Concurrent Composition Theorems for Differential Privacy

Salil Vadhan
Harvard University
Cambridge, MA, USA
salil_vadhan@harvard.edu

Wanrong Zhang
Harvard University
Cambridge, MA, USA
wanrongzhang@fas.harvard.edu

ABSTRACT
We study the concurrent composition properties of interactive differentially private mechanisms, whereby an adversary can arbitrarily interleave its queries to the different mechanisms. We prove that all composition theorems for non-interactive differentially private mechanisms extend to the concurrent composition of interactive differentially private mechanisms, whenever differential privacy is measured using the hypothesis testing framework of $f$-DP, which captures standard $(\epsilon, \delta)$-DP as a special case. We prove the concurrent composition theorem by showing that every interactive $f$-DP mechanism can be simulated by interactive post-processing of a non-interactive $f$-DP mechanism.

In concurrent and independent work, Lyu (NeurIPS ’22) proves a similar result to ours for $(\epsilon, \delta)$-DP, as well as a concurrent composition theorem for Rényi DP. We also provide a simple proof of Lyu’s concurrent composition theorem for Rényi DP. Lyu leaves the general case of $f$-DP as an open problem, which we solve in this paper.

CCS CONCEPTS
• Theory of computation → Interactive computation; Concurrent algorithms; Security and privacy → Privacy-preserving protocols; Privacy protections.

KEYWORDS
Differential Privacy, Interactive Mechanisms, Concurrent Composition

ACM Reference Format:
Salil Vadhan and Wanrong Zhang. 2023. Concurrent Composition Theorems for Differential Privacy. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC ’23), June 20–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3564246.3585241

1 INTRODUCTION
1.1 Differential Privacy
Differential privacy is a statistical notion of database privacy, which ensures that the output of an algorithm will still have approximately the same distribution if a single data entry were to be changed. Differential privacy can be defined in terms of a general database space $X$, and a binary neighboring relation on $X$, which we think of as capturing whether “two datasets” differ on one individual’s data. For example, if databases are real-valued and contain a fixed number $n$ of entries, then $X = \mathbb{R}^n$, and two datasets $x, x' \in \mathbb{R}^n$ are said to be neighboring if they differ in at most one coordinate.

Definition 1.1 (Differential Privacy [5]). A randomized algorithm $M : X \rightarrow R$ is $(\epsilon, \delta)$-differentially private if for every pair of neighboring datasets $x, x' \in X$, and for every subset of possible outputs $S \subseteq R$, $$\Pr[M(x) \in S] \leq \exp(\epsilon) \cdot \Pr[M(x') \in S] + \delta.$$ Thus, differential privacy requires that for all neighboring datasets $x, x'$, $M(x)$ and $M(x')$ are close as probability distributions (as measured by the parameters $\epsilon$ and $\delta$). A number of variants of differential privacy have been defined based on other ways of measuring closeness, leading to Concentrated differential privacy (CDP) [2, 9] and Rényi differential privacy (RDP) [14] and $f$-differential privacy ($f$-DP) [3].

1.2 Interactive Differential Privacy
Definition 1.1 considers only non-interactive mechanisms $M$ that release query answers in one shot, but data analysts often interact with a database in an adaptive fashion. In fact, many useful primitives in differential privacy such as the Sparse Vector Technique [6–8], and the Private Multiplicative Weights [11] allow analysts to ask an adaptive sequence of queries about a dataset. It motivates the study of interactive mechanisms to capture full-featured privacy-preserving data analytics. Here, we view the mechanism $M$ as a party in an interactive protocol, interacting with a (possibly adversarial) analyst.

Definition 1.2 (Interactive protocols). An interactive protocol $\langle A, B \rangle$ is any pair of functions on tuples of binary strings. The interaction between $A$ with input $x_A$ and $B$ with input $x_B$ is the following random process (denoted $(A(x_A), B(x_B))$):

1. Uniformly choose random coins $r_A$ and $r_B$ for $A$ and $B$, respectively.
2. Repeat the following for $i = 0, 1, \ldots$:
   a. If $i$ is even, let $m_i = A(x_A, m_1, m_2, \ldots, m_{i-1}; r_A)$.
   b. If $i$ is odd, let $m_i = B(x_B, m_0, m_2, \ldots, m_{i-1}; r_B)$.
   c. If $m_i = \text{halt}$, then exit loop.

The view of a party in an interactive protocol captures everything the party “sees” during the execution.

Definition 1.3 (View of a party in an interactive protocol). Let $\langle A, B \rangle$ be an interactive protocol. Let $r_A$ and $r_B$ be the random coins for $A$ and $B$, respectively. A’s view of $(A(x_A; r_A), B(x_B; r_B))$ is the tuple $\text{View}_A(A(x_A; r_A) \leftarrow B(x_B; r_B)) = (r_A, x_A, m_1, m_2, \ldots)$ consisting of all the messages received by $A$ in the execution of the protocol.
together with the private input $x_A$ and random coins $r_A$. $B$’s view of $(A(x_A; r_A), B(x_B; r_B))$ is defined symmetrically.

In the setting of differentially private mechanisms, Party $A$ is the mechanism, where the input $x_A$ is the dataset. party $B$ is the adversary that does not have an input $x_B$. Since we only care about the view of the adversary, we will drop the subscript and denote the view of the adversary as $\text{View}(B \leftrightarrow M(x))$. With this notation, interactive differential privacy is defined by asking for the views of an adversary on any pair of neighboring datasets $\text{View}(B \leftrightarrow M(x))$ and $\text{View}(B \leftrightarrow M(x'))$ satisfying the same $(\epsilon, \delta)$-closeness notion as in non-interactive differential privacy.

**Definition 1.4.** A randomized algorithm $M$ is an $(\epsilon, \delta)$-differentially private interactive mechanism if for every pair of neighboring datasets $X, X' \in X$, every adversary algorithm $B \in \mathcal{B}$, and every subset of possible views $S \subseteq \text{Range(View)}$, we have

$$\Pr[\text{View}(B \leftrightarrow M(x)) \in S] \leq \exp(\epsilon) \Pr[\text{View}(B \leftrightarrow M(x')) \in S] + \delta.$$

### 1.3 Concurrent Composition

A fundamental problem in differential privacy is studying how the privacy degrades under composition as more computations are performed on the same database. The composition property is particularly useful when we want to ask interactive queries on the same database, and it also allows us to design a complex differentially private algorithm by combining several building blocks. Formally, we define the composition of a sequence of non-interactive $k$ mechanisms $M_1, M_2, \ldots, M_k$ as the non-interactive mechanism $M = \text{Comp}(M_1, M_2, \ldots, M_k)$ defined as

$$M(x) := (M_1(x), M_2(x), \ldots, M_k(x)), \quad (1)$$

where each mechanism $M$ is executed using independent random coins.

The composition of non-interactive mechanisms has been studied extensively in the literature. The basic composition theorem [4] states that the privacy parameters add up linearly when composing private mechanisms. The advanced composition theorem [10] provides a tighter bound where the privacy parameter grows sublinearly under $k$-fold adaptive composition. Later, the optimal composition theorem [12, 15] gives an exact characterization of the privacy guarantee under $k$-fold composition. The relaxations of differential privacy including zero-concentrated differential privacy (zCDP) [2, 9], Rényi differential privacy (RDP) [14], and $f$-differential privacy ($f$-DP) [3] allows for tighter reasoning about composition. In the abovementioned work, some of them [10, 15] are framed in a way that the adversary can adaptively choose the mechanisms $M_1, M_2, \ldots, M_k$, and thus the adaptive composition can be viewed as an interactive mechanism.

**Definition 1.5 (Concurrent composition of interactive mechanisms[17]).** Let $M_1, \ldots, M_k$ be interactive mechanisms. $M = \text{ConComp}(M_1, \ldots, M_k)$ is the concurrent composition of mechanisms $M_1, \ldots, M_k$ defined as follows:

1. Random sample $r = (r_1, \ldots, r_k)$ where $r_j$ are random coin tosses for $M_j$.
2. Inputs for $M$ consists of $x = (x_1, \ldots, x_k)$ where $x_j$ is a private dataset for $M_j$.
3. $M(x, m_0, \ldots, m_{j-1}; r)$ is defined as follows:

![Figure 1: Concurrent composition of interactive mechanisms](image-url)
(a) Parse $m_{i-1}$ as $(j, q)$ where $j = 1, \ldots, k$ and $q$ is a query to $M_i$. If $m_{i-1}$ cannot be parsed correctly, output halt.

(b) Extract history $(m'_0, \ldots, m'_{r-1})$ from $(m_0, \ldots, m_{i-1})$ where $m'_i$ are all of the queries to mechanism $M_i$.

(c) Output $M_i(x_j, m'_0, \ldots, m'_{r-1}, r_j)$.

For an adversary $B$, we will use the notation $\text{View}(B \leftrightarrow (M_1, \ldots, M_k))$ as shorthand for $\text{View}(B \leftrightarrow \text{Comp}(M_1, \ldots, M_k))$.

Vadhan and Wang [17] showed that the advanced and optimal composition theorems extend to the concurrent composition of interactive pure DP mechanisms.

**Theorem 1.1** ([17]). Suppose that all non-interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $(\epsilon_i, \delta_i)$-differentially private for $\delta_1 = \delta_2 = \ldots = \delta_k = 0$, their composition $\text{Comp}(M_1, \ldots, M_k)$ is $(\epsilon, \delta)$-differentially private. Then for all interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $(\epsilon_i, \delta_i)$-differentially private for $\delta_1 = \delta_2 = \ldots = \delta_k = 0$, the concurrent composition $\text{ConComp}(M_1, \ldots, M_k)$ of interactive mechanisms $M_1, \ldots, M_k$ is $(\epsilon, \delta)$-differentially private.

They prove this by reducing the analysis of interactive pure DP mechanism to that of analyzing the Randomized Response mechanism [5, 20].

**Theorem 1.2** ([17]). Suppose that $M$ is an interactive $(\epsilon, 0)$-differentially private mechanism. Then for every pair of neighboring datasets $x, x'$, there exists an interactive post-processing function $\mathcal{P}$ such that for every adversary $B \in \mathcal{B}$, we have

\[
\text{View}(B \leftrightarrow M(x)) \equiv \text{View}(B \leftrightarrow \mathcal{P}(\text{RR}_x(0)))
\]

\[
\text{View}(B \leftrightarrow M(x')) \equiv \text{View}(B \leftrightarrow \mathcal{P}(\text{RR}_x(1))).
\]

Here $\mathcal{P}$ is an interactive post-processing function that depends on $M$ and a fixed pair of neighboring datasets $x, x'$. It receives a single bit as an output of $\text{RR}_x(0)$ or $\text{RR}_x(1)$, and then interacts with the adversary $B$.

Note that Theorem 1.1 and Theorem 1.2 do not apply to the case where the composed mechanisms $M_i$ are $(\epsilon_i, \delta_i)$-DP for $\delta_i > 0$. In this case, [17] only show a bound that is similar to the “group privacy” property of $(\epsilon, \delta)$-DP. In particular, if $\epsilon_1 = \epsilon_2 = \ldots = \epsilon_k = \epsilon$ and $\delta_1 = \delta_2 = \ldots = \delta_k = \delta$, they show that the concurrent composition $\text{ConComp}(M_1, M_2, \ldots, M_k)$ is $(\epsilon, \exp(\epsilon) \cdot (\exp(\epsilon) - 1) \cdot \delta)$-differentially private. This is suboptimal even compared to basic composition. It left as an open problem that if any composition theorems for non-interactive mechanisms can extend to all variants of DP interactive mechanisms.

**Open Question.** Does Theorem 1.1 extend to other variants of DP (such as $(\epsilon_i, \delta_i)$-DP with $\delta_i > 0$, Rényi DP, $f$-DP)?

### 1.4 Our Results on Concurrent Composition

In this paper, we close this gap and show that any composition theorems of non-interactive mechanisms also extend to the concurrent composition of interactive DP mechanisms for approximate DP. In particular, we show that Theorem 1.1 extends to the case that $\delta_1 > 0$.

**Theorem 1.3** (Concurrent Composition for $(\epsilon, \delta)$-DP Interactive Mechanisms). Suppose that for all non-interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $(\epsilon_i, \delta_i)$-differentially private for $i = 1, 2, \ldots, k$, their composition $\text{Comp}(M_1, \ldots, M_k)$ is $(\epsilon, \delta)$-differentially private. Then for all interactive mechanisms $M_1, \ldots, M_k$ with finite communication such that $M_i$ is $(\epsilon_i, \delta_i)$-differentially private for $i = 1, 2, \ldots, k$, the concurrent composition $\text{ConComp}(M_1, \ldots, M_k)$ of interactive mechanisms $M_1, \ldots, M_k$ is $(\epsilon, \delta)$-differentially private.

We also handle general $f$-DP as defined and discussed in the section below.

**Theorem 1.4** (Concurrent Composition for $f$-DP Interactive Mechanisms). Suppose that all non-interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $f_i$-DP for $i = 1, 2, \ldots, k$, their composition $\text{Comp}(M_1, \ldots, M_k)$ is $f$-DP. Then for all interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $f_i$-DP for $i = 1, 2, \ldots, k$, the concurrent composition $\text{ConComp}(M_1, \ldots, M_k)$ of interactive mechanisms $M_1, \ldots, M_k$ is $f$-DP.

Theorem 1.3 follows directly from Theorem 1.4 because $f$-DP defined below captures $(\epsilon, \delta)$-DP as a special case [3, 21]. Interestingly, the generalization to $f$-DP is important for our proof, even if we only want to prove Theorem 1.3. We explain the detailed proof technique in the section below.

In summary, our results show that there is no extra privacy loss due to the concurrent access to multiple interactive mechanisms. We can now safely run multiple interactive differentially private algorithms in parallel, while allowing communication with all them during their executions.

### 1.5 $f$-DP and Interactive vs. Noninteractive Hypothesis Testing

$f$-differential privacy ($f$-DP) [3] is a generalization of $(\epsilon, \delta)$-differential privacy based on the hypothesis testing interpretation of differential privacy. Differential privacy attempts to measure the difficulty of distinguishing two neighboring datasets based on the output of a mechanism. Specifically, an adversary considers the following hypothesis testing problem:

$$H_0 : \text{the dataset is } x \quad \text{versus} \quad H_1 : \text{the dataset is } x'.\$$

Denote by $Y$ and $Y'$ the output distributions of $M$ on the two neighboring datasets, namely $M(x)$ and $M(x')$. For a given rejection rule $\phi$, the type I error $\alpha_\phi = E[\phi(Y)]$ is the probability of rejecting $H_0$ when $H_0$ is true, while the type II error $\beta_\phi = 1 - E[\phi(Y')]$ is the probability of failing to reject $H_0$ when $H_1$ is true. A trade-off function serves as the optimal boundary of the achievable and unachievable regions of these errors.

**Definition 1.6** (Trade-off function [3]). For any two probability distributions $Y$ and $Y'$ on the same space, define the trade-off function $T(Y, Y') : [0, 1] \rightarrow [0, 1]$ as

$$T(Y, Y')(\epsilon) = \inf \{ \beta_\phi : \alpha_\phi \leq \epsilon \}, \quad (2)$$

where the infimum is taken over all (measurable) rejection rules $\phi$.

Proposition 1.7 gives the necessary and sufficient condition for $f$ to be a trade-off function.

**Proposition 1.7** (Class of trade-off functions [3]). A function $f : [0, 1] \rightarrow [0, 1]$ is a trade-off function if and only if $f$ is convex, continuous, non-increasing, and $f(x) \leq 1 - x$ for $x \in [0, 1]$. 
f-DP allows the full trade-off between type I and type II errors in the simple hypothesis testing problem to be governed by a trade-off function f. A larger trade-off functions implies stronger privacy guarantees.

**Definition 1.8** (f-differential privacy [3]). Let f be a trade-off function. A mechanism M : X → R is f-differentially private if for every pair of neighboring datasets x, x′ ∈ X, we have

\[ T(M(x), M(x')) ≥ f. \]

(ε, δ)-DP is a special case of f-DP, taking f = fε,δ, where fε,δ = max{0, 1 − δ − exp(ε)a, exp(−ε)(1 − δ − a)} [3, 21].

To prove Theorem 1.4 (and hence Theorem 1.3), we prove the following analogue of Theorem 1.2, showing that every interactive f-DP mechanism can be simulated by an interactive post-processing of a non-interactive mechanism.

**Theorem 1.5.** For every trade-off function f, every interactive f-DP mechanism M with finite communication, and every pair of neighboring datasets x, x′, there exists a non-interactive f-DP mechanism N and a randomized interactive post-processing mechanism P such that for every adversary B ∈ B, we have

\[
\text{View}(B ↔ M(x)) ≡ \text{View}(B ↔ P(N(x)))
\]

Similarly, to Theorem 1.5, in the case of (ε, δ)-DP, one can take the non-interactive mechanism N as the (ε, δ)-Randomized Response mechanism of [12]. Indeed, [12] shows that every non-interactive (ε, δ)-DP mechanism can be simulated as a post-processing of (ε, δ)-Randomized Response.

**Theorem 1.4** follows from **Theorem 1.5** in the same way as **Theorem 1.1** follows from **Theorem 1.2**. Indeed, **Theorem 1.4** implies that to analyze the concurrent composition of interactive mechanisms Mᵢ, it suffices to consider the composition of the non-interactive mechanisms Nᵢ. As a result, composition theorems for non-interactive mechanisms extend to the concurrent composition of interactive f-DP mechanisms.

**Theorem 1.5** is an interesting statement about statistical hypothesis testing even without the application to differential privacy. Normally, hypothesis testing is presented as the task of distinguishing between two distributions or sets of distributions. This is a non-interactive task: a sample from the distribution is generated and given to the hypothesis tester, which then tries to decide whether the distribution is in H₀ or H₁. However, suppose instead we consider the task of distinguishing between two interactive mechanisms M₀ and M₁, each of which responds to queries in a randomized and stateful manner. Since the mechanisms are stateful, the hypothesis tester may never learn everything there is to know about the mechanism; in particular it cannot find out how the mechanism would have answered if different queries had been asked in the past. This is in contrast to ordinary hypothesis testing, where the full sample from the distribution is given to the hypothesis tester. Nevertheless, by viewing M₀ as M(x) and M₁ as M(x'), **Theorem 1.5** implies that the two interactive mechanisms M₀ and M₁ can be simulated perfectly by noninteractive random variables N₀ = N(x) and N₁ = N(x') such that even if we give N₀ or N₁ to a hypothesis tester in its entirety (thereby revealing how M₀ or M₁ would answer all questions), it cannot distinguish them any better than it could distinguish M₀ and M₁. The trick, of course, is that the simulation is “perfect” only when executing a single interaction with M₀ or M₁ (with no rewinding to explore multiple paths in the interaction tree).

The proof of **Theorem 1.5** relies on the following two lemmas.

**Lemma 1.6** (Coupling property of f-DP). Let f be a trade-off function, and suppose we have random variables X, Y and X', Y' such that

\[ T(X, X') ≥ f \text{ and } T(Y, Y') ≥ f. \]

Then there exists couplings (X, Y) and (X', Y') such that

\[ T((X, Y)|| (X', Y')) ≥ f. \]

A coupling of random variables X and Y is any random vector (X, Y) such that the marginal distributions are identically distributed to X and Y respectively, i.e., X ≡ X and Y ≡ Y. Subject to this constraint on the marginal, X and Y can be arbitrarily correlated. Allowing correlations is critical to **Lemma 1.6**. For example, for the case of (ε, δ)-DP, if we keep X, Y and X', Y' independent, then we would just get the “group privacy” like bound.

**Lemma 1.7** (Chain rule of f-DP). For every pair of random variables X, X' with finite support, there exists a function ChainRuleₓₓ' such that for every random variable Y jointly distributed with X, and every random variable Y' jointly distributed with X', we have

\[ T((X, Y)|| (X', Y')) \]

= ChainRuleₓₓ'((T(Y|X = x, Y'|x' = x)) | x ∈ supp(X) ∩ supp(X')).

Moreover, ChainRule is a function that is “continuous in each variable” on the partially ordered set of trade-off functions (see Section 2 for formal definition).

**Lemma 1.7** says that the trade-off function between (X, Y) and (X', Y') can be determined by a collection of trade-off functions between Y and Y' conditioned on X = X' = x for every x ∈ supp(X) ∩ supp(X') through a ChainRule function. The terminology “chain rule” is by analogy with the standard chain rule for KL divergence, which says

\[ \text{KL}(X, Y)|| (X', Y') = \text{KL}(X|| X') + E_{x,X'} \text{KL}(Y|X = x|| Y'|X' = x). \]

(3)

So fixing X and X', we can calculate the KL divergence for arbitrary Y and Y' as a function of the KL divergences KL(Y|X = x|| Y'|X' = x). (ε, δ)-DP does not admit the chain rule property, because no pairs of (ε, δ) can exactly capture the “closeness” of (X, Y) and (X', Y') given a collection of {εj, δj} that characterize the “closeness” of |X = x and Y'|X' = x'. Working with the general f-DP allows us to capture a complete characterization of “privacy”.

To prove **Theorem 1.5** using **Lemmas 1.6** and 1.7, our strategy is to apply induction on the number of messages exchanged (which we can do since M has finite communication by assumptions). To reduce k rounds of interactions to k − 1 rounds, we consider the subsequent interaction conditioned on the first message. Depending on whether the first message sent from the mechanism M or the adversary B, we consider the following two cases.

**Case 1.** The adversary B sends the first query q₁ to the mechanism M. Fix a pair of neighboring datasets x, x'. Fixing q₁, we denote the subsequent interactive mechanism by Mₜq₁. By induction,
We use the chain rule to argue that the DP mechanism can be simulated by interactive post-processing of the pairs $(X_i, X'_i)$ for all $i \in I$, and a random variable $I$ distributed on $I$. If $D(X_i, X'_i) \leq d$ for all $i \in I$, then $D(X, X') \leq d$.

For the generalized notion $d$-DP, the difficulty of distinguishing two neighboring datasets is measured by the generalized distance between the distributions of an adversary’s views. The partially ordered set allows us to compare the level of privacy guarantees of mechanisms.

Definition 2.2 ($d$-DP). Let $(D, \leq, D)$ be a generalized probability distance. For $d \in D$, we call an interactive mechanism $M$ $d$-DP if for every adversary $B \in B$ and every pair of neighboring datasets $x, x'$, we have

$$D(\text{View}(B \leftrightarrow M(x)), \text{View}(B \leftrightarrow M(x'))) \leq d.$$ 

Let us instantiate the standard pure DP and its variants using the definition above by specifying the generalized distances.

**Example: pure DP.** For pure DP, a smaller $\varepsilon$ provides stronger privacy guarantee, so the partially ordered set $D$ is defined as $(\mathbb{R}^{\geq 0} \cup \{\infty\}, \leq)$. The distance mapping is the max-divergence $D_{\infty}$. For two probability distributions $P$ and $Q$, the max-divergence is

$$D_{\infty}(P||Q) := \sup_{T \subseteq \text{supp}(P)} \log \left( \frac{\Pr(P(x) \in T)}{\Pr(Q(x) \in T)} \right).$$

Max-divergence is closed under post-processing due to the data-processing inequality. Max-divergence satisfies joint convexity due to the following lemma.

**Lemma 2.1 ([19]).** For every two pairs of probability distributions $(P_0, Q_0)$ and $(P_1, Q_1)$, and every $\lambda \in (0, 1)$,

$$D_{\infty}((1 - \lambda)P_0 + \lambda P_1||((1 - \lambda)Q_0 + Q_1) \leq \max\{D_{\infty}(P_0||Q_0), D_{\infty}(P_1||Q_1)\}.$$ 

**Example: Rényi DP.** For Rényi DP of order $\alpha$, the partially ordered set $D$ is also $(\mathbb{R}^{\geq 0} \cup \{\infty\}, \leq)$. The distance mapping is $\alpha$-Rényi divergence for $\alpha \in (1, \infty)$. The Rényi divergence is defined as follows.

Definition 2.3 (Rényi divergence [16]). For two probability distributions $P$ and $Q$, the Rényi divergence of order $\alpha > 1$ is

$$D_{\alpha}(P||Q) = \frac{1}{\alpha - 1} \log \left( \mathbb{E}_{X \sim Q} \left[ \left( \frac{P(x)}{Q(x)} \right)^{\alpha} \right] \right).$$

Rényi divergence is also closed under post-processing due to the data-processing inequality, and it satisfies the joint convexity because an analogue of Lemma 2.1 also holds for Rényi divergence.

**Lemma 2.2 ([19]).** For every order $\alpha > 1$, every two pairs of probability distributions $(P_0, Q_0)$ and $P_1, Q_1$, and every $\lambda \in (0, 1)$,

$$D_{\alpha}((1 - \lambda)P_0 + \lambda P_1||((1 - \lambda)Q_0 + Q_1) \leq \max\{D_{\alpha}(P_0||Q_0), D_{\alpha}(P_1||Q_1)\}.$$
Therefore, we have

Definition 2.4

Lemma 2.3.

Example: $f$-DP. For $f$-DP, the partially ordered set $\mathcal{D}$ is defined as $(\mathcal{F}, \preceq)$, where $\mathcal{F}$ is the set of all trade-off functions that satisfies the conditions in Proposition 1.7. The partial ordering is defined as $f_1 \preceq f_2$ if $f_1(\alpha) \geq f_2(\alpha)$ holds for all $\alpha \in [0, 1]$. Note that the direction of the inequalities is reversed, corresponding to the fact that a larger trade-off function means less privacy loss. The distance mapping is the trade-off function $T$ in Definition 1.6.

$f$-DP also satisfies the two properties. First, $f$-DP is preserved under post-processing. We will only need to show the joint convexity of $f$-DP.

**Lemma 2.3.** Suppose we have a collection of random variables $(X_i, X'_i)_{i \in I}$ and a random variable $I$ distributed on $\mathcal{I}$. If $T(X_i, X'_i) \geq f$ for all $i \in I$, then $T(X_I, X'_I) \geq f$.

**Proof.** For any random variable $I$ distributed on $\mathcal{I}$, we have

$$T(X_I, X'_I)(\alpha) = \inf_{\phi} \left\{ E[1 - \phi(X'_I)] : E[\phi(X_I)] \leq \alpha \right\}$$

$$= \inf_{\phi} \left\{ E_{i \in I} [1 - \phi(X'_i)] : E_{i \in I} [\phi(X_i)] \leq \alpha \right\}$$

$$\geq \inf_{\phi} \left\{ E_{i \in I} [f(E[\phi(X_i)])] : E_{i \in I} [\phi(X_i)] \leq \alpha \right\}$$

($f$ non-decreasing)

$$\geq \inf_{\phi} \left\{ f(E_{i \in I} [\phi(X_i)]) : E_{i \in I} [\phi(X_i)] \leq \alpha \right\}$$

($f$ convex)

$$= f(\alpha) .$$

Therefore, we have $T(X_I, X'_I) \geq f$.

It is useful to work with distance posets that are complete.

**Definition 2.4 (Complete poset).** A partially ordered set (poset) $(\mathcal{D}, \preceq)$ is complete if for every nonempty subset $S \subseteq \mathcal{D}$ has a supremum $\sup(S)$, where $s \preceq \sup(S)$ for every $s \in S$, and $\sup(S) \preceq t$ for every $t$ satisfying $s \preceq t$ for every $s \in S$.

We note that $\sup(S)$ is always unique. The poset $((\mathbb{R}^{\geq 0}) \cup \{\infty\}, \preceq)$ used in pure DP and Rényi DP is complete by the usual completeness of the real numbers. For the poset $(\mathcal{F}, \preceq)$ used in $f$-DP, we prove it below. Note that if $(\mathcal{D}, \preceq)$ is complete then in Definition 2.2 we can take $d = \sup_{\mathcal{D}} D(\text{View}(B \leftrightarrow M(x)), \text{View}(B \leftrightarrow M(x'))) $ as the optimal privacy loss for a given interactive mechanism $M$.

**Lemma 2.4.** The partially ordered set $(\mathcal{F}, \preceq)$, where $\mathcal{F}$ consists of all trade-off functions satisfying the conditions in Proposition 1.7, is complete. Specifically, for $S \subseteq \mathcal{F}$, sup $S$ is a trade-off function defined as follows.

$$\sup S = h(\alpha) := \inf_{F: \text{supp}(F) \subseteq S} \{ E[F(A(F))] : E[A(F)] \leq \alpha \} , \quad (4)$$

where $F$ is a random variable that takes value in $S$ and $A : S \rightarrow [0, 1]$ is a function.

**Proof.** We first show that $h$ is the least upper bound for $S$. We shall show that for any tradeoff function $h'$ such that $f \leq h'$ for every $f \in S$, we have $h \preceq h'$. Let $F$ be a random variable such that $\text{supp}(F) \subseteq S$, and let $A : S \rightarrow [0, 1]$ be a function such that $E[A(F)] \leq \alpha$. As stated in Proposition 1.7, a trade-off function is convex and non-increasing, so by Jensen’s inequality, we have

$$h'(\alpha) \leq h'(E[A(F)]) \leq E[h'(A(F))].$$

By the definition of the partial ordering, we have $h'(\alpha) \leq f(\alpha)$ for every $\alpha \in [0, 1]$ and every $f \in S$, so $E[h'(A(F))] \leq E[F(A(F))].$ Therefore, we have

$$h'(\alpha) \leq E[F(A(F))].$$

Taking the infimum over $F$ and $A$ on both sides, we get $h'(\alpha) \leq h(\alpha)$, and therefore, $h \preceq h'$.

Next, we shall show that $h$ is a trade-off function. Following the proposition 1.7, it suffices to check the four properties for $h$. We
begin with proving the convexity of \( h \). For every \( a, b \in [0, 1] \), and every \( \lambda \in [0, 1] \), we have

\[
\begin{align*}
    h(\lambda a + (1 - \lambda)b) & = \inf_{F_A} \{E[F(A(F))] : E[A(F)] \leq \lambda a + (1 - \lambda)b\} \\
    & \leq \lambda \inf_{F_A} \{E[F(A(F))] : E[A(F)] \leq a\} \\
    & \quad + (1 - \lambda) \inf_{F_A} \{E[F(A(F))] : E[A(F)] \leq b\} \\
    & = \lambda h(a) + (1 - \lambda)h(b),
\end{align*}
\]

where inequality (6) is because that for every \( A_a, F_a \) that satisfies \( E[A(F)] \leq a \) and every \( A_b, F_b \) that satisfies \( E[A(F)] \leq b \), the linear combination \( A(\lambda F) = \lambda A_a(F_a) + (1 - \lambda)A_b(F_b) \) satisfies \( E[A(F)] \leq \lambda a + (1 - \lambda)b \). Thus, \( h \) is convex. \( h \) is non-increasing and continuous on \([0, 1]\) due to the monotonicity and continuity of \( f \in \mathcal{F} \) (Proposition 1.7). Finally, since \( f(x) \leq 1 - x \) for every \( f \in \mathcal{F} \), we have

\[
h(a) \leq E[F(A(F))] \leq E[1 - A(F)] \leq 1 - a.
\]

Therefore, \( h \) is a trade-off function, and \( \sup S \) exists.

\[\Box\]

A convenient consequence of joint convexity is that it suffices to consider deterministic adversaries.

**Lemma 2.5.** An interactive mechanism \( M \) is \( \mathcal{D} \cdot \mathcal{D} \) DP, if and only if for every pair of neighboring datasets \( x, x' \), for every deterministic adversary algorithm \( B \), we have \( D(\text{View}(B, M(x))), \text{View}(B, M(x'))) \leq d \).

**Proof.** The necessity is immediately implied by Definition 2.2. We shall prove the sufficiency. Let \( B \) be a randomized adversary. If we fix the coin tosses of \( B \) to a value \( r \), we obtain a deterministic adversary \( B_r \). By hypothesis, we have

\[
D(\text{View}(B, M(x))), \text{View}(B_r, M(x'))) \leq d.
\]

Now let random variable \( R \) be uniformly distributed over the coins of \( A \). Then the view of the randomized adversary \( B \) when interacting with \( M \) consists of the coins \( R \) and the view of the deterministic adversary \( B_R \). That is,

\[
\text{View}(B(\leftrightarrow M(x))) = (R, \text{View}(B_R(\leftrightarrow M(x))),)
\]

and similarly for \( x' \). By joint convexity, we deduce:

\[
\text{View}(B(\leftrightarrow M(x))) \leq d.
\]

\[\Box\]

### 3 COUPLING AND CHAIN RULE PROPERTIES OF \( f \)-DP

In this section, we prove that \( f \)-DP has the coupling and chain rule properties that we use to prove Theorems 1.4 and 1.5.

**Definition 3.1 (Coupling property).** We say that a generalized distance \( D \) has the coupling property if for any two pairs of random variables \( X, X', Y, Y' \), we have \( D(X, X') \leq d \) and \( D(Y, Y') \leq d \), then there exists a coupling of \( X \) and \( Y \) (denoted as \( (X, Y) \)), and a coupling of \( X' \) and \( Y' \) (denoted as \( (X', Y') \)), such that \( D((X, Y), (X', Y')) \leq d \).

**Lemma 3.1 (Lemma 1.6 restated).** \((\mathcal{F}, \preceq, T)\) has the coupling property: Suppose \( f \) is a trade-off function and we have random variables \( X, Y \) and \( X', Y' \) such that

\[
T(X, X') \geq f \quad \text{and} \quad T(Y, Y') \geq f.
\]

Then there exists couplings \((X, Y)\) and \((X', Y')\) such that

\[
T((X, Y)|(X', Y')) \geq f.
\]

We prove this lemma using the following result:

**Theorem 3.2 (Blackwell Theorem [3] (also see [1, 12])).** Let \( P, Q \) be probability distributions on \( X \) and \( P', Q' \) be probability distributions on \( Y \). The following two statements are equivalent:

1. \( T(P, Q) \leq T(P', Q') \).
2. There exists a randomized algorithm \( \text{Proc} : X \rightarrow Y \) such that \( \text{Proc}(P) = P' \) and \( \text{Proc}(Q) = Q' \).

**Proof of Lemma 3.1.** Since a function is called a trade-off function if it is equal to \( T(P, P') \) for some distribution \( P \) and \( P' \), for a given trade-off function \( f \), there exists a pair of random variables \( P, P' \) such that \( T(P, P') = f \). By the Blackwell Theorem, since \( T(P, P') = f \leq T(X, X') \), there exists a randomized algorithm \( P_0 \) such that \( P_0(P) \) and \( P_0(P') \) are identically distributed to \( X \) and \( X' \), respectively. Similarly, since \( T(P, P') = f \leq T(Y, Y') \), there exists a randomized algorithm \( P_1 \) such that \( P_1(P) \) and \( P_1(P') \) are identically distributed to \( Y \) and \( Y' \), respectively. We construct a coupling of \( X \) and \( Y \) as \((P_0(P), P_1(P))\), and a coupling of \( X' \) and \( Y' \) as \((P_0(P'), P_1(P'))\). Then the trade-off function between the two couplings satisfies the following inequality:

\[
T((P_0(P), P_1(P)), (P_0(P'), P_1(P')))) \geq T(P, P') = f,
\]

where Equation (7) follows from Lemma 2.9 in [3], completing the proof.

To formally state the chain rule property, we need a couple of definitions.

**Definition 3.2 (Continuous function).** Let \((A, \preceq)\) and \((B, \preceq)\) be complete posets. A function \( f : A \rightarrow B \) is continuous if \( f(\sup(S)) = \sup(f(S)) \) for every set \( S \subseteq A \).

Observe that every continuous function is monotone: if \( a \leq a' \) are elements of \( A \), then \( f(a') = f(\sup(a, a')) = \sup(f(a), f(a')) \geq f(a) \).

**Definition 3.3 (Continuous in each variable).** Let \( S \) be a finite set and \(|S| = n\). Let \((A, \preceq)\) and \((B, \preceq)\) be complete posets. A function \( f : A^S \rightarrow B \) is continuous in each variable if for every \( i \), and for every \( a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in A \), the function \( g(x) = f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n) \) is a continuous function from \( A \) to \( B \).

**Definition 3.4 (Chain rule).** We say that a generalized probability distance \( \mathcal{D}(\preceq, \Delta) \) satisfies the chain rule property if for every pair of random variables \((X, X')\) on the same domain \( X \), there is a function that is continuous in each variable: \( \text{ChainRule}_{X,X'} : \mathcal{D}(\supp(X), \supp(X')) \rightarrow \mathcal{D} \) such that for every pair of random variables \( Y, Y' \) where \( Y \) is jointly distributed with \( X \) and \( Y' \) is jointly
distributed with \( X' \), we have
\[
\begin{align*}
D((X, Y), (X', Y')) &= \text{ChainRule}_{X, X'}((D(Y|X = x, Y'|X' = x))_{x \in \text{supp}(X) \cap \text{supp}(X')}).
\end{align*}
\]
As an example, the standard chain rule of KL divergence is as follows.
\[
D_{KL}(X, X') = D_{KL}(Y|X) + D_{KL}(Y|X'|X) = D_{KL}(X|X') + E_{x \sim X} D_{KL}(Y|X = x|Y'|X' = x).
\]
So fixing \( X \) and \( X' \), we can calculate the KL divergence for arbitrary \( Y \) and \( Y' \) as a function of the KL divergences \( KL(Y|X = x) | Y'|X' = x \).

In Lemma 3.3, we show that \( f \)-DP has the chain rule property.

**Lemma 3.3 (Lemma 1.7 restated).** For every pair of random variables \( X, X' \) with finite support, there exists a function that is continuous in each variable \( \text{ChainRule}_{X, X'} \) such that for every random variable \( Y \) jointly distributed with \( X \), and every random variable \( Y' \) jointly distributed with \( X' \), we have
\[
T((X, Y), (X', Y')) = \text{ChainRule}_{X, X'}((T(Y|X = x, Y'|X' = x))_{x \in \text{supp}(X) \cap \text{supp}(X')}),
\]
where \( T \) is a trade-off function.

**Proof.**

**Claim 3.4.** The \( \text{ChainRule} \) function for \( f \)-DP is given as follows.
\[
\begin{align*}
\text{ChainRule}_{X, X'}(f_x)_{x \in \text{supp}(X) \cap \text{supp}(X')}((\alpha)) &= \inf_{\alpha \in [0, 1]} \left( E_{x \sim X'}[f_x(\alpha_x)] : E_{x \sim X} [\alpha_x] \leq \alpha \right). \tag{8}
\end{align*}
\]
We first prove this claim. Suppose \( Y \) is jointly distributed with \( X \), and \( Y' \) is jointly distributed with \( X' \). We consider hypothesis tests distinguishing \( (X, Y) \) and \( (X', Y') \). Let \( \phi \) be any decision rule for this testing, \( \alpha(\phi) \) and \( \beta(\phi) \) be the corresponding Type I error and Type II error, respectively. For a given instance \( x \in \text{supp}(X) \cap \text{supp}(X') \), let \( \phi_x(y) := \phi(x, y) \). Additionally, let \( f_x \) be the trade-off function conditioned on \( x \), i.e.,
\[
f_x(\alpha) := T(Y|X = x, Y'|X' = x)(\alpha). \tag{9}
\]
The type I error \( \alpha(\phi) \) and type II error \( \beta(\phi) \) are given as
\[
\alpha(\phi) = E[\phi(x, y)] = E_{x \sim X} E_{y \sim Y} [\phi_x(y)],
\]
and
\[
\beta(\phi) = 1 - E[\phi(x', y')] = 1 - E_{x \sim X'} E_{y \sim Y'} [\phi_x(y')].
\]
For every fixed \( x \in \text{supp}(X) \cap \text{supp}(X') \) and every decision rule \( \phi \) such that \( E_{y \sim Y} [\phi_x(y)] = \alpha_x \), by the definition of \( f_x \) in (9), we have
\[
1 - E_{y' \sim Y'} [\phi_x(y')] \leq f_x(\alpha_x).
\]
Therefore, the trade-off function between \( (X, Y) \) and \( (X', Y') \) satisfies the following inequality:
\[
T((X, Y), (X', Y'))(\alpha) = \inf_{\phi} \left\{ \beta(\phi) : \alpha(\phi) \leq \alpha \right\}
\]
\[
= \inf_{\phi} \left\{ 1 - E_{x \sim X} E_{y \sim Y} [\phi_x(y)] : E_{x \sim X} E_{y \sim Y} [\phi_x(y)] \leq \alpha \right\}
\]
\[
\geq \inf_{\alpha_x} \left\{ E_{x \sim X'}[f_x(\alpha_x)] : E_{x \sim X} [\alpha_x] \leq \alpha \right\}. \tag{10}
\]
On the other hand, by the definition of \( f_x \), for every \( 0 < \alpha_x < 1 \) and \( \delta > 0 \), there exists a decision rule \( \phi(\delta) \) such that \( 1 - E_{y \sim Y} [\phi_x(y)] \leq f_x(\alpha_x) + \delta \) and \( E_{y \sim Y} [\phi_x(y)] \leq \alpha_x \). Then we have
\[
\inf_{\alpha_x} \left\{ E_{x \sim X'}[f_x(\alpha_x)] : E_{x \sim X} [\alpha_x] \leq \alpha \right\}
\]
\[
\geq \inf_{\delta} \left\{ E_{x \sim X'}[1 - E_{y \sim Y} [\phi_x(y)]] - \delta \right\}
\]
\[
= \inf_{\delta} \left\{ \beta(\phi(\delta)) : \alpha(\phi(\delta)) \leq \alpha \right\} - \delta
\]
\[
\geq \inf_{\delta} \left\{ \beta(\phi(\delta)) : \alpha(\phi(\delta)) \leq \alpha \right\} - \delta
\]
\[
= T((X, Y), (X', Y'))(\alpha) - \delta.
\]
Let \( \delta \) go to 0, and combining with Equation (10), we have
\[
T((X, Y), (X', Y'))(\alpha) = \inf_{\alpha_x} \left\{ E_{x \sim X'}[f_x(\alpha_x)] : E_{x \sim X} [\alpha_x] \leq \alpha \right\}. \tag{11}
\]
completing the proof for this claim.

Next, we shall show that the ChainRule function defined in (8) is continuous in each variable. Our goal is to show that for every \( i \), every \( S_i \subseteq D \), and for every \( f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_n \in D \), we have
\[
\begin{align*}
\text{ChainRule}_{X, X'}(f_1, \ldots, f_{i-1}, \sup S_i, f_{i+1}, \ldots, f_n) &= \sup(\text{ChainRule}_{X, X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i).
\end{align*}
\]
Let \( S = (\text{ChainRule}_{X, X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i) \). For every random variable \( F \) such that \( \text{supp}(F) \subseteq S \), we have \( F \) taking values as \( \text{ChainRule}_{X, X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i \), so it is equivalent to consider a random variable \( F_i \) such that \( \text{supp}(F_i) \subseteq S_i \). For every function \( A : S \rightarrow [0, 1] \), we also slightly abuse the notation and use \( A(F_i) \) to represent \( A(F) \). For every \( \alpha \in [0, 1] \), we have
We also have that
\[ \alpha = \text{sup}(\text{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i)(\alpha) \]
\(= \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ \text{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, F_i, f_{i+1}, \ldots, f_n) \right]^2 \right\} \) (by Lemma 2.4)
\(= \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ \text{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, F_i, f_{i+1}, \ldots, f_n) \right]^2 \right\} \) (by (8))
\[ = \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) + F_i(A_j(F_i)) \right] \right\} \] (12)
\[ \leq \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) + F_i(A_j(F_i)) \right] \right\} \] (by Lemma 2.4)
\[ \leq \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) + F_i(A_j(F_i)) \right] \right\} \] (13)
\[ \leq \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) \right] \right\} \] (14)
\[ \leq \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) \right] \right\} \] (by (12))
\[ \leq \inf_{F_i : \supp(F_i) \subseteq S_i} \left\{ E_{F_i} \left[ f_j(A_j(F_i)) \cdot I(\beta \neq i) \right] \right\} \] (15)

\[ \text{View}(B \leftrightarrow M(x)) = \text{View}(B \leftrightarrow P(Y)) \]
\[ \text{View}(B \leftrightarrow M(x')) = \text{View}(B \leftrightarrow P(Y')). \]

Note that the theorem is stated for mechanisms with finite communication, which is formally defined as follows.

**Definition 4.1.** Let \((A, B)\) be an interactive protocol (as in Definition 1.2). We say that \(A\) has finite communication if for every \(x_A\) there is a constant \(c\), such that for all \(r_A, m_1, \ldots, m_{i-1}\), we have
\[ A(x_A, m_1, m_3, \ldots, m_{i-1}; r_A) = \text{halt}. \]
\[ A(x_A, m_1, m_3, \ldots, m_{i-1}; r_A) \leq c. \]

Here \(|y|\) denotes the bit length of string \(y\). \(B\) having finite communication is defined symmetrically.

**Proof of Theorem 4.1.** Our strategy is to apply the induction argument by the number of rounds of interactions. Fix a pair of neighboring datasets \(x, x'\). We consider two cases depending on whether the first message sent from the mechanism \(M\) or the adversary \(B\).

**Case 1.** The adversary \(B\) sends the first query \(q_1\) to the mechanism \(M\). Fixing \(q_1\), the subsequent interactive mechanism \(M_{q_1}\), with input \(x\) is defined by
\[ M_{q_1}(x, q_2, \ldots, q_m; r) = M(x, q_1, q_2, \ldots, q_m; r). \]

We claim that \(\text{View}(A \leftrightarrow M_{q_1})\) consists of \(m - 1\) messages, and \(M_{q_1}\) satisfies \(d\)-DP on the two neighboring datasets \(x\) and \(x'\).
By induction, there exists a randomized interactive post-processing $\mathcal{P}_{\alpha_{1}}$ and a pair of random variables $Y_{q_{1}}, Y'_{q_{1}}$ such that

$$D(Y_{q_{1}}, Y'_{q_{1}}) \leq d,$$

and

$$\mathcal{P}_{\alpha_{1}}(Y_{q_{1}}) \equiv M_{\alpha_{1}}(x) \quad \mathcal{P}_{\alpha_{1}}(Y'_{q_{1}}) \equiv M_{\alpha_{1}}(x').$$

By coupling property, there exists a pair of random variables $Y, Y'$ and a randomized post-processing function $Q_{\alpha_{1}}$ such that $D(Y, Y') \leq d$, and we have that

$$Q_{\alpha_{1}}(Y) = Y_{q_{1}},$$
$$Q_{\alpha_{1}}(Y') = Y'_{q_{1}}.$$

So $Y$ and $Y'$ are produced by coupling all possible queries. Then the interactive post-processing $\mathcal{P}$ is defined by $\mathcal{P}_{\alpha_{1}} \circ Q_{\alpha_{1}}$, i.e.,

$$\mathcal{P}(y, q_{2}, \ldots, q_{m}) = \mathcal{P}_{\alpha_{1}}(Q_{\alpha_{1}}(y), q_{2}, \ldots, q_{m}).$$

**Case 2.** The mechanism $M$ sends the first message $a_{1}$ to the adversary $B$. Let $q_{1}, \ldots, q_{m-1}$ be the queries from the adversary, and $A_{1}, \ldots, A_{m}$ be messages from the mechanism. Fixing $A_{1} = a_{1}$, the subsequent interactive mechanism $M_{a_{1}}$ is defined by

$$M_{a_{1}}(x, q_{1}, \ldots, q_{m-1}; g_{a}(r)) = M(x, q_{1}, \ldots, q_{m-1}; r).$$

$M_{a_{1}}$ uses its randomness to choose uniformly from randomness of $M$ conditioned on $M(x) = a_{1}$. Specifically, let $g_{a}$ be a random transformation such that if $R$ is uniform random for $M$, then for all $x$, $g_{a}(R)$ is uniform on the randomness of $M$ conditioned on $M(x) = a_{1}$.

We define the subsequent adversary

$$B_{a_{1}}(A_{2}, \ldots, A_{m}) = B(a_{1}, A_{2}, \ldots, A_{m}).$$

We know that for all adversary strategy $B \in \mathcal{B}$, we have $D(\text{View}(B \leftrightarrow M(x)), \text{View}(B \leftrightarrow M(x'))) \leq d$, so we have

$$\sup_{B} D(\text{View}(B \leftrightarrow M(x)), \text{View}(B \leftrightarrow M(x'))) \leq d.$$

We have

$$\sup_{B} D(\text{View}(B \leftrightarrow M(x)), \text{View}(B \leftrightarrow M(x'))) = \sup_{B} (\text{ChainRule}_{A_{1}, A'_{1}}(D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x)), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))))\nonumber$$

$$\sup_{B_{a_{1}}} \text{supp}(A_{1}) \supp(A'_{1})) = \sup_{B} \text{ChainRule}_{A_{1}, A'_{1}}(D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x)), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))))\nonumber$$

$$\text{supp}(A_{1}) \supp(A'_{1})) = \text{ChainRule}_{A_{1}, A'_{1}}(\sup_{B} D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x)), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))))\nonumber$$

$$\text{supp}(A_{1}) \supp(A'_{1})) = \text{ChainRule}_{A_{1}, A'_{1}}(\sup_{B} D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x)), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))))\nonumber$$

where (17) follows from the chain rule, and (18) is because that in the case that the first message $a_{1}$ comes from the mechanism, specifying an adversary $B$ for the entire mechanism is equivalent to specifying $B_{a_{1}}$ for every $a_{1}$, then the set of deterministic adversary strategies $B$ we need to sup over is a product set over adversary strategies $B_{a_{1}}$. Equation (19) follows from that the ChainRule$_{A_{1}, A'_{1}}$ function is continuous with respect to each variable. For every $a_{1} \in \text{supp}(A_{1}) \cap \text{supp}(A'_{1}),$ define $d_{a_{1}} = \sup_{B} D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x)), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))).$ Then $M_{a_{1}}$ is $d_{a_{1}} - \mathcal{D} \mathcal{P}$, by induction, there exists a pair of random variables $Y_{a_{1}}, Y'_{a_{1}}$ and a post-processing $\mathcal{P}_{a_{1}}$ such that

$$D(Y_{a_{1}}, Y'_{a_{1}}) \leq d_{a_{1}},$$

and

$$\mathcal{P}_{a_{1}}(Y_{a_{1}}) \equiv M_{a_{1}}(x) \quad \mathcal{P}_{a_{1}}(Y'_{a_{1}}) \equiv M_{a_{1}}(x').$$

Let $Y_{A_{1}}$ be the random variable that defined as $Y_{A_{1}}|_{A_{1}=a_{1}} \sim Y_{a_{1}}$. $Y'_{A_{1}}$ is defined similarly. By the chain rule, we have

$$D((A_{1}, Y_{A_{1}}), (A'_{1}, Y'_{A_{1}})) \leq \text{ChainRule}_{A_{1}, A'_{1}}(\sup_{B} D(\text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x))), \text{View}(B_{a_{1}} \leftrightarrow M_{a_{1}}(x'))) \nonumber$$

(20)

Proof of Theorem 4.2. Following Theorem 4.1, for every interactive $d_{1} - \mathcal{D} \mathcal{P}$ mechanism $M_{j}, j = 1, \ldots, k$, and every pair of neighboring datasets $x, x'$, there exists a pair of random variables $Y_{j}, Y'_{j}$ and an interactive post-processing $\mathcal{P}_{j}$ such that $D(Y_{j}, Y'_{j}) \leq d_{j}$, and for every adversary $B \in \mathcal{B}$, View$(B \leftrightarrow M_{j}(x))$ (resp., View$(B \leftrightarrow M_{j}(x'))$) is identically distributed as View$(B \leftrightarrow \mathcal{P}_{j}(Y_{j}))$ (resp., View$(B \leftrightarrow \mathcal{P}_{j}(Y'_{j}))$. Since $Y_{j}, Y'_{j}, j = 1, \ldots, k$, are noninteractive random variables, which can be viewed as the output distributions of a noninteractive mechanism $N_{j}$ on $x, x'$. Suppose Comp$(N_{1}, N_{1}, \ldots, N_{k})$ is $d_{1} - \mathcal{D} \mathcal{P}$. By the post-processing property, we know that Comp$(\mathcal{P}_{1}(N_{1}), \ldots, \mathcal{P}_{k}(N_{k}))$ is also $d_{1} - \mathcal{D} \mathcal{P}$. Therefore, we have that Comp$(M_{1}, \ldots, M_{k})$ is also $d_{1} - \mathcal{D} \mathcal{P}$. □
5 CONCURRENT COMPOSITION OF RÉNYI DP

In this section, we give a different and simpler proof of the optimal concurrent composition of Rényi DP given in [13]:

**Theorem 5.1 ([13]).** For all $\alpha > 1$, $k \in \mathbb{N}$, $e_1, \ldots, e_k > 0$, and all interactive mechanisms $M_1, \ldots, M_k$ such that $M_i$ is $(e_i, e_i)$-RDP for $i = 1, 2, \ldots, k$, the concurrent composition $\text{ConComp}(M_1, \ldots, M_k)$ of interactive mechanisms $M_1, \ldots, M_k$ is $(\alpha^k e_1, e_1)$-RDP.

We prove this theorem by characterizing optimal $\alpha$-RDP adversary strategy in Lemma 5.5.

**Definition 5.1 (Optimal $\alpha$-RDP adversary).** For an interactive mechanism $M$, and neighboring datasets $x$ and $x'$, an optimal $\alpha$-RDP adversary with respect to $x$ and $x'$ is a strategy $B^\text{OPT}$ such that for all adversary strategies $B$,

$$D_{\alpha}(\text{View}(B^\text{OPT} \leftrightarrow M(x))) \leq D_{\alpha}(\text{View}(B \leftrightarrow M(x))),$$

We show that the optimal adversary strategy against the concurrent composition of $k$ mechanisms can be decomposed as a product of optimal adversaries against each mechanism independently:

**Lemma 5.2.** Let $B^\text{OPT}(1), B^\text{OPT}(2), \ldots, B^\text{OPT}(k)$ be optimal $\alpha$-RDP adversaries against $M_1, M_2, \ldots, M_k$. Then

$$B^\text{OPT} = B^\text{OPT}(1) \times B^\text{OPT}(2) \times \ldots \times B^\text{OPT}(k)$$

is an optimal $\alpha$-RDP adversary against $\text{ConComp}(M_1, M_2, \ldots, M_k)$, where $B^\text{OPT}(1) \times B^\text{OPT}(2)$ denotes the adversary’s strategy where it takes $B^\text{OPT}(1)$ to interact with $M_1$ and takes $B^\text{OPT}(2)$ to interact with $M_2$.

Although this property of the optimal adversary strategy can be derived as a consequence of the optimal concurrent composition of Rényi DP in [13], we take a different approach to first prove this property and then use it to prove the optimal concurrent composition theorem for Rényi DP.

Our proof relies on the following two properties of Rényi divergence: the monotonicity property in Lemma 5.3 and the independence property in Lemma 5.4. The Rényi divergence is defined as follows.

**Definition 5.2 (Rényi divergence [16]).** For two probability distributions $P$ and $Q$, the Rényi divergence of order $\alpha > 1$ is

$$D_{\alpha}(P||Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^\alpha.$$

**Lemma 5.3.** For any two tuples of jointly distributed random variables $(U, V, W)$ and $(U', V', W')$ over the same measurable space, if for every $u \in \text{supp}(U)$, we have

$$D_{\alpha}(V|U=u||V'|U=u) \leq D_{\alpha}(W|U=u||W'|U=u),$$

then

$$D_{\alpha}((U, V)||(U', V')) \leq D_{\alpha}((U, W)||(U', W')).$$

**Proof.**

$$D_{\alpha}((U, V)||(U', V'))$$

$$= \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^\alpha$$

$$= \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

$$= \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

The following lemma describes the optimal adversary’s strategy against an interactive mechanism. The proof of Lemma 5.5 uses the monotonicity property of Rényi divergence.

**Lemma 5.5.** The optimal adversary $B^\text{OPT}$ with respect to $x$ and $x'$, which is the adversary strategy that maximizes the Rényi divergence of the views for all fixed $x, x'$, chooses the first query $q_1$ to maximize

$$D_{\alpha} \left( A_1, \text{View} \left( B^\text{OPT}_{q_1, A_1} \leftrightarrow M_{q_1, A_1} (x) \right) \right) \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

$$= D_{\alpha} \left( A_1, \text{View} \left( B^\text{OPT}_{q_1, A_1} \leftrightarrow M_{q_1, A_1} (x) \right) \right) \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

where $A_1 = M(x, q_1)$ and $A_1' = M(x', q_1)$, and $B^\text{OPT}_{q_1, A_1'}$ is any optimal adversary against $M_{q_1, A_1}$, which denotes the subsequent mechanism when fixing $q_1, a_1$.

**Proof.** We decompose the view of the adversary into two parts: the first answer $A_1$ to the query $q_1$, and the view of the subsequent interaction. Fixing $q_1$, for every adversary $B$, we have

$$D_{\alpha}(\text{View}(B \leftrightarrow M(x))) \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

$$= D_{\alpha} \left( A_1(q_1), \text{View} \left( B_{q_1, A_1} \leftrightarrow M_{q_1, A_1} (x) \right) \right) \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

$$= D_{\alpha} \left( A_1(q_1), \text{View} \left( B_{q_1, A_1} \leftrightarrow M_{q_1, A_1} (x) \right) \right) \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]^{\alpha - 1} \mathbb{E}_{x \sim Q} \left[ \frac{P(x)}{Q(x)} \right]$$

517
where $B_{q_1 A_1} \leftrightarrow M_{q_1 A_1}$ denotes the subsequent interaction. For every $a_1 \in \text{supp}(A_1) \cap \text{supp}(A'_1)$ and for every $B_{q_1 A_1}$, by the definition of $\mathcal{B}_{q_1 A_1}^{\text{OPT}}$, we have

\[
D_{q_1 A_1} \left( \text{View} \left( B_{q_1 A_1} \leftrightarrow M_{q_1 A_1} (x) \right) |_{A_1 = a_1} \right) \\
\leq D_{q_1 A_1} \left( \text{View} \left( B_{q_1 A_1}^{\text{OPT}} \leftrightarrow M_{q_1 A_1} (x') \right) |_{A_1 = a_1} \right)
\]

Thus, we have

\[
D_{q_1 A_1} \left( \left( A_1 (q_1), \text{View} \left( B_{q_1 A_1} \leftrightarrow M_{q_1 A_1} (x) \right) \right) \right) \\
\leq D_{q_1 A_1} \left( \left( A'_1 (q_1), \text{View} \left( B_{q_1 A_1}^{\text{OPT}} \leftrightarrow M_{q_1 A_1} (x') \right) \right) \right)
\]

where (21) follows from Lemma 5.3. It implies that in order to maximize (20), it suffices to choose $q_1$ to maximize the quantity in (21).

We then use Lemma 5.5 to prove Lemma 5.2.

**PROOF OF Lemma 5.2.** We will use induction on the rounds of messages to prove this lemma. We can use induction argument because of the assumption of finite communication. Without loss of generality, suppose the first query from the adversary is sent to $A_1$, and we use $\text{ConComp} = \text{ConComp}(A_1, A_2, \ldots, A_k)$ to simplify the notation. Following Lemma 5.5, the optimal adversary $\mathcal{B}_{q_1 A_1}^{\text{OPT}}$ chooses $q_1$ as follows.

\[
\mathcal{B}_{q_1 A_1}^{\text{OPT}} = \arg\max_{q_1} D_{q_1 A_1} \left( \left( A_1 (q_1), \text{View} \left( B_{q_1 A_1}^{\text{OPT}} \leftrightarrow \text{ConComp}_{q_1 A_1} (x) \right) \right) \right)
\]

where $A_1 = M_1 (x, q_1)$ and $A'_1 = M_1 (x', q_1)$. By induction, we assume that $\mathcal{B}_{q_1 A_1}^{\text{OPT}} = \mathcal{B}_{A_1}^{\text{OPT}(1)} \times \mathcal{B}_{A_1}^{\text{OPT}(2)} \times \cdots \times \mathcal{B}_{A_1}^{\text{OPT}(k)}$. Let

\[
\tilde{V}_{q_1 A_1} = \text{View} \left( B_{q_1 A_1}^{\text{OPT}(1)} \leftrightarrow M_{q_1 A_1} (x) \right),
\]

and similarly,

\[
\tilde{V}'_{q_1 A_1} = \text{View} \left( B_{q_1 A_1}^{\text{OPT}(1)} \leftrightarrow M_{q_1 A_1} (x') \right).
\]

Let

\[
\tilde{V} = \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(2)} \times \cdots \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)} \leftrightarrow \text{ConComp} (M_2, \ldots, M_k) (x) \right),
\]

and

\[
\tilde{V}' = \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(2)} \times \cdots \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)} \leftrightarrow \text{ConComp} (M_2, \ldots, M_k) (x') \right).
\]

With these notations, we have

\[
\arg\max_{q_1} D_{q_1 A_1} \left( \left( A_1, \text{View} \left( B_{q_1 A_1}^{\text{OPT}} \leftrightarrow \text{ConComp}_{q_1 A_1} (x) \right) \right) \right) \\
\left( A'_1, \text{View} \left( B_{q_1 A_1}^{\text{OPT}} \leftrightarrow \text{ConComp}_{q_1 A_1} (x') \right) \right)
\]

\[
= \arg\max_{q_1} D_{q_1 A_1} \left( \left( A_1, \tilde{V}_{q_1 A_1} \right) \| \left( A'_1, \tilde{V}'_{q_1 A_1} \right) \right)
\]

\[
= \arg\max_{q_1} D_{q_1 A_1} \left( \left( A_1, \tilde{V}_{q_1 A_1} \right) \| \left( A'_1, \tilde{V}'_{q_1 A_1} \right) + D_{q_1 A_1} (\tilde{V} || \tilde{V}') \right),
\]

(by Lemma 5.4)

\[
= \arg\max_{q_1} D_{q_1 A_1} \left( \left( A_1, \tilde{V}_{q_1 A_1} \right) \| \left( A'_1, \tilde{V}'_{q_1 A_1} \right) \right)
\]

\[
= \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)}.
\]

Therefore, the optimal adversary $\mathcal{B}_{q_1 A_1}^{\text{OPT}}$ chooses $q_1$ just as the optimal adversary $\mathcal{B}_{q_1 A_1}^{\text{OPT}(1)}$ against only $A_1$. Since $\mathcal{B}_{q_1 A_1}^{\text{OPT}} = \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)} \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(2)} \times \cdots \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)}$, we have $\mathcal{B}_{q_1 A_1}^{\text{OPT}} = \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)} \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(2)} \times \cdots \times \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)}$, completing the proof.

We now prove Theorem 5.1 using Lemma 5.2.

**PROOF OF Theorem 5.1.** Following Lemma 5.2, we have

\[
D_{q_1 A_1} \left( \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}} \leftrightarrow \text{ConComp} (x) \right) \right) \\
= D_{q_1 A_1} \left( \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)} \leftrightarrow \text{ConComp} (x') \right) \right)
\]

\[
= D_{q_1 A_1} \left( \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)} \leftrightarrow M_{1} (x) \right) \| \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(1)} \leftrightarrow M_{1} (x') \right) \right)
\]

\[
+ \cdots + D_{q_1 A_1} \left( \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)} \leftrightarrow M_{k} (x) \right) \| \text{View} \left( \mathcal{B}_{q_1 A_1}^{\text{OPT}(k)} \leftrightarrow M_{k} (x') \right) \right)
\]

\[
= \sum_{i=1}^{k} \epsilon_i,
\]

completing the proof.

**ACKNOWLEDGMENTS**

S.V. is supported by a grant from the Sloan Foundation and a Simons Investigator Award. W.Z. is supported by a Computing Innovation Fellowship from the Computing Research Association (CRA) and the Computing Community Consortium (CCC). A preliminary version of this work was presented as a poster at TPDP ’22 and a full version is posted on arXiv [18]. We are grateful to Xin Lyu for pointing out the error in the proof of our previously claimed concurrent composition theorem for Rényi DP [13]. We also thank an anonymous reviewer for pointing out the error in our previous proof of continuity.

**REFERENCES**

[1] David Blackwell. 1953. Equivalent comparisons of experiments. The annals of mathematical statistics (1953), 265–272.

[2] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Theory of Cryptography Conference. Springer, 635–658.

[3] Jinshuo Dong, Aaron Roth, and Weijie J Su. 2019. Gaussian differential privacy. arXiv preprint arXiv:1905.02383 (2019).

[4] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In Annual international conference on the theory and applications of cryptographic techniques. Springer, 486–503.
Concurrent Composition Theorems for Differential Privacy

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In Proceedings of the 3rd Conference on Theory of Cryptography (TCC ’06). 265–284.

[6] Cynthia Dwork, Moni Naor, Tzioni Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC ’10). 715–724.

[7] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In Proceedings of the 41st ACM Symposium on Theory of Computing (STOC ’09). 381–390.

[8] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science 9, 3–4 (2014), 211–407.

[9] Cynthia Dwork and Guy N Rothblum. 2016. Concentrated differential privacy. arXiv preprint arXiv:1605.01887 (2016).

[10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. IEEE, 51–60.

[11] Moritz Hardt and Guy N Rothblum. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In 2010 IEEE 51st annual symposium on foundations of computer science. IEEE, 61–70.

[12] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In International conference on machine learning. PMLR, 1376–1385.

[13] Xin Lyu. 2022. Composition Theorems for Interactive Differential Privacy. In Thirty-sixth Conference on Neural Information Processing Systems.

[14] Ilya Mironov. 2017. Rényi differential privacy. In 2017 IEEE 30th Computer Security Foundations Symposium (CSF). IEEE, 263–275.

[15] Jack Murtagh and Salil Vadhan. 2016. The complexity of computing the optimal composition of differential privacy. In Theory of Cryptography Conference. Springer, 157–175.

[16] Alfred Rényi. 1961. On measures of entropy and information. In Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, Vol. 1. Berkeley, California, USA.

[17] Salil Vadhan and Tianniao Wang. 2021. Concurrent Composition of Differential Privacy. In Theory of Cryptography Conference. Springer, 582–604.

[18] Salil Vadhan and Wanrong Zhang. 2022. Concurrent Composition Theorems for all Standard Variants of Differential Privacy. arXiv preprint arXiv:2207.08335 (2022).

[19] Tim Van Erven and Peter Harremos. 2014. Rényi divergence and Kullback-Leibler divergence. IEEE Transactions on Information Theory 60, 7 (2014), 3797–3820.

[20] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. J. Amer. Statist. Assoc. 60, 309 (1965), 63–69.

[21] Larry Wasserman and Shuheng Zhou. 2010. A statistical framework for differential privacy. J. Amer. Statist. Assoc. 105, 489 (2010), 375–389.

Received 2022-11-07; accepted 2023-02-06