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 (ϵ, δ)-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 (ϵ, δ)-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 $\mathcal{X}$, and a binary *neighboring* relation on $\mathcal{X}$, 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 $\mathcal{X} = \mathbb{R}^n$, 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 $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ is* (ϵ, δ)-differentially private *if for every pair of neighboring datasets $x, x' \in \mathcal{X}$, and for every subset of possible outputs $\mathcal{S} \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \le \exp(\epsilon) \cdot \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta.$$

Thus, differential privacy requires that for all neighboring datasets x, x', $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are close as probability distributions (as measured by the parameters ϵ and δ). 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 $\mathcal{M}$ 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 $\mathcal{M}$ as a party in an interactive protocol, interacting with a (possibly adversarial) analyst.

Definition 1.2 (Interactive protocols). *An* interactive protocol *(A, B) 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 $m_i = A(x_A, m_1, m_3, \ldots, m_{i-1}; r_A)$.
    (b) If i is odd, let $m_i = B(x_B, m_0, m_2, \ldots, m_{i-1}; r_B)$.
    (c) If $m_i =$ `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 (A, B) 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* $\mathrm{View}_A(A(x_A; r_A) \leftarrow B(x_B; r_B)) = (r_A, x_A, m_1, m_3, \ldots)$ *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.*

STOC '23, June 20–23, 2023, Orlando, FL, USA

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 $\mathrm{View}(B \leftrightarrow \mathcal{M}(x))$. With this notation, interactive differential privacy is defined by asking for the views of an adversary on any pair of neighboring datasets $\mathrm{View}(B \leftrightarrow \mathcal{M}(x))$ and $\mathrm{View}(B \leftrightarrow \mathcal{M}(x'))$ satisfying the same (ϵ, δ)-closeness notion as in non-interactive differential privacy.

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

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

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 $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ as the non-interactive mechanism $\mathcal{M} = \mathrm{Comp}(\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$ defined as

$$\mathcal{M}(x) := (\mathcal{M}_1(x), \mathcal{M}_2(x), \ldots, \mathcal{M}_k(x)), \tag{1}$$

where each mechanism $\mathcal{M}$ 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 $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, and thus the adaptive composition can be viewed as an interactive mechanism.

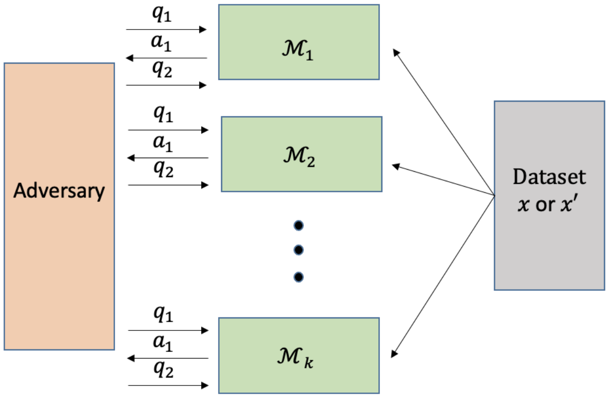

Figure 1: Concurrent composition of interactive mechanisms

In many cases, analysts may wish to perform multiple *interactive* analyses on the same dataset concurrently, which raises the question of *concurrent composition*, first studied for differential privacy in [17]. In this setting (illustrated in Figure 1), an adversary can arbitrarily interleave its queries to several differentially private mechanisms, and those queries might be correlated and depends on the answer received in other mechanisms. As a motivating example, several organizations might set up multiple DP query systems on datasets that may refer to the same set of individuals. Each query system has its own privacy budget ϵ. Suppose an adversary can concurrently access those systems, and a query sent to one system might depends on all the previous messages that received from other systems. For example, when we run two Sparse Vector mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ concurrently, the queries for $\mathcal{M}_1$ depends on previous answers from $\mathcal{M}_2$, and vice-versa, but we only know the overall privacy guarantees for $\mathcal{M}_1$ and $\mathcal{M}_2$ when they are executed independently. If the executions were *sequential*, meaning that the adversary completes its interaction with $\mathcal{M}_1$ before issuing any queries to $\mathcal{M}_2$, then we can hardwire the answers from $\mathcal{M}_1$ into the adversary's strategy. When attacking $\mathcal{M}_2$, the privacy loss for $\mathcal{M}_2$ will be bounded as usual. But when the queries are interleaved, it is no longer clear how to define a fixed adversary strategy against either of the mechanisms $\mathcal{M}_1$ or $\mathcal{M}_2$. It is not clear if the adversary can run any concurrent attack to break the privacy guarantees, and therefore, we wish to provide a provable guarantee to account for the total privacy loss in such systems. Formally, the concurrent composition of interactive mechanisms is defined as follows.

Definition 1.5 (Concurrent composition of interactive mechanisms[17])**.** *Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be interactive mechanisms. $\mathcal{M} = ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is the concurrent composition of mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ defined as follows:*

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

(a) *Parse m_{i-1} as (j,q) where $j=1,\ldots,k$ and q is a query to $\mathcal{M}_j$. If m_{i-1} cannot be parsed correctly, output* `halt`.
(b) *Extract history $(m_0^j,\ldots,m_{t-1}^j)$ from $(m_0,\ldots,m_{i-1})$ where m_i^j are all of the queries to mechanism $\mathcal{M}_j$.*
(c) *Output $\mathcal{M}_j(x_j, m_0^j,\ldots,m_{t-1}^j; r_j)$.*

For an adversary B, we will use the notation $\text{View}(B \leftrightarrow (\mathcal{M}_1,\ldots,\mathcal{M}_k))$ *as shorthand for* $\text{View}(B \leftrightarrow \textit{ConComp}(\mathcal{M}_1,\ldots,\mathcal{M}_k))$

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 for all non-interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ such that $\mathcal{M}_i$ is (ϵ_i,δ_i)-differentially private for $\delta_1=\delta_2=\ldots=\delta_k=0$, their composition $Comp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ is (ϵ,δ)-differentially private. Then for all interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ such that $\mathcal{M}_i$ is (ϵ_i,δ_i)-differentially private for $\delta_1=\delta_2=\ldots=\delta_k=0$, the concurrent composition $ConComp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ of interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ is (ϵ,δ)-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 $\mathcal{M}$ 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*

$$\text{View}(B \leftrightarrow \mathcal{M}(x)) \equiv \text{View}(B \leftrightarrow \mathcal{P}(RR_\epsilon(0)))$$
$$\text{View}(B \leftrightarrow \mathcal{M}(x')) \equiv \text{View}(B \leftrightarrow \mathcal{P}(RR_\epsilon(1))).$$

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

Note that Theorem 1.1 and Theorem 1.2 do not apply to the case where the composed mechanisms $\mathcal{M}_i$ are (ϵ_i,δ_i)-DP for $\delta_i>0$. In this case, [17] only show a bound that is similar to the "group privacy" property of (ϵ,δ)-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 $\text{ConComp}(\mathcal{M}_1,\mathcal{M}_2,\ldots,\mathcal{M}_k)$ is $(k\epsilon, \frac{\exp(k\epsilon)-1}{\exp(\epsilon)-1}\delta)$-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 (ϵ_i,δ_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 $\delta_i>0$:

Theorem 1.3 (Concurrent composition for (ϵ,δ)-DP interactive mechanisms). *Suppose that for all non-interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ such that $\mathcal{M}_i$ is (ϵ_i,δ_i)-differentially private for $i=1,2\ldots,k$, their composition $Comp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ is (ϵ,δ)-differentially private. Then for all interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ with finite communication such that $\mathcal{M}_i$ is (ϵ_i,δ_i)-differentially private for $i=1,2\ldots,k$, the concurrent composition $ConComp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ of interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ is (ϵ,δ)-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 for all non-interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ such that $\mathcal{M}_i$ is f_i-DP for $i=1,2\ldots,k$, their composition $Comp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ is f-DP. Then for all interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ such that $\mathcal{M}_i$ is f_i-DP for $i=1,2\ldots,k$, the concurrent composition $ConComp(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ of interactive mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ is f-DP.*

Theorem 1.3 follows directly from Theorem 1.4 because f-DP defined below captures (ϵ,δ)-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 (ϵ,δ)-differential privacy based on the hypothesis testing interpretation of differential privacy. Differential privacy attemps to measure the difficulty of distinguishing two neighboring datasets based on the ouput 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 $\mathcal{M}$ on the two neighboring datasets, namely $\mathcal{M}(x)$ and $\mathcal{M}(x')$. For a given rejection rule ϕ, the type I error $\alpha_\phi=\mathrm{E}[\phi(Y)]$ is the probability of rejecting H_0 when H_0 is true, while the type II error $\beta_\phi=1-\mathrm{E}[\phi(Y')]$ 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 *$T(Y,Y'):[0,1]\to[0,1]$ as*

$$T(Y,Y')(\alpha)=\inf\{\beta_\phi : \alpha_\phi\le\alpha\}, \tag{2}$$

where the infimum is taken over all (measurable) rejection rules ϕ.

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 $f:[0,1]\to[0,1]$ is a trade-off function if and only if f is convex, continuous, non-increasing, and $f(x)\le 1-x$ 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 $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ is* f-differentially private *if for every pair of neighboring datasets $x, x' \in \mathcal{X}$, we have*

$$T(\mathcal{M}(x), \mathcal{M}(x')) \geq f.$$

(ϵ, δ)-DP is a special case of f-DP, taking $f = f_{\epsilon,\delta}$, where $f_{\epsilon,\delta} = \max\{0, 1 - \delta - \exp(\epsilon)\alpha, \exp(-\epsilon)(1 - \delta - \alpha)\}$ [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 $\mathcal{M}$ with finite communication, and every pair of neighboring datasets x, x', there exists a non-interactive f-DP mechanism $\mathcal{N}$ and an randomized interactive post-processing mechanism $\mathcal{P}$ such that for every adversary $B \in \mathcal{B}$, we have*

$$\mathrm{View}(B \leftrightarrow \mathcal{M}(x)) \equiv \mathrm{View}(B \leftrightarrow \mathcal{P}(\mathcal{N}(x)))$$
$$\mathrm{View}(B \leftrightarrow \mathcal{M}(x')) \equiv \mathrm{View}(B \leftrightarrow \mathcal{P}(\mathcal{N}(x'))).$$

Similarly to Theorem 1.2, in the case of (ϵ, δ)-DP, one can take the non-interactive mechanism $\mathcal{N}$ 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 $\mathcal{M}_i$, it suffices to consider the composition of the non-interactive mechanisms $\mathcal{N}_i$. 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 noninteractive task: a sample from the distribution is generated and given to the hypothesis tester, which then tries to decide whether the distribution is in H_0 or H_1. However, suppose instead we consider the task of distinguishing between two interactive mechanisms $\mathcal{M}_0$ and $\mathcal{M}_1$, 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 $\mathcal{M}_0$ as $\mathcal{M}(x)$ and $\mathcal{M}_1$ as $\mathcal{M}(x')$, Theorem 1.5 implies that the two interactive mechanisms $\mathcal{M}_0$ and $\mathcal{M}_1$ can be simulated perfectly by noninteractive random variables $\mathcal{N}_0 = \mathcal{N}(x)$ and $\mathcal{N}_1 = \mathcal{N}(x')$ such that even if we give $\mathcal{N}_0$ or $\mathcal{N}_1$ to a hypothesis tester in its entirety (thereby revealing how $\mathcal{M}_0$ or $\mathcal{M}_1$ would answer *all* questions), it cannot distinguish them any better than it could distinguish $\mathcal{M}_0$ and $\mathcal{M}_1$. The trick, of course, is that the simulation is "perfect" only when executing a single interaction with $\mathcal{M}_0$ or $\mathcal{M}_1$ (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*

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

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

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

A coupling of random variables X and Y is any random vector $(\tilde{X}, \tilde{Y})$ such that the marginal distributions are identically distributed to X and Y respectively, i.e., $\tilde{X} \equiv X$ and $\tilde{Y} \equiv Y$. Subject to this constraint on the marginal, $\tilde{X}$ and $\tilde{Y}$ 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* $\mathrm{ChainRule}_{X,X'}$ *such that for every random variable Y jointly distributed with X, and every random variable Y' jointly distributed with X', we have*

$$T((X, Y), (X', Y')) = \mathrm{ChainRule}_{X,X'}\big((T(Y|X = x, Y'|X' = x))\big)_{x \in \mathrm{supp}(X) \cap \mathrm{supp}(X')}\big).$$

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 $x \in \mathrm{supp}(X) \cap \mathrm{supp}(X')$ through a ChainRule function. The terminology "chain rule" is by analogy with the standard chain rule for KL divergence, which says

$$\mathrm{KL}((X, Y)||(X', Y')) = \mathrm{KL}(X||X') + \mathrm{E}_{x \sim X}\mathrm{KL}(Y|X = x||Y'|X' = x). \tag{3}$$

So fixing X and X', we can calculate the KL divergence for arbitrary Y and Y' as a function of the KL divergences $\mathrm{KL}(Y|X = x||Y'|X' = x)$. (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 $\{\epsilon_j, \delta_j\}_j$ that characterize the "closeness" of $Y|X = x$ 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 $\mathcal{M}$ 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 $\mathcal{M}$ or the adversary B, we consider the following two cases.

Case 1. The adversary B sends the first query q_1 to the mechanism $\mathcal{M}$. Fix a pair of neighboring datasets x, x'. Fixing q_1, we denote the subsequent interactive mechanism by $\mathcal{M}_{q_1}$. By induction,

$\mathcal{M}_{q_1}$ can be simulated by a post-processing of a non-interactive f-DP mechanism $\mathcal{N}_{q_1}$. Then we obtain $\mathcal{N}(x)$ and $\mathcal{N}(x')$ by coupling the pairs $\mathcal{N}_{q_1}(x)$ and $\mathcal{N}_{q_1}(x')$ on all the values of q_1, which is finite by our assumption of finite communication. We note that the coupling lemma 1.6 extends to finite number of random variables $Y_1, \ldots, Y_k$ and $Y'_1, \ldots, Y'_k$ by induction on k. Following the coupling lemma, we have $T(\mathcal{N}(x), \mathcal{N}(x')) \geq f$. We can combine the interactive post-processing mechanisms $\mathcal{P}_{q_1}$ for all q_1 to obtain the interactive post-processing $\mathcal{P}$ that simulates $\mathcal{M}(x)$ and $\mathcal{M}(x')$ from $\mathcal{N}(x)$ and $\mathcal{N}(x')$. Thus, we have Theorem 1.5 holds for k rounds of interactions.

Case 2. The mechanism $\mathcal{M}$ sends the first message a_1 to the adversary B. Fix a pair of neighboring datasets x, x'. We denote the mechanism conditioned on every a_1 by $\mathcal{M}_{a_1}$, and let f_{a_1} be the trade-off function of $\mathcal{M}_{a_1}$ (maximized over all adversaries). We denote the random variable of the first message as A_1, A'_1 on datasets x, x', respectively. By induction, $\mathcal{M}_{a_1}$ can be simulated by a post-processing of a non-interactive f_{a_1}-DP mechanism $\mathcal{N}_{a_1}$. Thus, $\mathcal{M}$ can be simulated by a post-processing of the non-interactive mechanism $\mathcal{N}$ where $\mathcal{N}(x) \equiv (A_1, \mathcal{N}_{A_1}(x))$ and $\mathcal{N}(x') \equiv (A'_1, \mathcal{N}_{A'_1}(x'))$. We use the chain rule to argue that $T(\mathcal{N}(x), \mathcal{N}(x'))$ $\geq \text{ChainRule}_{A_1, A'_1}\left((f_{a_1})_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A'_1)}\right) = f$. We conclude that Theorem 1.5 holds for k rounds of interactions.

1.6 Independent Work by Lyu

In independent and concurrent work, Lyu [13] proves Theorem 1.3 with a different argument. They show that every interactive (ϵ, δ)-DP mechanism can be simulated by interactive post-processing of a non-interactive (ϵ, δ)-DP mechanism, via an argument that is specific to (ϵ, δ)-DP that does not seem to generalize to arbitrary tradeoff functions f. Indeed, they leave the the general case of f-DP as an open problem, which is solved by our Theorems 1.4 and 1.5.

On the other hand, Lyu [13] also proves an optimal concurrent composition theorem for Rényi DP of any fixed order. In an earlier version of our paper [18], we also claimed such a result, but our proof was incorrect (except for the case of Rényi DP of order $\alpha = 1$),[1] as pointed out to us by Lyu. In this revision, we give a simple proof of Lyu's theorem for Rényi DP by characterizing the optimal adversary strategy in Section 5.

2 GENERALIZED DEFINITIONS OF DP MECHANISMS

To prove our results and discuss the several variants of differential privacy, it is convenient to introduce a more general abstraction, where distances between probability distributions can be in an arbitrary partially ordered set.

Definition 2.1 (Generalized probability distance). *A* generalized probability distance measure *is a tuple* $(\mathcal{D}, \preceq, D)$ *such that*

1. $(\mathcal{D}, \preceq)$ *is a partially ordered set (poset).*
2. D *is a mapping that takes any two random variables* X, X' *over the same measurable space to an element* $D(X, X')$ *of* $\mathcal{D}$.
3. *(Post-processing.) The generalized distance mapping* D *is closed under post-processing, meaning that for every function* g, $D(g(X), g(X')) \preceq D(X, X')$.
4. *(Joint Convexity.) Suppose we have a collection of random variables* $(X_i, X'_i)_{i \in \mathcal{I}}$ *and a random variable* I *distributed on* $\mathcal{I}$. *If* $D(X_i, X'_i) \preceq d$ *for all* $i \in \mathcal{I}$, *then* $D(X_I, X'_I) \preceq d$.

For the generalized notion d-$\mathcal{D}$ 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-$\mathcal{D}$ DP). *Let* $(\mathcal{D}, \preceq, D)$ *be a generalized probability distance. For* $d \in \mathcal{D}$, *we call an interactive mechanism* $\mathcal{M}$ d-$\mathcal{D}$ DP *if for every adversary* $B \in \mathcal{B}$ *and every pair of neighboring datasets* x, x', *we have*

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

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 ϵ provides stronger privacy guarantee, so the partially ordered set $\mathcal{D}$ is defined as $((\mathbb{R}^{\geq 0}) \cup \{\infty\}, \leq)$. The distance mapping is the max-divergence D_∞. For two probability distributions P and Q, the max-divergence is

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

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

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

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

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* $\lambda \in (0, 1)$,

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

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

[1] Specifically, we stated and used a chain rule for Rényi divergence that only holds for order $\alpha = 1$ (i.e. KL divergence).

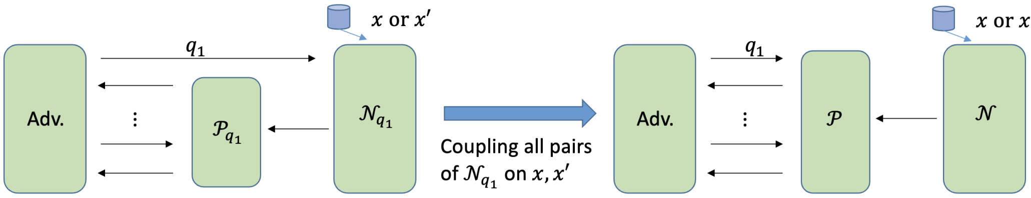

Figure 2: Case 1: The adversary B sends the first query q_1.

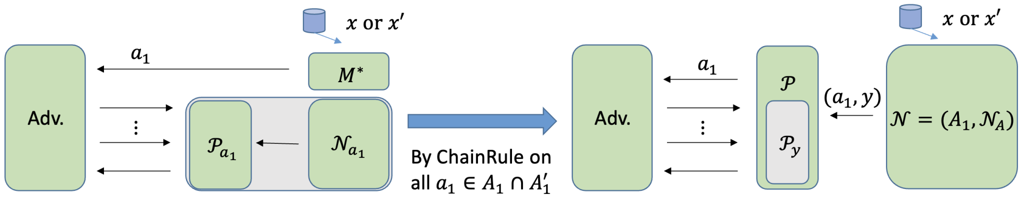

Figure 3: Case 2: The mechanism $\mathcal{M}$ sends the first message a_1.

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 $(X_i, X_i')_{i\in\mathcal{I}}$ and a random variable I distributed on $\mathcal{I}$. If $T(X_i, X_i') \geq f$ for all $i \in \mathcal{I}$, then $T(X_I, X_I') \geq f$.*

Proof. For any random variable I distributed on $\mathcal{I}$. We have

$$\begin{aligned}
T(X_I, X_I')(\alpha) &= \inf_{\phi}\left\{\mathrm{E}\left[1-\phi(X_I')\right] : \mathrm{E}[\phi(X_I')] \leq \alpha\right\}\\
&= \inf_{\phi}\left\{\mathrm{E}_{i\sim I}\mathrm{E}[1-\phi(X_i')] : \mathrm{E}_{i\sim I}\mathrm{E}[\phi(X_i)] \leq \alpha\right\}\\
&\geq \inf_{\phi}\left\{\mathrm{E}_{i\sim I}[f(\mathrm{E}[\phi(X_i)])] : \mathrm{E}_{i\sim I}\mathrm{E}[\phi(X_i)] \leq \alpha\right\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad (f \text{ non-decreasing})\\
&\geq \inf_{\phi}\left\{f(\mathrm{E}_{i\sim I}\mathrm{E}[\phi(X_i)]) : \mathrm{E}_{i\sim I}\mathrm{E}[\phi(X_i)] \leq \alpha\right\} \quad (f \text{ convex})\\
&= f(\alpha).
\end{aligned}$$

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 $((\mathbb{R}^{\geq 0}) \cup \{\infty\}, \leq)$ used in pure DP and Rènyi 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 $d = \sup_B D(\mathrm{View}(B \leftrightarrow \mathcal{M}(x)), \mathrm{View}(B \leftrightarrow \mathcal{M}(x')))$ as the optimal privacy loss for a given interactive mechanism $\mathcal{M}$.

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.*

$$\sup S = h(\alpha) := \inf_{\substack{F:\mathrm{supp}(F)\subseteq S\\ A:S\to[0,1]}} \left\{\mathrm{E}[F(A(F))] : \mathrm{E}[A(F)] \leq \alpha\right\}, \qquad (4)$$

where F is a random variable that takes value in S and $A : S \longrightarrow [0,1]$ 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 $f \preceq h'$ for every $f \in S$, we have $h \preceq h'$. Let F be a random variable such that $\mathrm{supp}(F) \subseteq S$, and let $A : S \to [0,1]$ be a function such that $\mathrm{E}[A(F)] \leq \alpha$. As stated in Proposition 1.7, a trade-off function is convex and non-increasing, so by Jensen's inequality, we have

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

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 $\mathrm{E}[h'(A(F))] \leq \mathrm{E}[F(A(F))]$. Therefore, we have

$$h'(\alpha) \leq \mathrm{E}[F(A(F))],$$

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

$$\begin{aligned} & h(\lambda a + (1-\lambda)b) && (5)\\ &= \inf_{F,A} \{\mathrm{E}[F(A(F))] : \mathrm{E}[A(F)] \le \lambda a + (1-\lambda)b\} \\ &\le \lambda \inf_{F,A} \{\mathrm{E}[F(A(F))] : \mathrm{E}[A(F)] \le a\} \\ &\quad + (1-\lambda) \inf_{F,A} \{\mathrm{E}[F(A(F))] : \mathrm{E}[A(F)] \le b\} && (6)\\ &= \lambda h(a) + (1-\lambda)h(b). \end{aligned}$$

where inequality (6) is because that for every A_a, F_a that satisfies $\mathrm{E}[A(F)] \le a$ and every A_b, F_b that satisfies $\mathrm{E}[A(F)] \le b$, the linear combination $A(F) = \lambda A_a(F_a) + (1-\lambda)A_b(F_b)$ satisfies $\mathrm{E}[A(F)] \le \lambda a + (1-\lambda)b$. 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 $f(x) \le 1-x$ for every $f \in \mathcal{F}$, we have

$$h(\alpha) \le \mathrm{E}[F(A(F))] \le \mathrm{E}[1 - A(F)] \le 1-\alpha.$$

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

□

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

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

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 $D(\mathrm{View}(B_r, \mathcal{M}(x)), \mathrm{View}(B_r, \mathcal{M}(x'))) \le d$. Now let random variable R be uniformly distributed over the coins of A. Then the view of the randomized adversary B when interacting with $\mathcal{M}$ consists of the coins R and the view of the deterministic adversary B_R. That is,

$$\mathrm{View}(B \leftrightarrow \mathcal{M}(x)) = (R, \mathrm{View}(B_R \leftrightarrow \mathcal{M}(x))),$$

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

$$\mathrm{View}(B \leftrightarrow \mathcal{M}(x)) \le d.$$

□

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 variable X, X' and Y, Y', we have $D(X, X') \le d$ and $D(Y, Y') \le d$, 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 $D((X, Y), (X', Y')) \le d$.*

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*

$$T(X, X') \ge f \quad \textit{and} \quad T(Y, Y') \ge f.$$

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

$$T((X, Y)||(X', Y')) \ge f.$$

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. *$T(P, Q) \le T(P', Q')$.*
2. *There exists a randomized algorithm $\mathrm{Proc} : X \to Y$ such that $\mathrm{Proc}(P) = P'$ and $\mathrm{Proc}(Q) = Q'$.*

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 rade-off function f, there exists a pair of random variables P, P' such that $T(P, P') = f$. By the Blackwell Theorem, since $T(P, P') = f \le T(X, X')$, there exists a randomized algorithm $\mathcal{P}_0$ such that $\mathcal{P}_0(P)$ and $\mathcal{P}_0(P')$ are identically distributed to X and X', respectively. Similarly, since $T(P, P') = f \le T(Y, Y')$, there exists a randomized algorithm $\mathcal{P}_1$ such that $\mathcal{P}_1(P), \mathcal{P}_1(P')$ is identically distributed to Y, Y', respectively. We construct a coupling of X and Y as $(\mathcal{P}_0(P), \mathcal{P}_1(P))$, and a coupling of X' and Y' as $(\mathcal{P}_0(P'), \mathcal{P}_1(P'))$. Then the trade-off function between the two couplings satisfies the following inequality.

$$\begin{aligned} & T((\mathcal{P}_0(P), \mathcal{P}_1(P)), (\mathcal{P}_0(P'), \mathcal{P}_1(P'))) \\ &\ge T(P, P') && (7)\\ &= f, && (8) \end{aligned}$$

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 $f : A \to B$ is* continuous *if $f(\sup(S)) = \sup(f(S))$ for every set $S \subseteq A$.*

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

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 $f : A^S \to B$ 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 $(\mathcal{D}, \preceq, D)$ satisfies the chain rule property if for every pair of random variables (X, X') on the same domain $\mathcal{X}$, there is a function that is continuous in each variable:* $\mathrm{ChainRule}_{X,X'} : \mathcal{D}^{\mathrm{supp}(X) \cap \mathrm{supp}(X')} \to \mathcal{D}$ *such that for every pair of random variables Y and Y' where Y is jointly distributed with X and Y' is jointly distributed with X', we have*

$$\begin{aligned} & D((X, Y), (X', Y')) \\ &\quad = \mathrm{ChainRule}_{X,X'}((D(Y|X = x, Y'|X' = x)_{x \in \mathrm{supp}(X) \cap \mathrm{supp}(X')}). \end{aligned}$$

As an example, the standard chain rule of KL divergence is as follows.

$$\begin{aligned} & D_{\mathrm{KL}}((X,Y)||(X',Y')) \\ &\quad = D_{\mathrm{KL}}(X||X') + D_{\mathrm{KL}}(Y|X||Y'|X') \\ &\quad = D_{\mathrm{KL}}(X||X') + \mathrm{E}_{x\sim X} D_{\mathrm{KL}}(Y|X=x||Y'|X'=x). \end{aligned}$$

So fixing X and X', we can calculate the KL divergence for arbitrary Y and Y' as a function of the KL divergences $\mathrm{KL}(Y|X=x||Y'|X'=x)$.

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 $\mathrm{ChainRule}_{X,X'}$ such that for every random variable Y jointly distributed with X, and every random variable Y' jointly distributed with X', we have*

$$\begin{aligned} & T((X,Y),(X',Y')) \\ &\quad = \mathrm{ChainRule}_{X,X'}\left((T(Y|X=x, Y'|X'=x))_{x\in\mathrm{supp}(X)\cap\mathrm{supp}(X')}\right). \end{aligned}$$

where T is a trade-off function.

Proof.

Claim 3.4. *The ChainRule function for f-DP is given as follows.*

$$\begin{aligned} & \mathrm{ChainRule}_{X,X'}\left((f_x)_{x\in\mathrm{supp}(X)\cap\mathrm{supp}(X')}\right)(\alpha) \\ &\quad = \inf_{\alpha_x\in[0,1]} \left\{ \mathrm{E}_{x\sim X'}[f_x(\alpha_x)] : \mathrm{E}_{x\sim X}[\alpha_x] \le \alpha \right\}. \end{aligned} \tag{9}$$

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 ϕ 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 $x \in \mathrm{supp}(X) \cap \mathrm{supp}(X')$, let $\phi_x(y) := \phi(x,y)$. Additionally, let f_x be the trade-off function conditioned on x, i.e.,

$$f_x(\alpha) := T(Y|X=x, Y'|X'=x)(\alpha). \tag{10}$$

The type I error α_ϕ and type II error β_ϕ are given as

$$\alpha_\phi = \mathrm{E}[\phi(x,y)] = \mathrm{E}_{x\sim X}\mathrm{E}_{y\sim Y}[\phi_x(y)],$$

and

$$\beta_\phi = 1 - \mathrm{E}[\phi(x',y')] = 1 - \mathrm{E}_{x\sim X'}\mathrm{E}_{y'\sim Y'}[\phi_{x'}(y')].$$

For every fixed $x \in \mathrm{supp}(X) \cap \mathrm{supp}(X')$ and every decision rule ϕ such that $\mathrm{E}_{y\sim Y}[\phi_x(y)] = \alpha_x$, by the definition of f_x in (10), we have

$$1 - \mathrm{E}_{y'\sim Y'}[\phi_{x'}(y')] \ge f_x(\alpha_x).$$

Therefore, the trade-off function between (X,Y) and (X',Y') satisfies the following inequality:

$$\begin{aligned} & T((X,Y),(X',Y'))(\alpha) \\ &\quad = \inf_\phi \left\{\beta_\phi : \alpha_\phi \le \alpha\right\} \\ &\quad = \inf_\phi \left\{1 - \mathrm{E}_{x\sim X'}\mathrm{E}_{y'\sim Y'}[\phi_{x'}(y')] : \mathrm{E}_{x\sim X}\mathrm{E}_{y\sim Y}[\phi_x(y)] \le \alpha\right\} \\ &\quad \ge \inf_{\{\alpha_x\}} \left\{\mathrm{E}_{x\sim X'}[f_x(\alpha_x)] : \mathrm{E}_{x\sim X}[\alpha_x] \le \alpha\right\}. \end{aligned} \tag{11}$$

On the other hand, by the definition of f_x, for every $0 < \alpha_x < 1$ and $\delta > 0$, there exists a decision rule ϕ^δ such that $1 - \mathrm{E}_{y'\sim Y'}[\phi^\delta_{x'}(y')] \le f_x(\alpha_x) + \delta$ and $\mathrm{E}_{y\sim Y}[\phi^\delta_x(y)] \le \alpha_x$. Then we have

$$\begin{aligned} & \inf_{\{\alpha_x\}} \left\{\mathrm{E}_{x\sim X'}[f_x(\alpha_x)] : \mathrm{E}_{x\sim X}[\alpha_x] \le \alpha\right\} \\ &\quad \ge \inf_\delta \left\{\mathrm{E}_{x\sim X'}[1 - \mathrm{E}_{y'\sim Y'}[\phi^\delta_{x'}(y')] - \delta] : \mathrm{E}_{x\sim X}[\mathrm{E}_{y\sim Y}[\phi^\delta_x(y)]] \le \alpha\right\}. \\ &\quad = \inf_\delta \left\{\beta_{\phi^\delta} : \alpha_{\phi^\delta} \le \alpha\right\} - \delta \\ &\quad \ge \inf_\phi \left\{\beta_\phi : \alpha_\phi \le \alpha\right\} - \delta \\ &\quad = T((X,Y),(X',Y'))(\alpha) - \delta. \end{aligned}$$

Let δ go to 0, and combining with Equation (11), we have

$$T((X,Y),(X',Y'))(\alpha) = \inf\left\{\mathrm{E}_{x\sim X'}[f_x(\alpha_x)] : \mathrm{E}_{x\sim X}[\alpha_x] \le \alpha\right\}. \tag{12}$$

completing the proof for this claim.

Next, we shall show that the ChainRule function defined in (9) is continuous in each variable. Our goal is to show that for every i, every $S_i \subseteq \mathcal{D}$, and for every $f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_n \in \mathcal{D}$, we have

$$\begin{aligned} & \mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, \sup S_i, f_{i+1}, \ldots, f_n) \\ &\quad = \sup(\mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i). \end{aligned}$$

Let $S = (\mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i)$. For every random variable F such that $\mathrm{supp}(F) \subseteq S$, we have F taking values as $\mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i$, so it is equivalent to consider a random variable F_i such that $\mathrm{supp}(F_i) \subseteq S_i$. For every function $A : S \to [0,1]$, we also slightly abuse the notation and use $A(F_i)$ to represent $A(F)$. For every $\alpha \in [0,1]$, we have

$$\begin{aligned} & \sup(\mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, f_i, f_{i+1}, \ldots, f_n) : f_i \in S_i)(\alpha) \\ &= \inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i \\ A:S\to[0,1]}} \left\{\mathrm{E}_{F_i}\left[\mathrm{ChainRule}_{X,X'}(f_1, \ldots, f_{i-1}, F_i, f_{i+1}, \ldots, f_n)(A(F_i))\right]\right. \\ &\quad \left. : \mathrm{E}_{F_i}[A(F_i)] \le \alpha\right\} && \text{(by Lemma 2.4)} \\ &= \inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i \\ A:S\to[0,1]}} \left\{\mathrm{E}_{F_i}\left[\inf_{\substack{A_1,\ldots,A_n \\ A_j:S_i\to[0,1]}} \left\{\mathrm{E}_{j\sim X'}\left[f_j(A_j(F_i)) \cdot \mathbb{I}(j \ne i)\right.\right.\right.\right. \\ &\quad \left.\left.\left.\left. + F_i(A_i(F_i)) \cdot \mathbb{I}(j = i)\right] : \mathrm{E}_{j\sim X}[A_j(F_i)] \le A(F_i)\right\}\right] : \mathrm{E}_{F_i}[A(F_i)] \le \alpha\right\} \\ &&& \text{(by (9))} \\ &= \inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i \\ A:S_i\to[0,1] \\ A_j:S_i\to[0,1]}} \left\{\mathrm{E}_{F_i}\mathrm{E}_{j\sim X'}\left[f_j(A_j(F_i)) \cdot \mathbb{I}(j \ne i) + F_i(A_i(F_i)) \cdot \mathbb{I}(j = i)\right]\right. \\ &\quad \left. : \mathrm{E}_{j\sim X}[A_j(F_i)] \le A(F_i), \mathrm{E}_{F_i}[A(F_i)] \le \alpha\right\}. \end{aligned} \tag{13}$$

We can interchange the expectation and the infimum in Equation (13) because $A_j(f_i)$, $j = 1, \ldots, n$, are independent across $f_i \in F_i$.

We also have that

$$\begin{aligned}
&\mathrm{ChainRule}_{X,X'}(f_1,\dots,f_{i-1},\sup S_i,f_{i+1},\dots,f_n)(\alpha)\\
=&\inf_{\alpha_1,\dots,\alpha_n}\Big\{\mathrm{E}_{j\sim X'}\big[f_j(\alpha_j)\cdot\mathbb{I}(j\neq i)+(\sup S_i)(\alpha_i)\cdot\mathbb{I}(j=i)\big]\\
&:\mathrm{E}_{j\sim X}[\alpha_j]\le\alpha\Big\} \qquad \text{(by (9))}\\
=&\inf_{\alpha_1,\dots,\alpha_n}\left\{\mathrm{E}_{j\sim X'}\left[f_j(\alpha_j)\cdot\mathbb{I}(j\neq i)+\inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i\\ A_i:S_i\to[0,1]}}\Big\{\mathrm{E}_{F_i}\left[F_i(A_i(F_i))\right]\right.\right.\\
&\left.\left.:\mathrm{E}[A_i(F_i)]\le\alpha_i\Big\}\cdot\mathbb{I}(j=i)\right]:\mathrm{E}_{j\sim X}[\alpha_j]\le\alpha\right\} \quad \text{(by Lemma 2.4)}\\
=&\inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i\\ A_i:S_i\to[0,1]\\ \alpha_1,\dots,\alpha_n}}\Big\{\mathrm{E}_{F_i}\mathrm{E}_{j\sim X'}\big[f_j(\alpha_j)\cdot\mathbb{I}(j\neq i)+F_i(A_i(F_i))\cdot\mathbb{I}(j=i)\big]\\
&:\mathrm{E}_{j\sim X}[\alpha_j]\le\alpha,\mathrm{E}_{F_i}[A_i(F_i)]\le\alpha_i\Big\}. \qquad (14)
\end{aligned}$$

The constraints in (13) are equivalent to the constraints in (14), which can be seen as follows. For all $\alpha_j(F_i)$ ($j=1,\dots,k$, and $F_i\in\mathrm{supp}(S_i)$) satisfy the constraints in (13), let $\alpha_j=\mathrm{E}_{F_i}[A_j(F_i)]$, then α_j satisfy the constraints in (14). On the other hand, for all α_j ($j=1,\dots,k$) satisfy the constraints in (14), let $\alpha_j=\mathrm{E}_{F_i}[A_j(F_i)]$, then $\alpha_j(F_i)$ satisfy the constraints in (13). Moreover, with $\alpha_j=\mathrm{E}_{F_i}[A_j(F_i)]$, we have

$$\begin{aligned}
&\mathrm{E}_{j\sim X'}\big[f_j(\alpha_j)\cdot\mathbb{I}(j\neq i)\big]\\
&=\mathrm{E}_{j\sim X'}\big[f_j(\mathrm{E}_{F_i}[A_j(F_i)])\cdot\mathbb{I}(j\neq i)\big]\\
&\le\mathrm{E}_{j\sim X'}\mathrm{E}_{F_i}\big[f_j(A_j(F_i))\cdot\mathbb{I}(j\neq i)\big], \qquad (15)
\end{aligned}$$

where (15) is because trade-off functions are convex. Hence,

$$\begin{aligned}
&\inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i\\ A_i:S_i\to[0,1]\\ \alpha_1,\dots,\alpha_n}}\mathrm{E}_{j\sim X'}\big[f_j(\alpha_j)\cdot\mathbb{I}(j\neq i)\big]\\
\le&\inf_{\substack{F_i:\mathrm{supp}(F_i)\subseteq S_i\\ A:S_i\to[0,1]\\ A_j:S_i\to[0,1]}}\mathrm{E}_{j\sim X'}\mathrm{E}_{F_i}\big[f_j(A_j(F_i))\cdot\mathbb{I}(j\neq i)\big].
\end{aligned}$$

On the other hand, by setting $A_j(F_i)=\alpha_j$ for all $F_i\in S_i$, the above equal sign is reached.

Therefore, we have

$$\begin{aligned}
&\mathrm{ChainRule}_{X,X'}(f_1,\dots,f_{i-1},\sup S_i,f_{i+1},\dots,f_n)\\
&\quad=\sup(\mathrm{ChainRule}_{X,X'}(f_1,\dots,f_{i-1},f_i,f_{i+1},\dots,f_n):f_i\in S_i).
\end{aligned}$$

□

4 CONCURRENT COMPOSITION OF d-$\mathcal{D}$ DP

Theorem 4.1 shows that if the generalized distance satisfies the coupling property in Definition 3.1 and the chain rule in Definition 3.4 , then every interactive d-$\mathcal{D}$ DP mechanism can be simulated by an interactive post-processing of a non-interactive d-$\mathcal{D}$ DP mechanism. This is a generalized statement of Theorem 1.5, as f-DP is an example of d-$\mathcal{D}$ DP.

Theorem 4.1 (Theorem 1.5 generalized). *Assume that the generalized probability distance measure $(\mathcal{D},\preceq,D)$ satisfies*

1. *$(\mathcal{D},\preceq)$ is complete.*
2. *$\mathcal{D}$ satisfies the chain rule.*
3. *every $d\in\mathcal{D}$ satisfies the coupling property.*

Then for every $d\in\mathcal{D}$ and every interactive d-$\mathcal{D}$ DP mechanism $\mathcal{M}$ with finite communication complexity, and every pair of two neighboring datasets x and x', there exists a pair of random variables Y,Y' and an randomized interactive post-processing mechanism $\mathcal{P}$ such that $D(Y,Y')\preceq d$, and for every adversary $B\in\mathcal{B}$, we have

$$\mathrm{View}(B\leftrightarrow\mathcal{M}(x))\equiv\mathrm{View}(B\leftrightarrow\mathcal{P}(Y)) \qquad (16)$$
$$\mathrm{View}(B\leftrightarrow\mathcal{M}(x'))\equiv\mathrm{View}(B\leftrightarrow\mathcal{P}(Y')). \qquad (17)$$

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 $r_A,m_1,\dots,m_{i-1}$, we have*

1. *If $\max\{i,|m_1|,\dots,|m_{i-1}|\}>c$, then*
$$A(x_A,m_1,m_3,\dots,m_{i-1};r_A)=\texttt{halt}.$$
2. *If $\max\{i,|m_1|,\dots,|m_{i-1}|\}\le c$, then*
$$\sum_{j=0}^{i-1}\big|A(x_A,m_1,m_3,\dots,m_j;r_A)\big|\le c.$$

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 $\mathcal{M}$ or the adversary B.

Case 1. The adversary B sends the first query q_1 to the mechanism $\mathcal{M}$. Fixing q_1, the subsequent interactive mechanism $\mathcal{M}_{q_1}$ with input x is defined by

$$\mathcal{M}_{q_1}(x,q_2,\dots,q_m,r)=\mathcal{M}(x,q_1,\dots,q_m,r).$$

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

$$D(Y_{q_1},Y'_{q_1})\preceq d,$$

and

$$\mathcal{P}_{q_1}(Y_{q_1})\equiv\mathcal{M}_{q_1}(x)\quad\mathcal{P}_{q_1}(Y'_{q_1})\equiv\mathcal{M}_{q_1}(x').$$

By coupling property, there exists a pair of random variables Y,Y' and a randomized post-processing function Q_{q_1} such that $D(Y,Y')\preceq d$, and we have that

$$Q_{q_1}(Y)=Y_{q_1},$$
$$Q_{q_1}(Y')=Y'_{q_1}.$$

So Y and Y' are produced by coupling all possible queries. Then the interactive post-processing $\mathcal{P}$ is defined by $\mathcal{P}_{q_1}\circ Q_{q_1}$, i.e.,

$$\mathcal{P}(y,q_1,q_2,\dots,q_m)=\mathcal{P}_{q_1}(Q_{q_1}(y),q_2,\dots,q_m).$$

Case 2. The mechanism $\mathcal{M}$ sends the first message a_1 to the adversary B. Let $q_1, \dots q_{m-1}$ be the queries from the adversary, and $A_1, \dots, A_m$ be messages from the mechanism. Fixing $A_1 = a_1$, the subsequent interactive mechanism $\mathcal{M}_{a_1}$ is defined by

$$\mathcal{M}_{a_1}(x, q_1, \dots, q_{m-1}; g_x(r)) = \mathcal{M}(x, q_1, \dots, q_{m-1}; r).$$

$\mathcal{M}_{a_1}$ uses its randomness to choose uniformly from randomness of $\mathcal{M}$ conditioned on $\mathcal{M}(x) = a_1$. Specifically, let g_x be a random transformation such that if R is uniform random for $\mathcal{M}$, then for all x, $g_x(R)$ is uniform on the randomness of $\mathcal{M}$ conditioned on $\mathcal{M}(x) = a_1$.

We define the subsequent adversary

$$B_{a_1}(A_2, \dots, A_m) = B(a_1, A_2, \dots, A_m).$$

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

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

We have

$$\begin{aligned}
&\sup_B D(\text{View}(B \leftrightarrow \mathcal{M}(x)), \text{View}(B \leftrightarrow \mathcal{M}(x'))) \\
=&\sup_B(\text{ChainRule}_{A_1,A_1'}((D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')})) \qquad (18) \\
=&\sup_{(B_{a_1})_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}} (\text{ChainRule}_{A_1,A_1'}((D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')})) \qquad (19) \\
=&\text{ChainRule}_{A_1,A_1'}(\sup_B(D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}), \qquad (20)
\end{aligned}$$

where (18) follows from the chain rule, and (19) 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 ever 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 (20) follows from that the $\text{ChainRule}_{A_1,A_1'}$ function is continuous with respect to each variable. For every $a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')$, define $d_{a_1} = \sup_{B_{a_1}} D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x')))$. Then $\mathcal{M}_{a_1}$ is d_{a_1}-$\mathcal{D}$ DP, 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

$$\mathcal{P}_{a_1}(Y_{a_1}) \equiv \mathcal{M}_{a_1}(x) \text{ and } \mathcal{P}_{a_1}(Y'_{a_1}) \equiv \mathcal{M}_{a_1}(x').$$

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

$$\begin{aligned}
&D((A_1, Y_{A_1}), (A_1', Y'_{A_1})) \\
&=\text{ChainRule}_{A_1,A_1'}((D(Y_{a_1}, Y'_{a_1}))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}) \\
&\leq\text{ChainRule}_{A_1,A_1'}((d_{a_1})_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}) \\
&\qquad\qquad\qquad\qquad \text{(by monotonicity of ChainRule)} \\
&=\text{ChainRule}_{A_1,A_1'}((\sup_{B_{a_1}} D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\quad\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}) \\
&=\text{ChainRule}_{A_1,A_1'}(\sup_B(D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\quad\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')}) \\
&=\sup_B(\text{ChainRule}_{A_1,A_1'}((D(\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x)), \\
&\quad\text{View}(B_{a_1} \leftrightarrow \mathcal{M}_{a_1}(x'))))_{a_1 \in \text{supp}(A_1) \cap \text{supp}(A_1')})) \\
&\leq d
\end{aligned}$$

Let $Y = (A_1, Y_{A_1})$ and $Y' = (A_1', Y'_{A_1})$, we define the post-processing $\mathcal{P}$ as

$$\mathcal{P}((a_1, y), q_1, \dots, q_{m-1}) = (a_1, \mathcal{P}_{a_1}(y, q_1, \dots, q_{m-1})).$$

□

We now use Theorem 4.1 to prove that the concurrent composition of interactive mechanisms can be reduced to the composition of the non-interactive mechanisms.

Theorem 4.2 (Theorem 1.4 generalized). *Suppose that the generalized probability distance $(\mathcal{D}, \preceq, D)$ satisfies the chain rule, every $d \in \mathcal{D}$ satisfies the coupling property, and $(\mathcal{D}, \preceq)$ is complete. Suppose for all non-interactive mechanism $\mathcal{M}_1, \dots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is d_i-$\mathcal{D}$ DP for $i = 1, 2 \dots, k$, their composition $Comp(\mathcal{M}_1, \dots, \mathcal{M}_k)$ is d-$\mathcal{D}$ DP, then the concurrent composition $ConComp(\mathcal{M}_1, \dots, \mathcal{M}_k)$ of interactive mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is d_i-$\mathcal{D}$ DP is also d-$\mathcal{D}$ DP.*

Proof of Theorem 4.2. Following Theorem 4.1, for every interactive d_j-$\mathcal{D}$ DP mechanism $\mathcal{M}_j$, $j = 1, \dots, k$, and every pair of neighboring datasets x, x', there exists a pair of 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 $B \in \mathcal{B}$, $\text{View}(B \leftrightarrow \mathcal{M}_j(x))$ (resp.,$\text{View}(B \leftrightarrow \mathcal{M}_j(x'))$) is identically distributed as $\text{View}(B \leftrightarrow \mathcal{P}_j(Y_j))$ (resp., $\text{View}(B \leftrightarrow \mathcal{P}_j(Y_j'))$). Since Y_j, Y_j', $j = 1, \dots, k$, are noninteractive random variables, which can be viewed as the output distributions of a noninteractive mechanism $\mathcal{N}_j$ on x, x'. Suppose $\text{Comp}(\mathcal{N}_1, \dots, \mathcal{N}_k)$ is d-$\mathcal{D}$ DP. By the post-processing property, we know that $\text{Comp}((\mathcal{P}_1(\mathcal{N}_1), \dots, \mathcal{P}_k(\mathcal{N}_k))$ is also d-$\mathcal{D}$ DP. Therefore, we have that $\text{ConComp}(\mathcal{M}_1, \dots, \mathcal{M}_k)$ is also d-$\mathcal{D}$ DP.

□

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 $\alpha > 1$, $k \in \mathbb{N}$, $\epsilon_1, \dots, \epsilon_k > 0$, and all interactive mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is (α, ϵ_i)-RDP for*

$i = 1, 2 \ldots, k$, the concurrent composition ConComp$(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ of interactive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ is $(\alpha, \sum_{i=1}^{k} \epsilon_i)$-RDP.

We prove this theorem by characterizing optimal α-RDP adversary strategy in Lemma 5.5.

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

$$\begin{aligned} &D_\alpha(\mathrm{View}(B^{\mathrm{OPT}} \leftrightarrow \mathcal{M}(x)) || \mathrm{View}(B^{\mathrm{OPT}} \leftrightarrow \mathcal{M}(x'))) \\ &\quad \geq D_\alpha(\mathrm{View}(B \leftrightarrow \mathcal{M}(x)) || \mathrm{View}(B \leftrightarrow \mathcal{M}(x'))). \end{aligned}$$

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 $B^{\mathrm{OPT}(1)}, B^{\mathrm{OPT}(2)}, \ldots, B^{\mathrm{OPT}(k)}$ be optimal α-RDP adversaries against $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$. Then*

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

is an optimal α-RDP adversary against ConComp$(\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$, where $B^{\mathrm{OPT}(1)} \times B^{\mathrm{OPT}(2)}$ denotes the adversary's strategy where it takes $B^{\mathrm{OPT}(1)}$ to interact with $\mathcal{M}_1$ and takes $B^{\mathrm{OPT}(2)}$ to interact with $\mathcal{M}_2$.

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*

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

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

$$\begin{aligned} &D_\alpha((U, V) || (U', V')) \\ &= \frac{1}{\alpha - 1} \log \sum_{u,v} \frac{(\Pr(U = u) \Pr(V = v|U = u))^\alpha}{(Pr(U' = u) \Pr(V' = v|U' = u))^{\alpha - 1}} \\ &= \frac{1}{\alpha - 1} \log \sum_{u} \frac{(\Pr(U = u))^\alpha}{(Pr(U' = u))^{\alpha - 1}} \left(\sum_v \frac{(\Pr(V = v|U = u))^\alpha}{(Pr(V' = v|U' = u))^{\alpha - 1}} \right) \\ &= \frac{1}{\alpha - 1} \log \sum_{u} \frac{(\Pr(U = u))^\alpha}{(Pr(U' = u))^{\alpha - 1}} \left(\exp(\alpha - 1) D_\alpha(V|_{U=u} || V'|_{U'=u}) \right). \end{aligned}$$

Hence, $D_\alpha((U, V) || (U', V'))$ is monotonically increasing as $D_\alpha(V|_{U=u} || V'|_{U'=u})$ increases. So if

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

then we have $D_\alpha((U, V) || (U', V')) \leq D_\alpha((U, W) || (U', W'))$.

□

Lemma 5.4. *For any two pairs of random variables U, U' and V, V', if U and V (U' and V' resp.) are independent, then*

$$D_\alpha((U, V) || (U', V')) = D_\alpha(U || U') + D_\alpha(V || V').$$

Proof.

$$\begin{aligned} &D_\alpha((U, V) || (U', V')) \\ &\quad = \frac{1}{\alpha - 1} \log \sum_{u,v} \frac{(\Pr(U = u) \Pr(V = v))^\alpha}{(Pr(U' = u) \Pr(V' = v))^{\alpha - 1}} \\ &\quad = \frac{1}{\alpha - 1} \log \sum_{u} \frac{(\Pr(U = u))^\alpha}{(Pr(U' = u))^{\alpha - 1}} \left(\exp(\alpha - 1) D_\alpha(V || V') \right) \\ &\quad = D_\alpha(U || U') + D_\alpha(V || V'). \end{aligned}$$

□

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^{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*

$$D_\alpha \left(\left(A_1, \mathrm{View} \left(B^{\mathrm{OPT}}_{q_1, A_1} \leftrightarrow M_{q_1, A_1}(x) \right) \right) || \left(A_1', \mathrm{View} \left(B^{\mathrm{OPT}}_{q_1, A_1'} \leftrightarrow M_{q_1, A_1'}(x') \right) \right) \right),$$

where $A_1 = \mathcal{M}(x, q_1)$ and $A_1' = \mathcal{M}(x', q_1)$, and $B^{\mathrm{OPT}}_{q_1, a_1}$ is any optimal adversary against $\mathcal{M}_{q_1, a_1}$, 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

$$\begin{aligned} &D_\alpha(\mathrm{View}(B \leftrightarrow \mathcal{M}(x)) || \mathrm{View}(B \leftrightarrow \mathcal{M}(x'))) \\ &\quad = D_\alpha \left(\left(A_1(q_1), \mathrm{View} \left(B_{q_1, A_1} \leftrightarrow \mathcal{M}_{q_1, A_1}(x) \right) \right) || \right. \\ &\qquad \left. \left(A_1'(q_1), \mathrm{View} \left(B_{q_1, A_1'} \leftrightarrow \mathcal{M}_{q_1, A_1'}(x') \right) \right) \right), \end{aligned} \tag{21}$$

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

$$\begin{aligned} &D_\alpha \left(\mathrm{View}(B_{q_1, A_1} \leftrightarrow \mathcal{M}_{q_1, A_1}(x))|_{A_1 = a_1} || \mathrm{View}(B_{q_1, A_1'} \leftrightarrow \mathcal{M}_{q_1, A_1'}(x'))|_{A_1' = a_1} \right) \\ &\leq D_\alpha \left(\mathrm{View}(B^{\mathrm{OPT}}_{q_1, A_1} \leftrightarrow \mathcal{M}_{q_1, A_1}(x))|_{A_1 = a_1} || \mathrm{View}(B^{\mathrm{OPT}}_{q_1, A_1'} \leftrightarrow \mathcal{M}_{q_1, A_1'}(x'))|_{A_1' = a_1} \right). \end{aligned}$$

Thus, we have

$$\begin{aligned}
&D_\alpha\left(\left(A_1(q_1), \mathrm{View}\left(B_{q_1,A_1} \leftrightarrow \mathcal{M}_{q_1,A_1}(x)\right)\right) ||\right.\\
&\left.\left(A_1'(q_1), \mathrm{View}\left(B_{q_1,A_1'} \leftrightarrow \mathcal{M}_{q_1,A_1'}(x')\right)\right)\right)\\
\le &D_\alpha\left(\left(A_1(q_1), \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1} \leftrightarrow \mathcal{M}_{q_1,A_1}(x)\right)\right) ||\right.\\
&\left.\left(A_1'(q_1), \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1'} \leftrightarrow \mathcal{M}_{q_1,A_1'}(x')\right)\right)\right), \qquad (22)
\end{aligned}$$

where (22) follows from Lemma 5.3. It implies that in order to maxmize (21), it suffices to choose q_1 to maximize the quantity in (22). □

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 $\mathcal{M}_1$, and we use $\mathrm{ConComp} = \mathrm{ConComp}(\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$ to simplify the notation. Following Lemma 5.5, the optimal adversary B^{OPT} chooses q_1 as follows.

$$\begin{aligned}
B^{\mathrm{OPT}} =&\mathrm{argmax}_{q_1} D_\alpha\left(\left(A_1, \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1} \leftrightarrow \mathrm{ConComp}_{q_1,A_1}(x)\right)\right) ||\right.\\
&\left.\left(A_1', \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1'} \leftrightarrow \mathrm{ConComp}_{q_1,A_1'}(x')\right)\right)\right),
\end{aligned}$$

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

$$V_{q_1,A_1} = \mathrm{View}\left(B^{\mathrm{OPT}(1)}_{q_1,A_1} \leftrightarrow \mathcal{M}_{q_1,A_1}(x)\right),$$

and similarly,

$$V'_{q_1,A_1} = \mathrm{View}\left(B^{\mathrm{OPT}(1)}_{q_1,A_1'} \leftrightarrow \mathcal{M}_{q_1,A_1'}(x')\right).$$

Let

$$\tilde{V} = \mathrm{View}\left(B^{\mathrm{OPT}(2)} \times \ldots \times B^{\mathrm{OPT}(k)} \leftrightarrow \mathrm{ConComp}(\mathcal{M}_2, \ldots, \mathcal{M}_k)(x)\right),$$

and

$$\tilde{V}' = \mathrm{View}\left(B^{\mathrm{OPT}(2)} \times \ldots \times B^{\mathrm{OPT}(k)} \leftrightarrow \mathrm{ConComp}(\mathcal{M}_2, \ldots, \mathcal{M}_k)(x')\right).$$

With these notations, we have

$$\begin{aligned}
&\mathrm{argmax}_{q_1} D_\alpha\left(\left(A_1, \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1} \leftrightarrow \mathrm{ConComp}_{q_1,A_1}(x)\right)\right) ||\right.\\
&\left.\left(A_1', \mathrm{View}\left(B^{\mathrm{OPT}}_{q_1,A_1'} \leftrightarrow \mathrm{ConComp}_{q_1,A_1'}(x')\right)\right)\right)\\
&\quad=\mathrm{argmax}_{q_1} D_\alpha\left(\left(A_1, V_{q_1,A_1}, \tilde{V}\right) || \left(A_1', V'_{q_1,A_1}, \tilde{V}'\right)\right)\\
&\quad=\mathrm{argmax}_{q_1}\left\{D_\alpha\left(\left(A_1, V_{q_1,A_1}\right) || \left(A_1', V'_{q_1,A_1}\right)\right) + D_\alpha(\tilde{V}||\tilde{V}')\right\},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(by Lemma 5.4)}\\
&\quad=\mathrm{argmax}_{q_1} D_\alpha\left(\left(A_1, V_{q_1,A_1}\right) || \left(A_1', V'_{q_1,A_1}\right)\right)\\
&\quad=B^{\mathrm{OPT}(1)}. \qquad (23)
\end{aligned}$$

Therefore, the optimal adversary B^{OPT} chooses q_1 just as the optimal adversary $B^{\mathrm{OPT}(1)}$ against only M_1. Since $B^{\mathrm{OPT}}_{q_1,A_1} = B^{\mathrm{OPT}(1)}_{q_1,A_1} \times B^{\mathrm{OPT}(2)} \times \ldots \times B^{\mathrm{OPT}(k)}$, we have $B^{\mathrm{OPT}} = B^{\mathrm{OPT}(1)} \times B^{\mathrm{OPT}(2)} \times \ldots \times B^{\mathrm{OPT}(k)}$, completing the proof. □

We now prove Theorem 5.1 using Lemma 5.2.

Proof of Theorem 5.1. Following Lemma 5.2, we have

$$\begin{aligned}
&D_\alpha\left(\mathrm{View}\left(B^{\mathrm{OPT}} \leftrightarrow \mathrm{ConComp}(x)\right) ||\mathrm{View}\left(B^{\mathrm{OPT}} \leftrightarrow \mathrm{ConComp}(x')\right)\right)\\
&\;=D_\alpha\left(\mathrm{View}\left(B^{\mathrm{OPT}(1)} \leftrightarrow \mathcal{M}_1(x)\right) ||\mathrm{View}\left(B^{\mathrm{OPT}(1)} \leftrightarrow \mathcal{M}_1(x')\right)\right) + \ldots\\
&\quad+ D_\alpha\left(\mathrm{View}\left(B^{\mathrm{OPT}(k)} \leftrightarrow \mathcal{M}_k(x)\right) ||\mathrm{View}\left(B^{\mathrm{OPT}(k)} \leftrightarrow \mathcal{M}_k(x')\right)\right)\\
&\;=\sum_i^k \epsilon_i,
\end{aligned}$$

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.

[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, Toniann 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:1603.01887* (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] Alfréd 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 Tianhao 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