A PROOFS OF THEOREM

A.1 Proof of Theorem 4.1

Proof. Assume that $x, x' \in \mathbb{D}_\perp$ are two sensitive values. And w.l.o.g, we let $x' - x = t$. Then the probability ratio of x and x' being randomized to the same output value is

$$\begin{aligned}\frac{Pr[O=o|x]}{Pr[O=o|x']} &= \frac{e^{-|x-o|\cdot\epsilon/2}}{\sum_{j\in\mathbb{D}_\perp} e^{-|x-j|\cdot\epsilon/2}} \cdot \frac{\sum_{j\in\mathbb{D}_\perp} e^{-|x'-j|\cdot\epsilon/2}}{e^{-|x'-o|\cdot\epsilon/2}} \\ &= \frac{e^{-|x-o|\cdot\epsilon/2}}{e^{-|x+t-o|\cdot\epsilon/2}} \cdot \frac{\sum_{j\in\mathbb{D}_\perp} e^{-|x+t-j|\cdot\epsilon/2}}{\sum_{j\in\mathbb{D}_\perp} e^{-|x-j|\cdot\epsilon/2}} \\ &\le e^{t\cdot\epsilon/2}\cdot e^{t\cdot\epsilon/2} = e^{t\cdot\epsilon}.\end{aligned}$$

□

A.2 Proof of Theorem 4.2

Proof. Let $x_1, x_2 \in \mathbb{D}_\perp$ be two sensitive values, and $x_1 < x_2$, $x_2 - x_1 = t$. We calculate the probability that the output value o_2 of x_2 is greater than the output value o_1 of x_1. We have

$$\begin{aligned}Pr[o_2 > o_1] = &\sum_{o_2\in[L,R]}\sum_{o_1\in[L,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ = &\sum_{o_2\in(L,x_1]}\sum_{o_1\in[L,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ + &\sum_{o_2\in(x_1,x_2]}\sum_{o_1\in[L,x_1)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ + &\sum_{o_2\in(x_1,x_2]}\sum_{o_1\in[x_1,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ + &\sum_{o_2\in(x_2,R]}\sum_{o_1\in[L,x_1)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ + &\sum_{o_2\in(x_2,R]}\sum_{o_1\in[x_1,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2]\end{aligned}$$

For brevity, we denote $q = e^{-\epsilon/2}$, $D = ((1+q-q^{x_1-L+1}-q^{R-x_1+1})\cdot(1+q-q^{x_2-L+1}-q^{R-x_2+1}))^{-1}$.

$$\begin{aligned}&\sum_{o_2\in(L,x_1]}\sum_{o_1\in[L,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ &= \sum_{o_2\in(L,x_1]}\sum_{o_1\in[L,o_2)} \frac{q^{|o_1-x_1|}}{\sum_{v_1\in[L,R]} q^{|v_1-x_1|}}\cdot\frac{q^{|o_2-x_2|}}{\sum_{v_2\in[L,R]} q^{|v_2-x_2|}} \\ &= D\cdot(1-q)^2\cdot\sum_{o_2\in(L,x_1]}\sum_{o_1\in[L,o_2)} q^{x_2-o_2}\cdot q^{x_1-o_1} \\ &= D\cdot(q^{x_1+x_2-2L+1} - q^{x_2-L+1} + \frac{q^{x_2-x_1+1}-q^{x_1+x_2-2L+1}}{1+q}) \\ &= D\cdot(\frac{q^{x_1+x_2-2L+2}+q^{x_2-x_1+1}}{1+q} - q^{x_2-L+1})\end{aligned}$$

Similarly, we have

$$\begin{aligned}&\sum_{o_2\in(x_1,x_2]}\sum_{o_1\in[L,x_1)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ &= D\cdot(q - q^{x_1-L+1} - q^{x_2-x_1+1} + q^{x_2-L+1})\end{aligned}$$

$$\begin{aligned}&\sum_{o_2\in(x_1,x_2]}\sum_{o_1\in[x_1,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ &= D\cdot(1-((1-q)(x_2-x_1)+1)\cdot q^{x_2-x_1})\end{aligned}$$

$$\begin{aligned}&\sum_{o_2\in(x_2,R]}\sum_{o_1\in[L,x_1)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ &= D\cdot(q^2 - q^{x_1-L+2} - q^{R-x_2+2} + q^{R-x_2+x_1-L+2})\end{aligned}$$

$$\begin{aligned}&\sum_{o_2\in(x_2,R]}\sum_{o_1\in[x_1,o_2)} Pr[O=o_1|x_1]\cdot Pr[O=o_2|x_2] \\ &= D\cdot(q - q^{R-x_2+1} - \frac{q^{x_2-x_1+2}+q^{2R-x1-x2+2}}{1+q})\end{aligned}$$

By summation, we have

$$\begin{aligned}Pr[o_2 > o_1] = D\cdot((1+q)^2 + &\frac{q^{x_1+x_2-2L+2}+q^{2R-x_1-x_2+2}}{1+q} + \\ &q^{R-L-t+2} - q(q+1)(q^{x_1-L}-q^{R-x_2}))\end{aligned}$$

$Pr[o_2 > o_1]$ can be regard as a function of x_1, and $Pr[o_2 > o_1]$ minimizes at $x_1 = L$ or $x_1 = R - t$. Without loss of generality, we let $x_1 = L$

$$\begin{aligned}&Pr[o_2 > o_1] \\ &\ge \frac{(1+q)^2 + \frac{1}{1+q}(q^{t+2}+q^{2R-2L-t+2}) + q^{R-L-t+2}}{(1+q-q-q^{R-L+1})\cdot(1+q-q^{t+1}-q^{R-L-t+1})} - \\ &\qquad \frac{q(q+1)(1+q^{R-L-t}) + (\frac{2q^2+q+1}{1+q}+(1-q)t)\cdot q^t}{(1+q-q-q^{R-L+1})\cdot(1+q-q^{t+1}-q^{R-L-t+1})} \\ &= \frac{1+q+\frac{q^{2R-2L-t+2}}{1+q} - q^{R-L-t+1} - (\frac{q^2+q+1}{1+q}+(1-q)t)\cdot q^t}{(1-q^{R-L+1})\cdot(1+q-q^{t+1}-q^{R-L-t+1})} \\ &\ge \frac{1+q-q^{R-L-t+1} - (\frac{q^2+q+1}{1+q}+(1-q)t)\cdot q^t}{1+q-q^{t+1}-q^{R-L-t+1}} \\ &= 1 - \frac{\frac{1}{1+q}+(1-q)\cdot t}{1+q-q^{t+1}-q^{|D|-t}}\cdot q^t\end{aligned}$$

Thus,

$$Pr[o_2 > o_1] \ge 1 - \frac{(1-q^2)\cdot t+1}{(1+q-q^{t+1}-q^{|\mathbb{D}_\perp|-t})(1+q)}\cdot q^t$$

□

A.3 Proof of Theorem 4.3

Proof. Assume that $x_1, x_2 \in \mathbb{D}_\perp$ are two sensitive values, $o_1, o_2 \in \mathbb{D}_\perp$ are their corresponding output. And w.l.o.g, we let $x_2 - x_1 = t$. The perturbation mechanism of GRR is

$$Pr[GRR(x) = o] = \begin{cases} p_1 = \frac{e^\epsilon}{|\mathbb{D}_\perp|+e^\epsilon-1}, o = x \\ p_2 = \frac{1}{|\mathbb{D}_\perp|+e^\epsilon-1}, o \ne x \end{cases} \tag{2}$$

We have

$$Pr[o_2 > o_1|x_2 > x_1] = Pr[o_1 = x_1 \land o_2 = x_2]$$
$$+ Pr[o_1 = x_1 \land o_2 > x_1 \land o_2 \neq x_2] + Pr[o_2 = x_2 \land o_1 < x_2 \land o_1 \neq x_1]$$
$$+ Pr[o_1 < o_2 \land o_1 \neq x_1 \land o_2 \neq x_2]$$
$$= p_1^2 + p_1p_2(R - x_1 - 1) + p_1p_2 \cdot (x_2 - L - 1)$$
$$+ p_2^2 \cdot (\frac{1}{2} \cdot |\mathbb{D}_\perp|(|\mathbb{D}_\perp| - 1) - (R - x_1 - 1) - (x_2 - L - 1) - 1)$$
$$= p_1^2 + p_1p_2 \cdot (|\mathbb{D}_\perp| + t - 3) + p_2^2 \cdot (\frac{1}{2}|\mathbb{D}_\perp|(|\mathbb{D}_\perp| - 3) - t + 2)$$
$$= p_1^2 + p_1p_2 \cdot (|\mathbb{D}_\perp| - 3) + p_2^2 \cdot (\frac{1}{2}|\mathbb{D}_\perp|(|\mathbb{D}_\perp| - 3) + 2) + p_2(p_1 - p_2)t$$

□

A.4 Proof of Theorem 4.5

Proof. We assume that x and x' are two values with $|x - x'| \leq t$. The partition to which x belongs is $\mathcal{P}_i$ and x' belongs to partition $\mathcal{P}_j$, where $\mathcal{P}_i$ and $\mathcal{P}_j$ are at most $\lceil \frac{t}{\theta} \rceil$ partitions apart. Denote $M(x)$ as Algorithm 2 with input x, we first compute the probability ratio of $M(x)$ and $M(x')$ are in the same partition $\mathcal{P}_{\hat{m}}$ is

$$\frac{Pr[M(x) = \mathcal{P}_{\hat{m}}|x \in \mathcal{P}_i]}{Pr[M(x') = \mathcal{P}_{\hat{m}}|x' \in \mathcal{P}_j]}$$
$$= \frac{e^{-|i-\hat{m}|\cdot\frac{\epsilon_{prt}}{2}}}{\sum_{v\in[k]} e^{-|i-v|\cdot\frac{\epsilon_{prt}}{2}}} \cdot \frac{\sum_{v\in[k]} e^{-|j-v|\cdot\frac{\epsilon_{prt}}{2}}}{e^{-|j-\hat{m}|\cdot\frac{\epsilon_{prt}}{2}}}$$
$$= e^{\frac{\epsilon_{prt}}{2}\cdot(|j-\hat{m}|-|i-\hat{m}|)} \cdot \frac{\sum_{v\in[k]} e^{-|j-v|\cdot\frac{\epsilon_1}{2}}}{\sum_{v\in[k]} e^{-|i-v|\cdot\frac{\epsilon_1}{2}}}$$
$$\leq e^{|j-i|\cdot\frac{\epsilon_{prt}}{2}} \cdot e^{|j-i|\cdot\frac{\epsilon_{prt}}{2}} \leq e^{\lceil \frac{t}{\theta} \rceil\cdot\epsilon_{prt}}.$$

For x and x' mapped to the same partition, the maximum probability ratio of x and x' being desensitized to the same output value can be obtained when x and x' fall respectively on the left and right sides outside the partition. It satisfies

$$Pr[O = o|M(x) = \mathcal{P}_{\hat{m}}, x \in \mathcal{P}_i]$$
$$\leq e^{\theta\cdot\epsilon_{ner}} Pr[O = o|M(x') = \mathcal{P}_{\hat{m}}, x' \in \mathcal{P}_j].$$

Therefore, we have

$$Pr[O = o|x \in \mathcal{P}_i] = \sum_{\mathcal{P}_{\hat{m}} \ni o} Pr[M(x) = \mathcal{P}_{\hat{m}}|x \in \mathcal{P}_i]\cdot$$
$$Pr[O = o|M(x) = \mathcal{P}_{\hat{m}}, x \in \mathcal{P}_i]$$
$$\leq \sum_{\mathcal{P}_{\hat{m}} \ni o} e^{\lceil \frac{t}{\theta} \rceil\epsilon_{prt}} \cdot Pr[M(x') = \mathcal{P}_{\hat{m}}|x' \in \mathcal{P}_j]\cdot$$
$$e^{\theta\epsilon_{ner}} \cdot Pr[O = o|M(x') = \mathcal{P}_{\hat{m}}, x' \in \mathcal{P}_j]$$
$$= e^{\lceil \frac{t}{\theta} \rceil\epsilon_{prt}+\theta\epsilon_{ner}} Pr[O = o|x' \in \mathcal{P}_j].$$

□

A.5 Proof of Theorem 4.6

Proof. Let $x_1, x_2 \in \mathbb{D}$ be two sensitive data points. Specifically, $x_1 < x_2$, and $t = x_2 - x_1$, $T = \lfloor \frac{t}{\theta} \rfloor$. We calculate the probability that the output value o_2 of x_2 is greater than the output value o_1 of x_1. Denote $\hat{\mathcal{P}}_{m(x_1)}$ as the partition that x_1 is mapped, $\hat{\mathcal{P}}_{m(x_2)}$ as the partition that x_2 is mapped. If $\hat{\mathcal{P}}_{m(x_1)}$ and $\hat{\mathcal{P}}_{m(x_2)}$ are two different partitions and $\hat{\mathcal{P}}_{m(x_1)}$ is on the left of $\hat{\mathcal{P}}_{m(x_2)}$, then it has

$$Pr[o_2 > o_1|\hat{\mathcal{P}}_{m(x_2)} > \hat{\mathcal{P}}_{m(x_1)}] = 1$$

According to the result of Theorem 4.2, we can directly get the probability that $\hat{\mathcal{P}}_{m(x_1)}$ is on the left of $\hat{\mathcal{P}}_{m(x_2)}$ as

$$Pr[\hat{\mathcal{P}}_{m(x_2)} > \hat{\mathcal{P}}_{m(x_1)}] \geq 1 - \frac{(1-q^2)\cdot T + 1}{(1 + q - q^{T+1} - q^{k-T})(1+q)} \cdot q^T$$

The probability that x_1 and x_2 are mapped to the same partition is

$$Pr[\hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}]$$
$$= \sum_{\mathcal{P}_o \in [\mathcal{P}_1, \mathcal{P}_k]} Pr[RM(x_1) = \mathcal{P}_o] \cdot Pr[RM(x_2) = \mathcal{P}_o]$$
$$= \sum_{\mathcal{P}_o \in [[\mathcal{P}_1, \mathcal{P}_k]]} \frac{q^{|m(x_1)-o|}}{\sum_{\mathcal{P}_c \in [\mathcal{P}_1, \mathcal{P}_k]} q^{|m(x_1)-c|}} \cdot \frac{q^{|m(x_2)-o|}}{\sum_{\mathcal{P}_c \in [\mathcal{P}_1, \mathcal{P}_k]} q^{|m(x_2)-c|}}$$

By calculating the above probability summation formula, we can get

$$Pr[\hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}] \geq \frac{(1-q)^2(T+1)}{(1+q)^2} \cdot q^T$$

Since $x_2 > x_1$, when o_1 and o_2 are in the same partition, it has

$$Pr[o_2 > o_1|\hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}] > Pr[o_1 > o_2|\hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}]$$

So we can approximate the probability $Pr[o_2 > o_1, \hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}]$ as

$$Pr[o_2 > o_1, \hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}] \geq \frac{(1-q)^2(T+1)}{2(1+q)^2} \cdot q^T$$

Finally, we have

$$Pr[o_2 > o_1]$$
$$= Pr[o_2 > o_1, \hat{\mathcal{P}}_{m(x_2)} = \hat{\mathcal{P}}_{m(x_1)}] + Pr[o_2 > o_1, \hat{\mathcal{P}}_{m(x_2)} > \hat{\mathcal{P}}_{m(x_1)}]$$
$$\geq 1 - \frac{((1-q^2)\cdot T + 1)\cdot q^T}{(1 + q - q^{T+1} - q^{k-T})(1+q)} + \frac{(1-q)^2(T+1)\cdot q^T}{2(1+q)^2}$$

□

A.6 Proof of Theorem 4.11

Proof. Let v_1 and v_2 are two values with $v_1 - v_2 = t$, $[l, u] \subseteq \mathbb{D}_\perp$ is the range of output value o after adding bounded discrete Laplace noise. Denote N_1 and N_2 as the random noise sampling from bounded discrete Laplace noise $Lap_{\mathbb{Z}}(\frac{1}{\epsilon})$, where $N_1 \in [l - v_1, u - v_1]$ and $N_2 \in [l - v_2, u - v_2]$. We prove that

$$\frac{Pr[v_1 + N_1 = o]}{Pr[v_2 + N_2 = o]} = \frac{Pr[N_1 = o - v_1]}{Pr[N_2 = o - v_2]} \leq e^{2t\epsilon}$$

When $v_1 < v_2 < l < u$, it has $l - v_1 > 0$, $u - v_1 > 0$, $l - v_2 > 0$, and $u - v_2 > 0$. Then we have the ratio of $Pr[N_1 = o - v_1]$ and $Pr[N_2 = o - v_2]$ is

$$\frac{Pr[N_2 = o - v_2]}{Pr[N_1 = o - v_1]} = \frac{e^{-(l-v_1)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{e^{-(l-v_2)\epsilon}(1 - e^{-(u-l+1)\epsilon})} \cdot \frac{e^{-|o-v_2|\epsilon}}{e^{-|o-v_1|\epsilon}}$$
$$= e^{((o-v_1)-(o-v_2)\epsilon} \cdot e^{((l-v_2)-(l-v_1))\epsilon}$$
$$= e^{(v_2-v_1)\epsilon} \cdot e^{(v_1-v_2)\epsilon} = 1$$

When $l < u < v_1 < v_2$, it has $l - v_1 < 0$, $u - v_1 < 0$, $l - v_2 < 0$, and $u - v_2 < 0$. Then we have

$$\begin{aligned}\frac{Pr[N_1 = o - v_1]}{Pr[N_2 = o - v_2]} &= \frac{e^{(u-v_2)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{e^{(u-v_1)\epsilon}(1 - e^{-(u-l+1)\epsilon})} \cdot \frac{e^{-|o-v_1|\epsilon}}{e^{-|o-v_2|\epsilon}} \\ &= e^{((u-v_2)-(u-v_1))\epsilon} \cdot e^{(-(o-v_2)+(o-v_1))\epsilon} \\ &= e^{(v_1-v_2)\epsilon} \cdot e^{(v_2-v_1)\epsilon} = 1\end{aligned}$$

When $l < v_1 < v_2 < u$, it has $l - v_1 < 0$, $u - l > 0$, $l - v_2 < 0$, and $u - v_2 > 0$. Then we have

$$\begin{aligned}&\frac{Pr[N_1 = o - v_1]}{Pr[N_2 = o - v_2]} \\ &= \frac{1 - e^{-(-(l-v_2)+1)\epsilon} - e^{-(u-v_2+1)\epsilon} + e^{-\epsilon}}{1 - e^{-(-(l-v_1)+1)\epsilon} - e^{-(u-v_1+1)\epsilon} + e^{-\epsilon}} \cdot \frac{e^{-|o-v_1|\epsilon}}{e^{-|o-v_2|\epsilon}} \\ &\leq e^{t\epsilon} \cdot e^{t\epsilon} \cdot \frac{e^{-t\epsilon} - e^{-(-(l-v_2)+1+t)\epsilon} - e^{-(u-v_2+1+t)\epsilon} + e^{-(1+t)\epsilon}}{1 - e^{-(-(l-v_1)+1)\epsilon} - e^{-(u-v_1+1)\epsilon} + e^{-\epsilon}} \\ &\leq e^{2t\epsilon}\end{aligned}$$

When $v_1 < l < v_2 < u$, it has $l - v_1 > 0$, $u - v_1 > 0$, $l - v_2 < 0$, and $u - v_2 > 0$. Then we have

$$\begin{aligned}&\frac{Pr[N_2 = o - v_2]}{Pr[N_1 = o - v_1]} \\ &= \frac{e^{-|o-v_2|\epsilon}}{e^{-|o-v_1|\epsilon}} \cdot \frac{e^{-(l-v_1)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{1 - e^{-(-(l-v_2)+1)\epsilon} - e^{-(u-v_2+1)\epsilon} + e^{-\epsilon}} \\ &= e^{((o-v_1)-|o-v_2|)\epsilon} \cdot \frac{e^{-(l-v_1)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{1 - e^{((l-v_2)-1)\epsilon} - e^{-(u-v_2+1)\epsilon} + e^{-\epsilon}} \\ &\leq e^{t\epsilon} \cdot \frac{e^{-(l-v_1)\epsilon}}{e^{(l-v_2)\epsilon}} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{-(l-v_2)\epsilon} - e^{-\epsilon} - e^{(2v_2-(u+l)-1)\epsilon} + e^{(v_2-l-1)\epsilon}} \\ &= e^{t\epsilon} \cdot e^{(v_1+v_2-2l)\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{-(l-v_2)\epsilon} - e^{-\epsilon} - e^{2v_2-(u+l)-1} + e^{(v_2-l-1)\epsilon}} \\ &\leq e^{2t\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{-(l-v_2)\epsilon} - e^{-\epsilon} - e^{(2v_2-(u+l)-1)\epsilon} + e^{(v_2-l-1)\epsilon}} \\ &\leq e^{2t\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{1 - e^{-(u-l+1)\epsilon}} = e^{2t\epsilon}\end{aligned}$$

When $l < v_1 < u < v_2$, it has $l - v_1 < 0$, $u - v_1 > 0$, $l - v_2 < 0$, and $u - v_2 < 0$. Then we have

$$\begin{aligned}&\frac{Pr[N_1 = o - v_1]}{Pr[N_2 = o - v_2]} \\ &= \frac{e^{-|o-v_1|\epsilon}}{e^{-|o-v_2|\epsilon}} \cdot \frac{e^{(u-v_2)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{1 - e^{-(-(l-v_1)+1)\epsilon} - e^{-(u-v_1+1)\epsilon} + e^{-\epsilon}} \\ &= e^{((v_2-o)-|o-v_1|)\epsilon} \cdot \frac{e^{(u-v_2)\epsilon}(1 - e^{-(u-l+1)\epsilon})}{1 - e^{((l-v_1)-1)\epsilon} - e^{-(u-v_1+1)\epsilon} + e^{-\epsilon}} \\ &\leq e^{t\epsilon} \cdot \frac{e^{(u-v_2)\epsilon}}{e^{-(u-v_1)\epsilon}} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{(u-v_1)\epsilon} - e^{(u+l-2v_1-1)\epsilon} - e^{-\epsilon} + e^{(u-v_1-1)\epsilon}} \\ &= e^{t\epsilon} \cdot e^{(2u-v_1-v_2)\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{(u-v_1)\epsilon} - e^{(u+l-2v_1-1)\epsilon} - e^{-\epsilon} + e^{(u-v_1-1)\epsilon}} \\ &\leq e^{2t\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{e^{(u-v_1)\epsilon} - e^{(u+l-2v_1-1)\epsilon} - e^{-\epsilon} + e^{(u-v_1-1)\epsilon}} \\ &\leq e^{2t\epsilon} \cdot \frac{1 - e^{-(u-l+1)\epsilon}}{1 - e^{-(u-l+1)\epsilon}} = e^{2t\epsilon}\end{aligned}$$

In summary, for any output range $[l, u] \subseteq \mathbb{D}_\perp$, the ratio of $Pr[N_1 = o - v_1]$ and $Pr[N_2 = o - v_2]$ satisfies

$$\frac{Pr[v_1 + N_1 = o]}{Pr[v_2 + N_2 = o]} = \frac{Pr[N_1 = o - v_1]}{Pr[N_2 = o - v_2]} \leq e^{2t\epsilon}$$

□

A.7 Proof of Lemma 4.10

Proof. According to the definition of discrete Laplace in Definition 4.8, we have

$$\begin{aligned}&\sum_{z\in\mathbb{Z}} \frac{e^{1/\lambda} - 1}{e^{1/\lambda} + 1} \cdot e^{-|z|/\lambda} = \sum_{z\in[l,u]} \tau \cdot \frac{e^{1/\lambda} - 1}{e^{1/\lambda} + 1} \cdot e^{-|z|/\lambda} = 1 \\ &\Rightarrow \tau = \frac{\sum_{z\in\mathbb{Z}} e^{-|z|/\lambda}}{\sum_{z\in[l,u]} e^{-|z|/\lambda}}.\end{aligned}$$

Here, we have

$$\sum_{z\in\mathbb{Z}} e^{-|z|/\lambda} = \frac{2}{1 - e^{-1/\lambda}} \cdot \lim_{n\to\infty} 1 - e^{-n/\lambda} \simeq \frac{2}{1 - e^{-1/\lambda}}.$$

$$\sum_{z\in[l,u]} e^{-|z|/\lambda} = \begin{cases} \frac{e^{u/\lambda}(1-e^{-(u-l+1)/\lambda})}{1-e^{-1/\lambda}}, & l < u < 0, \\ \frac{1-e^{(-l+1)/\lambda}}{1-e^{-1/\lambda}} + \frac{e^{-1/\lambda}(1-e^{-u/\lambda})}{1-e^{-1/\lambda}}, & l < 0 < u, \\ \frac{e^{-l/\lambda}(1-e^{-(u-l+1)/\lambda})}{1-e^{-1/\lambda}}, & 0 < l < u. \end{cases}$$

Finally, we can calculate τ to get the distribution. □

B PROOFS OF COMPOSITION THEOREM OF DLDP AND PARTITION-DLDP

Theorem B.1. *Let M_i be an ϵ_i-dLDP mechanism for $i \in [k]$. Then $M_{[k]}(x) = (M_1(x), ..., M_k(x))$ satisfies $(\sum_{i=1}^k \epsilon_i) - dLDP$.*

Proof. Let x, y are two values with $|x-y| \leq t$. Fix any $(r_1, ..., r_k)$ from output domain, then we have:

$$\begin{aligned}\frac{Pr[M_{[k](x)=(r_1,...,r_k)}]}{Pr[M_{[k](y)=(r_1,...,r_k)}]} &= \frac{Pr[M_1(x) = r_1] \cdot ... \cdot Pr[M_k(x) = r_k]}{Pr[M_1(y) = r_1] \cdot ... \cdot Pr[M_k(y) = r_k]} \\ &= (\frac{Pr[M_1(x) = r_1]}{Pr[M_1(y) = r_1]}) \cdot ... \cdot (\frac{Pr[M_k(x) = r_k]}{Pr[M_k(x) = r_k]}) \\ &\leq \prod_{i=1}^k e^{t\cdot\epsilon_i} = e^{t\cdot\sum_{i=1}^k \epsilon_i}\end{aligned}$$

□

Theorem B.2. *Let M_i be an $(\epsilon_{prt}^i, \epsilon_{ner}^i)$-partition-dLDP mechanism for $i \in [k]$. Then $M_{[k]}(x) = (M_1(x), ..., M_k(x))$ satisfies $(\sum_{i=1}^k \epsilon_{prt}^i, \sum_{i=1}^k \epsilon_{ner}^i)$-partition-dLDP.*

Proof. Let x, y are two values with $|x - y| \leq t$. The partition to which x belongs is $\mathcal{P}_i$ and x' belongs to partition $\mathcal{P}_j$, where $\mathcal{P}_i$ and $\mathcal{P}_j$ are at most $\lceil \frac{t}{\theta} \rceil$ partitions apart. Fix any $(r_1, ..., r_k)$ from output domain, then we have:

$$\begin{aligned}\frac{Pr[M_{[k](x)=(r_1,...,r_k)}]}{Pr[M_{[k](y)=(r_1,...,r_k)}]} &= \frac{Pr[M_1(x) = r_1] \cdot ... \cdot Pr[M_k(x) = r_k]}{Pr[M_1(y) = r_1] \cdot ... \cdot Pr[M_k(y) = r_k]} \\ &= (\frac{Pr[M_1(x) = r_1]}{Pr[M_1(y) = r_1]}) \cdot ... \cdot (\frac{Pr[M_k(x) = r_k]}{Pr[M_k(x) = r_k]}) \\ &\leq \prod_{i=1}^k e^{\lceil \frac{t}{\theta} \rceil \epsilon_{prt}^i + \theta\epsilon_{ner}^i} = e^{\lceil \frac{t}{\theta} \rceil \sum_{i=1}^k \epsilon_{prt}^i + \theta \sum_{i=1}^k \epsilon_{ner}^i}\end{aligned}$$

□

C EXTENDED RELATED WORKS

Distance-based LDP (dLDP). The traditional DP mechanism always considers the worst case, which leads to adding excessive noise for normal cases. Kifer et al. [46] propose a semantic framework called "Pufferfish", which can generate customized privacy definitions in different scenarios. Inspired by the Pufferfish, He et al. [38] initiate a policy to specify the concept of secrets and constraints, and formally introduce the definition of dDP. Geng et al. [35] propose staircase mechanism to guarantee diverse levels of differential privacy for different instances. Nevertheless, the staircase mechanism fails to provide more sophisticated probability distribution within one partition. Because LDP is not as dependent on trusted servers as DP, LDP mechanisms are more prevalent in practical applications. *(Reviewer3:D3)* The formal dLDP definition is first proposed and applied in Location-Based Systems to guarantee location privacy within a specific distance [12, 64]. Following the intuition of d_χ-privacy in [18], Alvim et al. [11] define Metric-LDP, a variant of dLDP. Afterward, dLDP shows its broad applicability in vast scenarios [15, 19, 37, 53, 58, 63]. However, the potential capabilities of dLDP in ordinal information preserving remain undiscussed.

Privacy-Preserving Tree Boosting on Vertical FL. Traditional tree boosting algorithms have drawn public privacy concerns for their direct access to raw datasets, which leads to the emergence of privacy-preserving tree boosting. In SecureBoost proposed by Cheng et al. [21], the user holding labels send gradients and hessians encrypted with HE to other users for sorting. Fu et al. [34] proposed VF^2Boost to optimize SecureBoost from the perspective of engineering implementation. However, the training process is still extremely time-consuming since a lot of cryptographic operations are irreducible. A scheme based on Multi-Party Computation (MPC) is proposed by Abspoel et al. [9], in which only the split points of the features' values are revealed in the whole training procedure. Wu et al. [62] design another MPC-based scheme that guarantees high security for users' records. Although these schemes avoid complex cryptographic operations, the massive communication overhead caused by MPC is unbearable. To solve this problem, Tian et al. [55] propose a scheme based on Local Differential Privacy (LDP), called FederBoost. Since the randomness introduced by LDP, the accuracy of the trained model is not satisfying.

D SUPPLEMENTARY EXPERIMENT

D.1 Computation overhead compared with OPE

(Reviewer1:D2.c) To the best of our knowledge, there is no federated tree boosting scheme that uses Order-Preserving Encryption (OPE). However, it can be implemented by replacing the desensitization algorithms in our framework with an OPE scheme. In this experiment, we employ pyope 0.2.2 library for Boldyreva symmetric OPE scheme. The only difference between the two frameworks is the data desensitization algorithms used while the rest is the same, so we only need to compare the computation time of the desensitization algorithms. We randomly generate 100, 000 uniformly distributed data in the range [1, 100], process them using our algorithms and OPE respectively, and record the computation time. This experiment is conducted on a PC with Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz and 32GB memory. As shown in Table 4, our desensitization algorithms are 40-200 times faster than OPE.

ϵ Method	$\epsilon = 0.08$	$\epsilon = 1.28$
OPE	241.9227s	
GLobal-map	1.3874s	1.2764s
Local-map($\theta = 4$)	3.0140s	3.0189s
Local-map($\theta = 10$)	5.8893s	5.7287s
Adj-map($\theta = 4, \alpha = 0.4$)	4.1220s	3.0685s
Adj-map($\theta = 4, \alpha = 1$)	4.1060s	2.9861s
Adj-map($\theta = 4, \alpha = 10$)	4.0763s	3.0060s
Adj-map($\theta = 10, \alpha = 0.4$)	5.6929s	3.4936s
Adj-map($\theta = 10, \alpha = 1$)	5.5541s	3.5944s
Adj-map($\theta = 10, \alpha = 10$)	5.5308s	3.4848s

Table 4: The computaion time of desensitizing $100,000$ **values which follow uniform distribution within** $[1, 100]$**.**

D.2 Theoretical Order-Preserving Probability

We compare the theoretical order-preserving probability of proposed algorithms. Although the lower bound of order-preserving probabilities γ are formally deduced in Section 4.2, the formulas are complicated and challenging to interpret directly. We visualize both the exhaustively accumulative probability and the derived lower bound of the probability for intuitive comparison.

Order-Preserving Probability Comparisons. We traverse all the possible perturbation results that maintain the original order of a pair of values to calculate the exact order-preserving probability for comparison. To show the gap between the LDP algorithms and the distance-based LDP algorithms in order-preserving capability, we take GRR, which typically satisfies LDP definition, as the object of comparison. Since GRR satisfies the LDP definition, it's difficult to compare fairly it with our dLDP algorithms under the same ϵ. Hence we set the privacy budget of GRR as ϵ and $\theta\epsilon$ respectively.

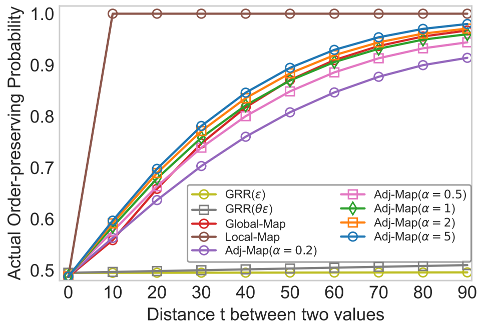
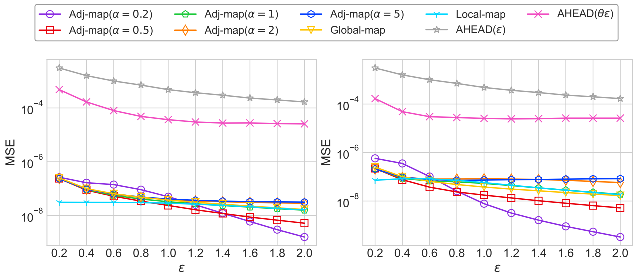


Figure 10: Comparison of actual order-preserving probability of algorithms, where $\epsilon = 0.1, \theta = 10, |\mathbb{D}_\perp| = 100$**.**

(Reviewer2:D3) As shown in Figure 10, GRR provides order-preserving probabilities γ close to 0.5 for all value pairs, which is close to randomly shuffling the order of values. The algorithms proposed in this paper provide the utility with order-preserving probabilities between 0.5 and 1. They make a trade-off between the utility and the privacy of the order-preserving encryption and LDP mechanisms. Among the algorithms proposed in this paper, Local-map gives the highest γ because the desensitized partition is

$dist$ Method	$dist = 0$ (t=5)	$dist = 1$ (t=15)	$dist = 2$ (t=25)	$dist = 3$ (t=35)	$dist = 4$ (t=45)	$dist = 5$ (t=55)	$dist = 6$ (t=65)	$dist = 7$ (t=75)	$dist = 8$ (t=85)	$dist = 9$ (t=95)
GRR(ϵ)	0.4950	0.4951	0.4953	0.4954	0.4955	0.4956	0.4957	0.4958	0.4959	0.4960
GRR($\theta\epsilon$)	0.4958	0.4975	0.4991	0.5008	0.5025	0.5041	0.5058	0.5074	0.5091	0.5108
Local-map	0.5024	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Adj-map($\alpha = 0.2$)	0.4312	0.5036	0.5829	0.6565	0.7210	0.7757	0.8213	0.8586	0.8890	0.9133
Adj-map($\alpha = 0.5$)	0.4185	0.5198	0.6206	0.7071	0.7776	0.8332	0.8763	0.9092	0.9339	0.9522
Adj-map($\alpha = 1$)	0.4133	0.5282	0.6378	0.7287	0.8004	0.8551	0.8962	0.9263	0.9482	0.9639
Adj-map($\alpha = 2$)	0.4105	0.5332	0.6476	0.7406	0.8126	0.8666	0.9062	0.9348	0.9551	0.9693
Adj-map($\alpha = 5$)	0.4087	0.5365	0.6539	0.7482	0.8201	0.8735	0.9122	0.9398	0.9591	0.9724

Table 5: The theoretical lower bound γ of order-preserving probability for any pair of data points x_1 and x_2, where $x_1 \in \mathcal{P}_i$, $x_2 \in \mathcal{P}_j$, $dist = j - i$, $t = |x_1 - x_2|$, $|\mathbb{D}_\perp| = 100$, $\theta = 10$, $\epsilon = 0.1$.



(a) $\theta = 5$

(b) $\theta = 10$

Figure 12: The MSE of Range Query on Synthetic Dataset.

deterministic. When $t \geq \theta$, which means two values fall in different partitions, Local-map guarantees that the order-preserving probability of these two values is 1. For Adj-map, α is the ratio of ϵ_{prt} and $\theta \cdot \epsilon_{ner}$. It's consistent with the analysis in section 4.2 that the smaller α is, the higher γ is. Furthermore, the lines of Global-map and Adj-map almost coincide when $\alpha = 1$, which echoes the result in Figure 3.

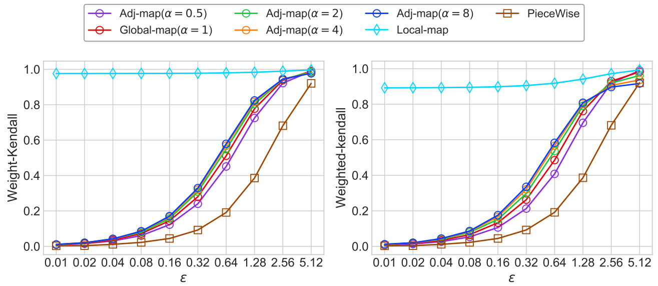
Theoretical Analysis Verification. Although we can exhaustively aggregate the accurate order-preserving probability by traversal of all possible output, it's extremely time-consuming when data domain $|\mathbb{D}_\perp|$ is large. To simplify calculations, We deduced the theoretical γ in Section 4.2 and tabulate the calculation results as Table 5. Note that γ may be less than 0.5. The reason is that in Definition 2.4, $Pr[y_i > y_j] \geq Pr[y_i < y_j] \nRightarrow Pr[y_i > y_j] \geq 0.5$ since $Pr[y_i > y_j] + Pr[y_i < y_j] < 1$ when taking $Pr[y_i = y_j]$ into consideration. Besides, there is scaling in the derivation of γ, so the result in Table 5 may be less than the accurate values. In general, the results in Table 5 are essentially consist with the accurate aggregation in Figure 10. Thus we can efficiently make approximations based on theoretical derivation without exhaustive calculation.

D.3 Weighted-Kendall on Synthetic Dataset

The results are shown in Figure 11.

D.4 The MSE of range query on Synthetic Dataset

The results are shown in Figure 12.



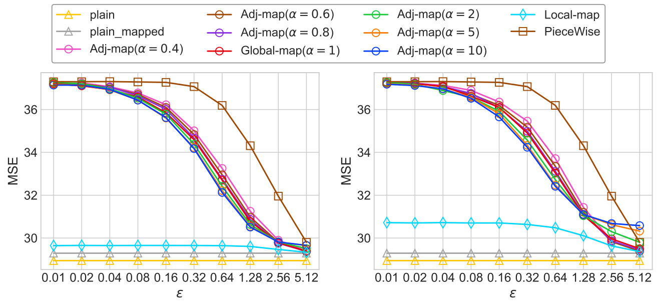
(a) $\theta = 2$

(b) $\theta = 4$

Figure 11: Weighted-Kendall on Synthetic dataset containing $10k$ uniformly distributed values.

D.5 Prediction MSE of GBDT Models for Regression Trained on CASP Dataset

The results are shown in Figure 13.



(a) $\theta = 2$

(b) $\theta = 4$

Figure 13: Prediction MSE of GBDT Models for Regression Trained on CASP Dataset.

D.6 Prediction accuracy of XGBoost models trained by OpBoost

The results are shown in Table 6.

D.7 Communication and Computation overhead of OpBoost

The results are shown in Table 7 and Table 8.

Task Method	$\epsilon = 0.08,\ \theta = 2$			$\epsilon = 0.08,\ \theta = 4$			$\epsilon = 1.28,\ \theta = 2$			$\epsilon = 1.28,\ \theta = 4$		
	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.
Local-map	0.9947×	0.9930×	0.9208×	0.9566×	0.9670×	0.8187×	1.0003×	0.9958×	0.9361×	0.9602×	0.9812×	0.8690×
Adj-map($\alpha = 0.4$)	0.5413×	0.5297×	0.6388×	0.6722×	0.4444×	0.6348×	0.6483×	0.9279×	0.8136×	0.6433×	0.9144×	0.8039×
Adj-map($\alpha = 0.6$)	0.6343×	0.5393×	0.6391×	0.7252×	0.5266×	0.6363×	0.6398×	0.9380×	0.8229×	0.6703×	0.9249×	0.8123×
Adj-map($\alpha = 0.8$)	0.6689×	0.5753×	0.6415×	0.6553×	0.4846×	0.6389×	0.6656×	0.9359×	0.8277×	0.6404×	0.9270×	0.8140×
Global-map($\alpha = 1$)	0.5935×	0.5630×	0.6394×	0.6095×	0.5513×	0.6380×	0.6797×	0.9384×	0.8283×	0.6300×	0.9309×	0.8141×
Adj-map($\alpha = 2$)	0.7311×	0.5981×	0.6401×	0.6788×	0.5507×	0.6414×	0.6735×	0.9458×	0.8321×	0.6533×	0.9324×	0.8123×
Adj-map($\alpha = 5$)	0.6751×	0.6177×	0.6435×	0.5870×	0.6106×	0.6394×	0.7188×	0.9478×	0.8338×	0.8921×	0.9400×	0.8065×
Adj-map($\alpha = 10$)	0.6817×	0.6329×	0.6449×	0.6213×	0.5815×	0.6424×	0.7733×	0.9488×	0.8350×	0.7554×	0.9372×	0.8051×
Piecewise	0.5821×	0.1187×	0.6249×	0.5821×	0.1187×	0.6249×	0.7487×	0.7847×	0.7014×	0.7487×	0.7847×	0.7014×

Table 6: Prediction accuracy of XGBoost models trained by OpBoost. We show the ratio of each accuracy to the accuracy of the model trained on the raw dataset. Three kinds of tasks are conducted on Adult, Pen-digits, and CASP datasets, respectively.

Task Method	Party A						Party B					
	GBDT			XGBoost			GBDT			XGBoost		
	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.
Local-map	74	349	96	198	768	385	521849	481377	1318036	522097	482215	1318612
Global-map($\alpha = 1$)	26	637	165	203	695	401	521753	481953	1318173	522107	482069	1318645
Adj-map($\alpha = 0.4$)	26	740	178	224	820	459	521753	482159	1318199	522148	482321	1318760
Adj-map($\alpha = 10$)	26	728	178	221	821	459	521753	482136	1318200	522143	482323	1318760
Piecewise	26	2313	253	500	12278	2318	521753	485299	1318348	522703	505220	1322469

Table 7: Total Communication (Bytes) of each Party in OpBoost by using different order-preserving desensitization algorithms with $\epsilon = 0.08$, $\theta = 4$. Three kinds of tasks are conducted on Adult, Pen-digits, and CASP datasets, respectively.

Task Method	Sampling with Bounded DLAP						Sampling with EXP					
	GBDT			XGBoost			GBDT			XGBoost		
	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.	2-Cls.	M-Cls.	Reg.
Local-map	1655.1	2926.5	2697.8	951.7	1089.7	2151.1	913.8	2284.8	826.1	218.1	424.6	325.7
Global-map($\alpha = 1$)	1154.4	2484.6	1556.5	462.3	1587.8	1010.7	904.0	2253.5	838.7	214.1	428.4	316.8
Adj-map($\alpha = 0.4$)	2187.3	3382.7	4070.2	1476.9	1587.8	3481.0	952.8	2248.6	968.2	268.4	477.7	448.2
Adj-map($\alpha = 10$)	9135.1	10052.8	21504.3	8313.3	8054.4	20584.5	953.8	2287.7	975.7	272.5	481.7	447.8
Piecewise	958.5	2163.8	953.1	305.5	650.4	495.1	–	–	–	–	–	–

Table 8: Run time (ms) of the entire training process of OpBoost by using different order-preserving desensitization algorithms with $\epsilon = 0.08$, $\theta = 4$. Three kinds of tasks are conducted on Adult, Pen-digits, and CASP datasets, respectively.