{\frac{(k+1)^\lambda}{(k+\lambda)k^{\lambda-1}}}\exp\left(-\frac{\lambda+1}{\sigma^2(k+1)}\right)$. When $k=0$, $q'(k) \sim \mathcal{N}(1,\sigma^2)$, $p'(k) \sim \mathcal{N}(0,\sigma^2)$ and $\mathbb{E}_{q'(k)}(p'(k)/q'(k))^\lambda = \exp\left[(\lambda^2-\lambda)/(2\sigma^2)\right]$.

Moreover, from Lemma 4, we can make the approximation of lines 12 and 13.

We can then use the same argument in Zhu & Wang (2019) to obtain the desired result.

■

**Remark 5.** One can obtain a tighter bound if $\mathbb{E}_{q'(k)}(p'(k)/q'(k)-1)^\lambda \geq 0$ for odd $\lambda$ (Zhu & Wang, 2019). Unfortunately this does not apply to our scenario as $\mathbb{E}_{q'(k)}(p'(k)/q'(k)-1) < 0$ by direct computation.

### B.2 Shuffled check-in with a generic $(\epsilon_0, \delta_0)$-LDP randomizer

Here, we give another approach of privacy accounting of shuffled check-in with a generic $(\epsilon_0, \delta_0)$-LDP randomizer, following arguments given at the end of Section 4.1.

**Shuffling conversion.**

1. Convert shuffle DP to shuffle RDP.
2. Use shuffle RDP to evaluate subsampled shuffle RDP
3. Substitute it into Equation 5 to evaluate the composition of RDP.

To calculate the shuffle DP with $(\epsilon_0, \delta_0)$-LDP randomizer, we use Theorem 3.8 given by Feldman et al. (2022). The subsampled RDP can be calculated using Theorem 9 of Wang et al. (2019).

Note that Step 2 is a calculation of $O(\lambda)$ in terms of time complexity (Wang et al., 2019). Step 3 involves evaluating the summation with respect to $n$ as in Equation 5, which is of $O(n)$. One also needs to convert the RDP notion back to approximate DP using Lemma 2, which is an $O(\lambda)$ operation.

Table 1 summarizes the two approaches introduced in this paper. Although the subsampling shuffling conversion approach has lower time complexity, we expect the RDP bound to be looser than the shuffling conversion approach as the conversion to RDP occurs one step later.

## C Proofs

### C.1 Lemmas

The following lemma is useful for reducing the computational complexity in our proofs.

**Lemma 4.** *Let f(k) be any monotonically decreasing function with respect to k for* $k \in \mathbb{N}$, $1 \leq k \leq n$ *and* $n \in \mathbb{N}$. *Also let* $0 \leq \gamma \leq 1$. *Then, the following inequality holds:*

$$\sum_{k=1}^{n} \binom{n}{k} f(k) \leq f(1) e^{-\Delta^2 \mu/2} + f((1-\Delta)\mu + 1). \tag{16}$$

*where* $\mu = \gamma n$, $\Delta$ *is an arbitrary number conditioned on* $(1-\Delta)\mu$ *being integer and* $0 \leq \Delta \leq 1$.

*Proof.* We split the summation over $k$ to two parts: from $k = 1$ to $k$ equal or less than $(1-\Delta)n\gamma$, and those equal or larger than $(1-\Delta)n\gamma + 1$. Note that the Chernoff bound states that: $\mathbb{P}[X \leq (1-\Delta)\mu] \leq e^{-\Delta^2\mu/2}$ for all $0 < \Delta < 1$. Then,

$$\begin{aligned}
\sum_{k=1}^{n} \binom{n}{k} f(k) &= \sum_{k_1=1}^{(1-\Delta)\mu} \binom{n}{k_1} f(k_1) + \sum_{k_2=(1-\Delta)\mu+1}^{n} \binom{n}{k_2} f(k_2) \\
&\leq \sum_{k_1=1}^{(1-\Delta)\mu} \binom{n}{k_1} f(1) + \sum_{k_2=(1-\Delta)\mu+1}^{n} \binom{n}{k_2} f(k_2) \\
&\leq \mathbb{P}[k \leq (1-\Delta)\mu] f(1) + \sum_{k_2=(1-\Delta)\mu+1}^{n} \binom{n}{k_2} f(k_2) \\
&\leq f(1) e^{-\Delta^2\mu/2} + \sum_{k_2=(1-\Delta)\mu+1}^{n} \binom{n}{k_2} f(k_2) \\
&\leq f(1) e^{-\Delta^2\mu/2} + \sum_{k_2=(1-\Delta)\mu+1}^{n} \binom{n}{k_2} f((1-\Delta)\mu + 1) \\
&\leq f(1) e^{-\Delta^2\mu/2} + f((1-\Delta)\mu + 1)
\end{aligned}$$

■

The following lemma is similar to Lemma 4 but allows more flexibility by using the upper limit of summation as a “knob" to control the trade-off between computational complexity and accuracy.

**Lemma 5.** *Let f(k) be any monotonically decreasing function with respect to k for* $k \in \mathbb{N}$, $1 \leq k \leq n$ *and* $n \in \mathbb{N}$. *Also let* $0 \leq \gamma \leq 1$. *Then, the following inequality holds:*

$$\sum_{k=1}^{n} \binom{n}{k} f(k) \leq \sum_{k=1}^{(1+\Delta)\mu} \binom{n}{k} f(k) + f((1+\Delta)\mu + 1) e^{-\Delta^2\mu/(2+\Delta)}$$

*where* $\mu = \gamma n$, $\Delta$ *is an arbitrary number conditioned on* $(1+\Delta)\mu$ *being integer and* $0 \leq \Delta$.

*Proof.* By noting that the Chernoff tail-bound states that: $\mathbb{P}[X \geq (1+\Delta)\mu] \leq e^{-\Delta^2\mu/(2+\Delta)}$ for all $0 \leq \Delta$, the proof is similar to the proof of Lemma 4 and is straightforward. ■

### C.2 Proof of Theorem 2 (upper bound)

We first recite the result from Girgis et al. (2021a) for the upper bound of a subsampled shuffle mechanism.

**Lemma 6** (RDP upper bound of subsampled shuffle mechanism with discrete $\epsilon_0$-LDP randomizer (Girgis et al., 2021a))**.** *For any* $n \in \mathbb{N}$, $k \leq n$, $\epsilon_0 \geq 0$, *and any integer* $\lambda \geq 2$, *the RDP of the subsampled shuffle mechanism* $\mathcal{M}$ *is upper-bounded by*

$$\epsilon(\lambda) \leq \frac{1}{\lambda - 1} \log\left(1 + 4\binom{\lambda}{2}\gamma^2 \frac{\left(e^{\epsilon_0} - 1\right)^2}{\overline{k} e^{\epsilon_0}} + \sum_{j=3}^{\lambda} \binom{\lambda}{j} \gamma^j j \Gamma\left(j/2\right) \left(\frac{2\left(e^{2\epsilon_0} - 1\right)^2}{\overline{k} e^{2\epsilon_0}}\right)^{j/2} + \Upsilon_k\right), \tag{17}$$

*where* $\overline{k} = \lfloor \frac{k-1}{2e^{\epsilon_0}} \rfloor + 1$, $\gamma = \frac{k}{n}$, *and* $\Gamma\left(z\right) = \int_0^\infty x^{z-1}e^{-x}dx$ *is the Gamma function. The term* $\Upsilon$ *is given by* $\Upsilon_k = \left(\left(1+\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right)^\lambda - 1 - \lambda\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right) e^{-\frac{k-1}{8e^{\epsilon_0}}}$.

We make the following straightforward observation.

**Lemma 7.** *The term inside the logarithm in the RHS of Equation 17 is a monotonically decreasing function with respect to* $k$.

We finally prove the main theorem below. Assume that $\Delta$ is chosen such that $(1-\Delta)n\gamma$ is an integer. We split the summation over $k$ to two parts: from $k = 1$ [5] to $k$ equal or less than $(1-\Delta)n\gamma$, and those equal or larger than $(1-\Delta)n\gamma + 1$. We then have, from Equation 17 and Lemma 4 (replacing the term inside the logarithm in the RHS of Equation 17 with $f$, justified by Lemma 7),

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D})}{\mathcal{M}(\mathcal{D}')}\right)^\lambda\right] &\leq 1 + 4\binom{\lambda}{2}\gamma^2\left(e^{\epsilon_0}-1\right)^2\left(e^{-\epsilon_0-\Delta^2 n\gamma/2} + e^{-\epsilon_0}/\tilde{k}\right) \\
&\quad + \sum_{j=3}^{\lambda}\binom{\lambda}{j}\gamma^j j\Gamma\left(j/2\right)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{e^{2\epsilon_0}}\right)^{j/2}\left(e^{-\Delta^2 n\gamma/2} + \tilde{k}^{-j/2}\right) \\
&\quad + \Upsilon_1 e^{-\Delta^2 n\gamma/2} + \Upsilon_{(1-\Delta)n\gamma+1},
\end{aligned} \tag{18}$$

where $\tilde{k} = \lfloor \frac{(1-\Delta)n\gamma}{2\epsilon_0} \rfloor + 1$, as desired (Equation 7).

### C.3 Proof of Theorem 3 (lower bound)

We first consider binary randomized response within the (subsampled) shuffle model, which has been studied before in Cheu et al. (2019); Feldman et al. (2022); Girgis et al. (2021d); Koskela et al. (2021). Here, we give the full treatment for completeness.

First, we prove the following lemma.

**Lemma 8.** *For binary randomized response of a subsampled shuffling mechanism, the follow lower bound holds.*

$$\mathbb{E}_{q_k}\left[\left(\frac{p_k}{q_k}\right)^\lambda\right] \geq 1 + \binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}}. \tag{19}$$

*Proof.* The binary randomized response satisfying $\epsilon_0$-LDP takes input $x \in \{0,1\}$ and outputs with probability $\Pr[\mathcal{A}_{bin}(x) = x] = \frac{e^{\epsilon_0}}{e^{\epsilon_0}+1}$. Let $\rho = \frac{1}{e^{\epsilon_0}+1}$. We consider adjacent databases $D, D' \in \{0,1\}^k$ with instances $D = (0,\ldots,0)$ ,$D' = (1,0,\ldots,0)$. Let the output number of one's of the shuffler be $m$ ($m \in [k]$). The distributions of $m$ for $D$ and $D'$ are respectively,

$$\begin{aligned}
\mu_0(m) &= \binom{k}{m}\rho^m(1-\rho)^{k-m}, \\
\mu_1(m) &= (1-\rho)\binom{k-1}{m-1}\rho^{m-1}(1-\rho)^{k-m} + \rho\binom{k-1}{m}\rho^m(1-\rho)^{k-m-1}.
\end{aligned} \tag{20}$$

Hence, a subsampled shuffle mechanism with binary randomized response can be represented by

$$\begin{aligned}
q_k(m) &= \mu_0(m) \\
p_k(m) &= (1-\gamma)\mu_0(m) + \gamma\mu_1(m)
\end{aligned}$$

Next, we obtain

$$\frac{\mu_1(m)}{\mu_0(m)} = \frac{(1-\rho)\binom{k-1}{m-1}\rho^{m-1}(1-\rho)^{k-m} + \rho\binom{k-1}{m}\rho^m(1-\rho)^{k-m-1}}{\binom{k}{m}\rho^m(1-\rho)^{k-m}}$$

[5] The $k = 0$ term can be ignored as it has negligible contribution to RDP: when no one is checking in, two neighboring databases have the same output.

$$
\begin{aligned}
&= \frac{m}{k}\frac{(1-\rho)}{\rho} + \frac{k-m}{k}\frac{\rho}{1-\rho} \\
&= \frac{m}{k}e^{\epsilon_0} + \frac{k-m}{k}e^{-\epsilon_0}.
\end{aligned}
$$

Using the above result, we calculate

$$
\begin{aligned}
\frac{p_k}{q_k} - 1 &= (1-\gamma) + \gamma\frac{\mu_1(m)}{\mu_0(m)} - 1 \\
&= \gamma\left(\frac{m}{k}\left[e^{\epsilon_0} - e^{-\epsilon_0}\right] + e^{-\epsilon_0} - 1\right) \\
&= \gamma\frac{e^{2\epsilon_0}-1}{ke^{\epsilon_0}}\left(m - \frac{k(e^{\epsilon_0}-1)}{e^{2\epsilon_0}-1}\right) \\
&= \gamma\frac{e^{2\epsilon_0}-1}{ke^{\epsilon_0}}\left(m - \frac{k}{e^{\epsilon_0}+1}\right),
\end{aligned}
$$

where in the last step, we have used $e^{2\epsilon_0} - 1 = (e^{\epsilon_0}-1)(e^{\epsilon_0}+1)$.

We can now calculate the moment by binomial expansion (with respect to $\lambda$),

$$
\begin{aligned}
\mathbb{E}_{q_k}\left[\left(\frac{p_k}{q_k}\right)^{\lambda}\right] &= \mathbb{E}_{q_k}\left[\left(1 + \frac{p_k}{q_k} - 1\right)^{\lambda}\right] \\
&= 1 + \binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}} + \sum_{j=3}^{\lambda}\binom{\lambda}{j}\gamma^j\left(\frac{\left(e^{2\epsilon_0}-1\right)}{ke^{\epsilon_0}}\right)^j \mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^j
\end{aligned}
\tag{21}
$$

where the expectation is over $\text{Bin}(k,\rho)$. We have also used $\mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^2 = k\rho(1-\rho)$ in the second term. Note that $\mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^j$ is the $j$-th *central* moment of the binomial distribution.

The above equation may further be simplified using Jensen's inequality, $\mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^j \geq \left[\mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)\right]^j = 0$:

$$
\begin{aligned}
\mathbb{E}_{q_k}\left[\left(\frac{p_k}{q_k}\right)^{\lambda}\right] &= 1 + \binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}} + \sum_{j=3}^{\lambda}\binom{\lambda}{j}\gamma^j\left(\frac{\left(e^{2\epsilon_0}-1\right)}{ke^{\epsilon_0}}\right)^j \mathbb{E}\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^j \\
&\geq 1 + \binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}}.
\end{aligned}
$$

■

We finally move to proving Equation 8. We use the fact that binomial distribution is concentrated at its mean $\mu$, and with the Chernoff bound: $\mathbb{P}[X \geq (1+\Delta)\mu] \leq e^{-\Delta^2\mu/(2+\Delta)}$ for all $\Delta \geq 0$.

Note that the term depending on $k$ in Equation 19 is a monotonically decreasing function with respect to $k$. Hence, similar to Lemma 4,

$$
\begin{aligned}
\mathbb{E}_{i,k\sim\mathcal{M}(\mathcal{D}')}\left[\left(\frac{\mathcal{M}(\mathcal{D})}{\mathcal{M}(\mathcal{D}')}\right)^{\lambda}\right] &\geq \sum_{k=1}^{n}\binom{n}{k}\gamma^k(1-\gamma)^{n-k}\cdot\left(1 + \binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}}\right) \\
&= 1 + \sum_{k=1}^{n}\binom{n}{k}\gamma^k(1-\gamma)^{n-k}\cdot\binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{ke^{\epsilon_0}} \\
&\geq 1 + \left(1 - e^{-\Delta^2 n\gamma/(2+\Delta)}\right)\binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{(1+\Delta)n\gamma e^{\epsilon_0}},
\end{aligned}
$$

because $\mathbb{P}[X \leq (1+\Delta)\mu] \geq 1 - e^{-\Delta^2\mu/(2+\Delta)}$. We have henceforth obtained the desired result.

Table 2: Neural network architecture used in our experiment.

| Layer | Parameters |
|---|---|
| Convolution | 16 filters of $8 \times 8$ , strides 2 |
| Max-pooling | $2 \times 2$ |
| Convolution | 32 filters of $4 \times 4$, strides 2 |
| Max-pooling | $2 \times 2$ |
| Linear | 32 units |
| Softmax | 10 units |

## D Omitted Experimental Details and Results

**Network architecture.** The neural network we use in the experiments presented in Section 5 is as in Table 2.

**Baseline for privacy accounting of shuffled check-in with generic LDP mechanism.** We here describe in more detail our method of shuffled check-in with generic LDP mechanism without using Theorem 1.

Our shuffled check-in protocol has an issue that cannot be dealt with using standard approximate DP-based subsampling/shuffling techniques: the number of clients checking in is a variable, following a probability distribution equal to those followed by $k$ in Equation 5.

To resolve this, we devise the following method. Given $0 \leq \delta' \leq 1$, we first find $l$ such that, $\mathbb{P}[k \leq l] = 1 - \delta'$ for $k$ and the corresponding binomial probability distribution defined in Equation 5. Fixing the number of clients to be $l$, we the use the following to *conservatively* bound the approximate DP of subsampled shuffling:

- No privacy amplification occurs ($\epsilon = \epsilon_0$) due to shuffling (see Footnote 3).
- Privacy amplification by subsampling is accounted for by assuming conservatively that $l$ clients are sampled. More precisely, let $\gamma' = l/n$, privacy amplification by subsampling leads to $\epsilon_1 = \log(1 + \gamma'(e^{\epsilon_0} - 1))$ and $\delta_1 = \gamma'\delta_0$ (Balle et al., 2018).

Consequently, the approximate DP of shuffled check-in for a single round is $(\epsilon_1, \delta_1 + \delta')$ by absorbing $\delta'$ to the final DP. To compose the mechanism, we convert the approximate DP to RDP as done in Section 4.1 and use Lemma 1. [6]

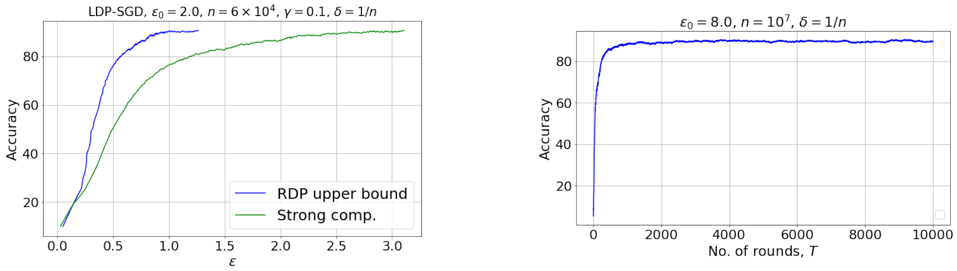
**LDP-SGD.** Our experiment utilizes the LDP-SGD algorithm (Duchi et al., 2018; Erlingsson et al., 2020) where the underlying LDP discrete randomizer satisfies $\epsilon_0$-LDP. We set the number of clients to be 60,000, each holding a data instance from the MNIST train dataset. The classification accuracy of the trained model is evaluated with the MNIST test dataset (containing 10,000 instances). The check-in rate $\gamma$ is set to be 0.1, $\epsilon_0 = 2$, and the clipping size is $C = 0.1$.

We train the network until the test accuracy reaches and stabilizes at around 90% (taking around 6,800 rounds). The accumulated budget is then calculated using our RDP upper bound (Equation 7) and Girgis et al. (2021c). As can be observed from Figure 3a, a privacy budget of $\epsilon \simeq 1$ is sufficient to train the model to reach 90% of accuracy using our RDP approach, while the strong composition-based approach requires $\epsilon \simeq 3$.

**Federated DP-SGD.** An experiment in addition to the one given in the main text is presented here. Given $\epsilon_0$, we fix $\delta_0 = 1/n^{1.5}$ to calculate the noise multiplier $z$ of the Gaussian mechanism, which adds Gaussian noise, $N(0, z^2C^2I)$ to the gradient (Abadi et al., 2016).

In Figure 3b, we show the accuracy-round curve of our Federated DP-SGD experiment with the following parameters: $\epsilon_0 = 8$, $n = 10^7$ (by bootstrapping from the train dataset), $\gamma = 10^{-4}$, clipping size $C = 0.05$. It should be noted that the average number of clients checking in is 1,000, around the same order as the ones used in the settings given in Section 5. It can then be seen that the accuracy can reach the optimal 90%, and converges faster with a larger $\epsilon_0$, noting that $n$ mainly affects the

[6] We find that composition using the strong composition theorem (Kairouz et al., 2015) leads to a much looser privacy accounting.



(a) LDP-SGD

(b) Federated DP-SGD.

Figure 3: Distributed learning under the shuffled check-in protocol.

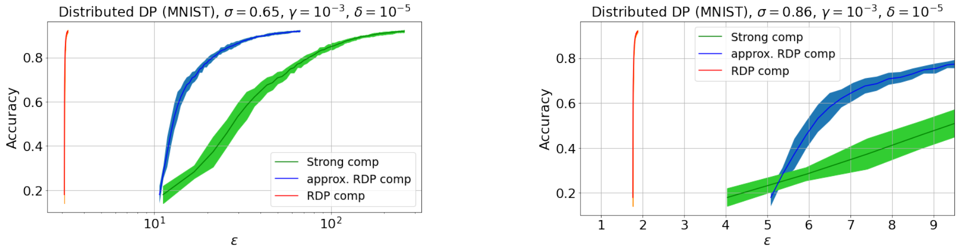


Figure 4: Distributed learning under the distributed check-in protocol. “approx. RDP comp" refers to approximation on the RDP using Lemma 4, while “RDP comp" refers to approximation on the RDP using Lemma 5.

final $\epsilon$, not the convergence. Note also that we have not evaluated the accuracy with respect to all ranges of $\epsilon$ in this experiment as it is computationally too expensive.

**Distributed DP-SGD.** Finally, we present an empirical study of the distributed check-in protocol. The setup is similar to the ones presented above, i.e., training a neural network with the MNIST dataset. We train with $\gamma = 10^{-3}$, $C = 0.04$, and two values of $\sigma$, $0.65$ and $0.86$. We show the RDP results calculated with two approximation methods, Lemma 4 (or Equation 11, setting $\Delta = 0.5$) and Lemma 5 (setting $(1 + \Delta)\mu = 200$).

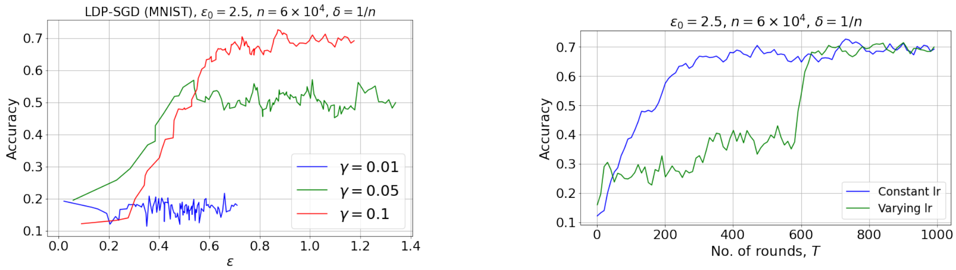
In Figure 4, we show the results (showing mean and error with experiments repeated 5 times). While the RDP approximation using Lemma 4 is tight compared to strong composition, Lemma 5 gives a much tighter result. The reason is $f(1)$ as in Equation 16 of Lemma 4 can be quite large and leads to a loose bound. Lemma 5 enables one to calculate $f(i)$ terms of small $i$'s in a more controlled manner, subsequently leading to tighter results, with a trade-off of computational complexity.

### D.1 Additional Studies

To understand the dependence on various hyperparameters and datasets, we perform additional empirical studies in this subsection, using LDP-SGD particularly.

**Varying learning rates.** Reducing learning rates in late epochs of training is known to achieve better learning performance in general. Here, we investigate this with the shuffled check-in protocol. For epoch 30 to 60, we reduce the learning rate by 50 %, and for the rest of the training, we reduce it to 10 % of the initial learning rate. See Figure 5b for the comparison where the experiments are run 3 times with different random seeds. At least within the scope of our investigation, we do not see any significant improvement in the final accuracy when varying learning rates.

**Varying check-in rate.** We study how check-in rate affects the training performance. In general, larger check-in rates are preferred for learning (corresponding to larger batch size), as seen in Figure 5a. Note that within the shuffled check-in protocol, although decreasing check-in rate enhances the privacy-amplification-by-subsampling effect, privacy-amplification-by-shuffling has an opposite effect of degrading privacy due to smaller number of shuffled/anonymized users. Overall, increasing



(a) Varying check-in rates.

(b) Varying learning rates.

Figure 5: Varying hyperparameters for LDP-SGD.

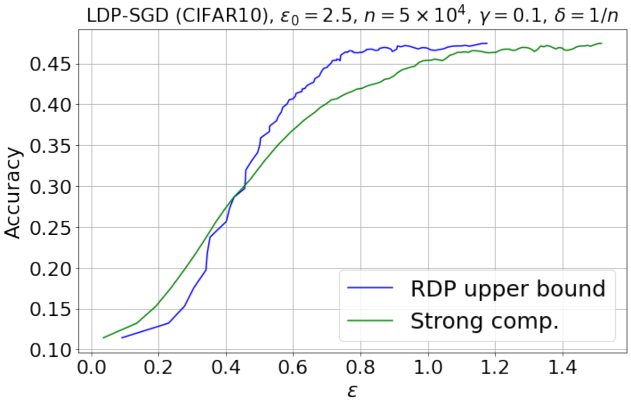


Figure 6: LDP-SGD using the CIFAR10 dataset.

batch size (increasing check-in rate) does not affect privacy accounting much. However, empirically it leads to better performance (accuracy) for LDP-SGD.

**CIFAR10.** For private distributed learning, in addition to MNIST, we experiment with CIFAR10, with the results shown in Figure 6. Here, we first pre-train a neural network with CIFAR100 without adding noise assuming that CIFAR100. Then, the final layer is fine-tuned privately with CIFAR10. Again, our RDP approach is advantageous over methods based on strong composition.