entropy

MDPI

Article

Two-Party Zero-Error Function Computation with Asymmetric Priors †

Basak Guler [1], Aylin Yener [1,*] Prithwish Basu [2] and Ananthram Swami [3]

1 The Pennsylvania State University, University Park, PA 16802, USA; basak@psu.edu
2 Raytheon BBN Technologies, Cambridge, MA 02138, USA; prithwish.basu@raytheon.com
3 Army Research Laboratory, Adelphi, MD 20783, USA; a.swami@ieee.org
* Correspondence: yener@ee.psu.edu; Tel.: +1-814-865-4337

† Earlier versions of this work have partially appeared at the IEEE GlobalSIP Symposium on Network Theory, December 2013, IEEE Data Compression Conference (DCC'14), March 2014, and IEEE Data Compression Conference (DCC'16), March 2016.

Received: 11 July 2017; Accepted: 13 November 2017; Published: 23 November 2017

Abstract: We consider a two party network where each party wishes to compute a function of two correlated sources. Each source is observed by one of the parties. The true joint distribution of the sources is known to one party. The other party, on the other hand, assumes a distribution for which the set of source pairs that have a positive probability is only a subset of those that may appear in the true distribution. In that sense, this party has only partial information about the true distribution from which the sources are generated. We study the impact of this asymmetry on the worst-case message length for zero-error function computation, by identifying the conditions under which reconciling the missing information prior to communication is better than not reconciling it but instead using an interactive protocol that ensures zero-error communication without reconciliation. Accordingly, we provide upper and lower bounds on the minimum worst-case message length for the communication strategies with and without reconciliation. Through specializing the proposed model to certain distribution classes, we show that partially reconciling the true distribution by allowing a certain degree of ambiguity can perform better than the strategies with perfect reconciliation as well as strategies that do not start with an explicit reconciliation step. As such, our results demonstrate a tradeoff between the reconciliation and communication rates, and that the worst-case message length is a result of the interplay between the two factors.

Keywords: data compression; function computation; partial information; characteristic graphs

1. Introduction

Consider a scenario in which two parties make a query over distributed correlated databases. Each party observes data from one database, whereas the query has to be evaluated over the data observed by both users separately. Suppose that one party knows all data combinations that may lead to an answer to some query, whereas the other party is missing some of these combinations. The parties are allowed to communicate with each other. The goal is to find the minimum amount of communication required so that both parties can retrieve the correct answer for any query. We model this scenario as interactive communication in which two parties interact to compute a function of two correlated discrete memoryless sources. Each source is observed by one party. One party knows the true joint distribution of the sources, whereas the other party is missing some source pairs that may occur with positive probability and assumes another distribution in which these missing pairs have zero probability. Communication takes place in multiple interactive rounds, at the end of which a function of the two correlated sources has to be computed at both parties with zero-error. We study the impact of this partial knowledge about the true distribution on the worst-case message length.

In a function computation scenario, one party observes a random variable X, whereas the other party observes a random variable Y, where each realization of (X, Y) is generated from some probability distribution p_{XY}. The two parties wish to compute a function $f(X, Y)$ by exchanging a number of messages in multiple rounds. Conventionally, the true distribution from which the sources are generated is available as common knowledge to both parties. This work extends this framework to the scenario in which the true distribution of the sources is available at one of the communicating parties only, while the distribution assumed at the other party has missing information compared to the true distribution. That is, the second party has only partial knowledge about the source pairs that are realized with positive probability according to the true distribution.

In order to identify the impact of partial information on the worst-case message length, we consider three interactive communication protocols. The first interactive protocol we consider is to reconcile the partial information between the two parties in a way to allow the second party to learn the true joint distribution, and then utilize the true distribution for function computation. The reconciliation stage transforms the problem into the conventional zero-error function computation problem with zero-error. Although this is a natural approach in that it ensures that both sides are in agreement about the true distribution, this protocol requires additional bits to be transmitted between the two parties for reconciling the distribution information, which, in turn may increase the overall message length. The second protocol we consider provides an alternative interaction strategy in which the two parties do not reconcile the true distribution, but instead use a function computation strategy that allows error-free computation under the distribution uncertainty. In doing so, this protocol alleviates the costs that may have incurred for reconciling the distributions. The message length for the function computation part, however, may be larger compared to that of the previous scheme. The last interaction protocol quantifies a trade-off between the two interaction protocols, by allowing the two parties to partially reconcile the distributions. In this protocol, each party learns the true distribution up to a class of distributions. The function computation step then ensures error-free computation under any distribution within the reconciled class of distributions. By doing so, we create different levels of common knowledge about the distribution to investigate the relation between the cost of various degrees of partial reconciliation and the resulting compression performance.

By leveraging the proposed interaction protocols, we identify the conditions under which it is better or worse to reconcile the partial information than to not reconcile the distributions, i.e., using a zero-error encoding scheme with possibly increased message length. Accordingly, we develop upper and lower bounds on the worst-case zero-error message length for computing the function at both parties under different reconciliation and communication strategies. Our results demonstrate that, reconciling the partial information, although often reducing the communication cost, may or may not reduce the overall worst-case message length. In effect, the worst-case message length results from an interplay between reconciliation and communication costs. As such, partial reconciliation of the true distribution is sometimes strictly better than the remaining two interaction strategies.

Related Work

For the setting when both parties know the true joint distribution of the sources, interactive communication strategies have been studied in [1] to enable both sides to learn the source observed by the other party with zero-error. Reference [2] has considered the impact of the number of interaction rounds on the worst-case message length, as well as upper and lower bounds on the worst-case message length. The optimal zero-error communication strategy for minimizing the worst-case message length, even for the setting in which the communicating parties know the exact true distribution of the sources, has since been an open problem. The zero-error communication problem has also been considered for communicating semantic information [3,4]. Our work is also related to the field of communication complexity, which studies the minimum amount of communication required to compute a function of two sources [5]. Known as the direct-sum theorem, it was shown in [6] that computing multiple instances of a function can reduce the minimum amount of communication required per instance.

The main distinction between the communication-complexity approaches and the setups from [1,2] is that the models from [1,2] emphasize utilizing the source distribution and in particular its support set to reduce the amount of communication, which is also referred to as the computation of a partial function ([7], Section 4.7).

In addition to the zero-error setup, interactive communication has also been considered for computing a function at one of the communicating parties with vanishing error probability [8]. Subsequently, interactive communication has been considered for computing a function of two sources simultaneously at both parties with vanishing error probability [9]. The two-party scenario has been extended to a multi-terminal function computation setup in [10], in which each party observes an independent source and broadcasts its message to all the nodes in the network. A related study in [11] investigates the role of side information when communicating interactively a source known by one party to another with vanishing error. Interactive communication has also been leveraged in [11] for one-way recovery of a source known by one party at the other side with vanishing error in the presence of side information.

This work is also related to zero-error communication strategies in non-interactive data compression scenarios. In particular, we leverage graphical representations of the confusable source and distribution terms, which are reminiscent of characteristic graphs introduced in [12] to study the zero-error capacity of a channel. Subsequently, characteristic graphs have been utilized for zero-error compression of a source in the presence of decoder side information [13,14]. They have been utilized to characterize graph entropy and chromatic entropy in [15,16], respectively, in [8] to characterize the rate region for the lossless computation of a function, and in [17] to obtain achievable rates for lossy function computation. Such graphical representations have also been leveraged for non-interactive set reconciliation [18]. Another relevant application is zero-error source coding with compound decoder side information considered in [19].

Many existing and emerging network applications, e.g., sensor networks, cyber-physical systems, social media, and semantic networks, facilitate interaction between multiple terminals to share information towards achieving a common objective [20–23]. As such, it is essential for such systems to mitigate the ambiguities that may result from the imperfect knowledge available at the communicating parties. The case when the communicating parties assume different prior distributions while communicating a source from one party to another has recently been considered for the non-interactive setting. In [24], communicating a source with vanishing error is considered in the presence of side information when the joint probability distributions assumed at the encoder and the decoder are different. Reference [25] has incorporated shared randomness to facilitate compression when the source distribution assumed by the two parties are different from each other. Deterministic compression strategies are investigated in [26] for the case when no shared randomness is present. In this work, we study interactive function computation with partial priors for the asymmetric scenario when the true joint distribution of the sources is available at one party only [27].

2. Problem Setup

This section introduces our two-party communication setup with asymmetric priors. The following notation is adopted in the sequel. We use $\mathcal{X}$ for a set with cardinality $|\mathcal{X}|$, and define $x^n = (x_1, \ldots, x_n)$ where $x^1 = x$ [28]. The difference between defining a sequence $x^n = (x_1, \ldots, x_n)$ vs. taking the n^{th} power of a given number will be clear from context. We denote $\{0,1\}^* = \cup_{n=1}^{\infty} \{0,1\}^n$. The support set of a distribution $p(x,y)$ over a set $\mathcal{X} \times \mathcal{Y}$ is represented as,

$$\text{supp}(p) \triangleq \{(x,y) \in \mathcal{X} \times \mathcal{Y} : p(x,y) > 0\}, \tag{1}$$

where

$$\text{supp}(p^n) = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : p(x_i, y_i) > 0 \text{ for } i = 1, \ldots, n\}. \tag{2}$$

The chromatic number of a graph G is given by $\chi(G)$. $\ell(\cdot)$ represents the length (number of bits) of a bit stream. Finally, for a bipartite graph $G = (V, U, E)$ with vertex sets V, U and an edge set E, we let Δ_X and Δ_Y denote the maximum degree of any node $v \in V$ and $u \in U$, respectively.

2.1. System Model

Consider discrete memoryless correlated sources (X, Y) defined over a finite set $\mathcal{X} \times \mathcal{Y}$. The sources are generated from a distribution $p(x, y) \in \mathcal{P}$ where $\mathcal{P}$ is a finite set of probability distributions. Nodes 1 and 2 observe $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$, respectively, with probability $p^n(x^n, y^n) = \prod_{i=1}^n p(x_i, y_i)$. The distribution $p(x, y)$ is fixed over the course of n time instants. We refer to $p(x, y)$ as the *true distribution* of sources (X, Y) as it represents nature's selection for the distribution of sources (X, Y). User 1 knows the true distribution $p(x, y)$. The source distribution known to user 2, however, may be different from the true distribution. In particular, user 2 assumes a distribution $q(x, y) \in \mathcal{Q}$ such that $\text{supp}(q) \subseteq \text{supp}(p)$ where $\mathcal{Q}$ is a finite set. The set of distributions $\mathcal{P}$ is known by both users, but the actual selections for $p(x, y)$ and $q(x, y)$ are only known at the corresponding user. In that sense, $q(x, y)$ provides some, although incomplete, information to user 2 about $p(x, y)$.

Each of the two parties is requested to compute a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{F}$ for each term of the source sequence (X^n, Y^n), which we represent as

$$f^n(x^n, y^n) \triangleq (f(x_1, y_1), \ldots, f(x_n, y_n)), \tag{3}$$

where $\mathcal{F}$ is a finite set. In particular, user 1 recovers some $Z_1^n \in \mathcal{F}^n$ whereas user 2 recovers some $Z_2^n \in \mathcal{F}^n$ such that *zero-error* probability condition

$$\Pr[f^n(X^n, Y^n) \neq Z_1^n] = Pr[f^n(X^n, Y^n) \neq Z_2^n] = 0, \tag{4}$$

is satisfied, which is evaluated over the true distribution $p(x, y)$. Note that, whenever $f(x, y)$ is a bijective function, Equation (4) reduces to the conventional zero-error interactive data compression where each source symbol is perfectly recovered at the other party [1].

The two users employ an interactive communication protocol, in which they send binary strings called *messages* at each round. A *codeword* represents a sequence of messages exchanged by the two users in multiple rounds. In particular, for an r round communication, the encoding function is given by some variable-length scheme $\phi : \mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}^*$ for which the codeword $\phi(x^n, y^n) = (\phi_1(x^n, y^n), \ldots, \phi_r(x^n, y^n))$ is the sequence of messages exchanged for the pair $(x^n, y^n) \in \mathcal{S}^n$, where $\phi_i(x^n, y^n)$ represents the message transmitted by both parties at round i and $\phi^i(x^n, y^n) = (\phi_1(x^n, y^n), \ldots, \phi_i(x^n, y^n))$ denotes the sequence of messages exchanged through the first i rounds for $i \in \{1, \ldots, r\}$. The encoding at each round is based only on the symbols known to the user and on the messages exchanged between the two users in the previous rounds, so that

$$\phi_i(x^n, y^n) = (\phi_i^X(x^n, \phi^{i-1}(x^n, y^n)), \phi_i^Y(y^n, \phi^{i-1}(x^n, y^n))), \tag{5}$$

where $\phi_i^X(x^n, \phi^{i-1}(x^n, y^n)) \in \{0,1\}^*$ and $\phi_i^Y(y^n, \phi^{i-1}(x^n, y^n)) \in \{0,1\}^*$ are the messages transmitted from users 1 and 2 at round i, respectively. The encoding protocol is deterministic and agreed upon by both parties in advance. Accordingly, we define

$$\phi^X(x^n, y^n) = (\phi_1^X(x^n), \ldots, \phi_r^X(x^n, \phi^{r-1}(x^n, y^n))), \tag{6}$$

and

$$\phi^Y(x^n, y^n) = (\phi_1^Y(y^n), \ldots, \phi_r^Y(y^n, \phi^{r-1}(x^n, y^n))), \tag{7}$$

as the sequences of messages transmitted from users 1 and 2, respectively, in r rounds. Another condition is the prefix-free message property to ensure that whenever one user sends a message, the other

user knows when the message ends. This necessitates that for all $(x^n, y^n), (x^n, \hat{y}^n) \in \text{supp}(p^n)$, then $\phi^{i-1}(x^n, y^n) = \phi^{i-1}(x^n, \hat{y}^n)$ for some $i \in \{2, \cdots, r\}$ requires that $\phi_i^Y(y^n, \phi^{i-1}(x^n, y^n))$ is not a proper prefix of $\phi_i^Y(\hat{y}^n, \phi^{i-1}(x^n, \hat{y}^n))$. Same applies for user 1 when we interchange the roles of X and Y. In addition, we require the coordinated termination criterion to ensure both parties know when communication ends. In particular, given some $(x^n, y^n), (x^n, \hat{y}^n) \in \text{supp}(p^n)$, we require that $\phi(x^n, y^n)$ is not a proper prefix of $\phi(x^n, \hat{y}^n)$. Same condition applies when the roles of X and Y are interchanged. The last condition we require is the unique message property. In particular, if $(x^n, y^n), (x^n, \hat{y}^n) \in \text{supp}(p^n)$, then $\phi^{i-1}(x^n, y^n) = \phi^{i-1}(x^n, \hat{y}^n)$ implies that $\phi_i^X(x^n, \phi^{i-1}(x^n, y^n)) = \phi_i^X(x^n, \phi^{i-1}(x^n, \hat{y}^n))$. The same applies when the roles of X and Y are changed. Null transmissions are allowed at any round.

The worst-case codeword length for mapping ϕ is given by

$$l_\phi^{(n)} = \max_{(x^n, y^n) \in \text{supp}(p^n)} \frac{1}{n} \ell(\phi(x^n, y^n)) \quad \text{bits/symbol.} \tag{8}$$

where $\ell(\cdot)$ is the number of bits in a bit stream. The optimal worst-case codeword length is given by

$$l^{(n)} = \min_{\phi} \; l_\phi^{(n)}. \tag{9}$$

The zero-error condition in Equation (4) ensures that, for any given function, the worst-case codeword length of the optimal communication protocol is the same for all distributions in $\mathcal{P}$, i.e., for any $p, p' \in \mathcal{P}$, as long as $\text{supp}(p) = \text{supp}(p')$. We utilize this property for designing interactive protocols by constructing graphical structures as described next. It is useful to note that the results throughout the paper hold when the parties only know the support of the distributions $p(x, y)$ and $q(x, y)$ in the problem set up considered in this paper as described next. For each $p(x, y) \in \mathcal{P}$, we define a bipartite graph $G_p = (\mathcal{X}, \mathcal{Y}, E_p)$ with vertex sets $\mathcal{X}$, $\mathcal{Y}$, and an edge set E_p. An edge $(x, y) \in E_p$ exists if and only if $p(x, y) > 0$.

Observe that we have $G_p = G_{p'}$ for any $p(x, y), p'(x, y) \in \mathcal{P}$ with $\text{supp}(p) = \text{supp}(p')$. One can therefore partition $\mathcal{P}$ into groups of distributions that have the same support set, such that the set of distributions in each partition maps to a unique bipartite graph. We represent this set of resulting bipartite graphs by $\mathcal{G}$, and denote each element $G \in \mathcal{G}$ by $G = (\mathcal{X}, \mathcal{Y}, E_G)$. The bipartite graph structure used for partitioning the distributions in $\mathcal{P}$ is related to the notion of ergodic decomposition from [29], in that each bipartite graph represents a class of distributions with the same ergodic decomposition. For each $G \in \mathcal{G}$, we denote

$$\mathcal{S}_G^n = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : (x_i, y_i) \in E_G, \; i = 1, \ldots, n\}, \tag{10}$$

and note that for any distribution $p(x, y) \in \mathcal{P}$ whose support set can be represented by the bipartite graph G, one has $\mathcal{S}_G^n = \text{supp}(p^n)$.

Given $G \in \mathcal{G}$, we define the following sets. For each $x^n \in \mathcal{X}^n$, we define an ambiguity set

$$\mathcal{I}_{X,G}(x^n) = \{f^n(x^n, y^n) \in \mathcal{F}^n : (x_i, y_i) \in E_G, \; y_i \in \mathcal{Y}, \; i = 1, \ldots, n\}, \tag{11}$$

where each element is a sequence of function values, and $\lambda_G(x^n) \triangleq |\mathcal{I}_{X,G}(x^n)|$ denotes the number of distinct sequences of function values. Similarly, for each $y^n \in \mathcal{Y}^n$, we define an ambiguity set

$$\mathcal{I}_{Y,G}(y^n) = \{f^n(x^n, y^n) \in \mathcal{F}^n : (x_i, y_i) \in E_G, \; x_i \in \mathcal{X}, \; i = 1, \ldots, n\}, \tag{12}$$

with $\mu_G(y^n) \triangleq |\mathcal{I}_{Y,G}(y^n)|$. Next, we let

$$\lambda_G \triangleq \max_{x \in \mathcal{X}} \lambda_G(x), \tag{13}$$

and note that $\max_{x^n \in \mathcal{X}^n} \lambda_G(x^n) = (\lambda_G)^n$. Similarly, we define

$$\mu_G \triangleq \max_{y \in \mathcal{Y}} \mu_G(y), \tag{14}$$

and note that $\max_{y^n \in \mathcal{Y}^n} \mu_G(y^n) = (\mu_G)^n$. We denote the maximum vertex degrees for graph G by

$$\Delta_X \triangleq \max_{x \in \mathcal{X}} |\{y \in \mathcal{Y} : (x,y) \in E_G\}|, \quad \Delta_Y \triangleq \max_{y \in \mathcal{Y}} |\{x \in \mathcal{X} : (x,y) \in E_G\}|. \tag{15}$$

Lastly, using Equations (11) and (12), for each $(x^n, y^n) \in \mathcal{S}_G^n$ we define

$$\mathcal{I}_G(x^n, y^n) = \mathcal{I}_{X,G}(x^n) \cup \mathcal{I}_{Y,G}(y^n), \tag{16}$$

An illustrative example of the bipartite graph is given in Figure 1 for the function $f(x,y) = (x+y) \bmod 4$ and the probability distribution

$$p(x,y) = \begin{cases} \frac{1}{|\mathcal{X}|+2} & \text{if } x = 1 \text{ or } y = 3 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

over the finite set $\mathcal{X} = \{1, \ldots, 5\}$ and $\mathcal{Y} = \{1, \ldots, 3\}$.

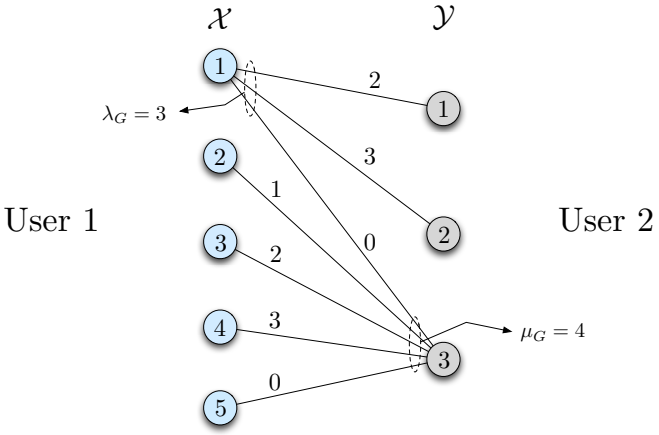


Figure 1. Bipartite graph representation of the probability distribution from Equation (17). Edge labels represent the function values $f(x,y) = (x+y) \bmod 4$. Note that the maximum vertex degree is $\Delta_X = 3$ for $x \in \mathcal{X}$ and $\Delta_Y = 5$ for $y \in \mathcal{Y}$ whereas $\lambda_G = 3$ and $\mu_G = 4$.

Finally, we review a basic property of zero-error interactive protocols, which is key to our analysis in the sequel. The straightforward proof immediately follows, e.g., from ([1], Lemma 1, Corollary 2).

Proposition 1. *Let $[\phi_k(x^n, y^n)]_{k=1}^r$ be the concatenation of all $\phi_k(x^n, y^n)$ for $k = 1, \ldots, r$. Then, for each $(x^n, y^n) \in \mathcal{S}_G^n$, the set of sequences corresponding to the symbols in $\mathcal{I}_G(x^n, y^n)$ should be prefix-free.*

Proof. The proof follows from the following observation. Suppose for some $(x^n, y^n) \in \mathcal{S}_G^n$, we have $(\hat{x}^n, y^n), (x^n, \hat{y}^n) \in \mathcal{S}_G^n$ where $[\phi_k(\hat{x}^n, y^n)]_{k=1}^r$ is a prefix of $[\phi_k(x^n, \hat{y}^n)]_{k=1}^r$. Then, from ([1], Lemma 1), we have $\phi(\hat{x}^n, y^n) = \phi(x^n, y^n) = \phi(x^n, \hat{y}^n)$. Now, if $f^n(x^n, y^n) \neq f^n(x^n, \hat{y}^n)$, then user 1 will not be able to distinguish between the two function values as the message sequences are the same for both. Similarly, if $f^n(x^n, y^n) \neq f^n(\hat{x}^n, y^n)$, then user 2 will not be able to distinguish between the two function values. Hence, $[\phi_k(\hat{x}^n, y^n)]_{k=1}^r$ cannot be a prefix of $[\phi_k(x^n, \hat{y}^n)]_{k=1}^r$ whenever $f^n(\hat{x}^n, y^n) \neq f^n(x^n, \hat{y}^n)$. From the same argument, $[\phi_k(x^n, y^n)]_{k=1}^r$ cannot be a prefix of $[\phi_k(x^n, \hat{y}^n)]_{k=1}^r$ whenever

$f(x^n, y^n) \neq f(x^n, \hat{y}^n)$ otherwise user 1 will not be able to recover the correct function value. The same applies to user 2 when the roles of X and Y are changed. Therefore, for any given $(x^n, y^n) \in \mathcal{S}_G^n$, we need at least $|\mathcal{I}_G(x^n, y^n)|$ prefix-free sequences, one for each element of $\mathcal{I}_G(x^n, y^n)$. Otherwise, one of the above three cases will occur and at least one user will not be able to distinguish the correct function value. □

2.2. Motivating Example

Consider two interacting users, user 1 observing $x \in \mathcal{X} = \{1, \ldots, 7\}$ and user 2 observing $y \in \mathcal{Y} = \{1, \ldots, 7\}$ according to the distribution

$$p(x,y) = \begin{cases} 1/5 & \text{if}(x,y) \in \{(3,1),(3,2),(3,5),(6,5),(7,5)\} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where both users want to compute a function of $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$f(x,y) = \begin{cases} 0 & \text{if } x - y > 0 \\ 1 & \text{if } -1 \leq x - y \leq 0 \\ 2 & \text{otherwise} \end{cases} . \tag{19}$$

First, assume that users 1 and 2 both know the distribution $p(x, y)$; we will call this the *symmetric priors* case. In this case, one can readily observe from Equations (18) and (19) that the function value $f(x, y) = 1$ will never occur, hence the two parties can discard that value beforehand. That is, in this case users 1 and 2 know beforehand that they only need to distinguish between two function values, $f(x, y) = 0$, which occurs when $(x, y) \in \{(3,1),(3,2),(6,5),(7,5)\}$, and $f(x, y) = 2$, which occurs when $(x, y) = (3, 5)$. We now detail five interaction protocols as follows. The first one is a naïve protocol where user 1 sends x to user 2, and user 2 sends y to user 1, after which both users can compute $f(x, y)$. To do so, users 1 and 2 need $\lceil \log 7 \rceil = 3$ bits each, i.e., a total of 6 bits is needed. Second, consider a protocol in which user 1 sends x to user 2, and user 2 calculates $f(x, y)$ and sends the result back to user 1. To do so, user 1 needs to use $\lceil \log 7 \rceil = 3$ bits. User 2 on the other hand needs to send only $\log 2 = 1$ bit, since there are at most 2 possible function values. This protocol uses 4 bits in total in two rounds. Same applies to the third protocol where we exchange the roles of users 1 and 2. Since users 1 and 2 know the support set of $p(x, y)$, i.e., the pairs of (x, y) for which $p(x, y) > 0$, a fourth protocol would involve sending only $\lceil \log 3 \rceil + \lceil \log 3 \rceil = 4$ bits in total, in which user 1 sends one of $x \in \{3, 6, 7\}$, whereas user 2 sends one of $y \in \{1, 2, 5\}$. Lastly, consider a different protocol where user 1 sends "0" if $x \in \{6, 7\}$, and a "1" otherwise, which is sufficient for user 2 to infer whether $f(x, y) = 0$ or $f(x, y) = 2$ depending on the y he observes, since $f(x, y) = 1$ is not possible with these (x, y) values. Therefore, user 2 computes $f(x, y)$ and sends the result back to user 1 by using $\log 2 = 1$ bit. This protocol requires only $\log 2 + \log 2 = 2$ bits in two rounds and at the end both users learn $f(x, y)$. As is clear from this example, communicating all distinct pairs of symbols is not always the best strategy, and resources can be saved by using a more efficient strategy.

Next, consider the following variation on the example. Users 1 and 2 again wish to compute $f(x, y)$ given in Equation (19), but this time the joint distribution of the sources $p(x, y)$ is selected from a set of distributions $\mathcal{P} = \{p_1, p_2, p_3\}$ where $p_1(x, y)$ is defined as in Equation (18), and we have

$$p_2(x,y) = \begin{cases} 1/7 \text{ if } (x,y) \in \{(3,2),(3,3),(3,4),(3,5),(4,5),(5,5),(6,5)\} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{20}$$

and

$$p_3(x,y) = \begin{cases} 1/5 & \text{if } (x,y) \in \{(3,1),(3,3),(3,5),(4,5),(7,5)\} \\ 0 & \text{otherwise} \end{cases} . \tag{21}$$

As described in the beginning of Section 2.1, one can represent the structure of these distributions and the corresponding function values via bipartite graphs. Such a bipartite graph for the probability distribution $p_2(x,y)$ in Equation (20) is given in Figure 2.

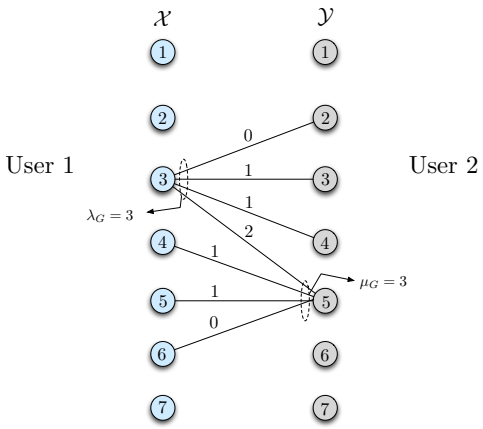


Figure 2. Shared bipartite graph with $n = 1$. $\mathcal{X} = \mathcal{Y} = \{1, \ldots, 7\}$ representing the distribution $p_2(x,y)$ from Equation (20). Edge labels represent the function values $f(x,y)$ defined in Equation (19). Maximum vertex degrees are $\Delta_X = \Delta_Y = 4$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ whereas $\lambda_G = \mu_G = 3$.

User 1 observes $p(x,y)$, i.e., the true distribution. User 2 knows the set $\mathcal{P}$, but not the specific choice in $\mathcal{P}$. User 2 instead observes a distribution $q(x,y)$ from a set $\mathcal{Q} = \{q_1, q_2\}$.

$$q_1(x,y) = \begin{cases} 1/3 & \text{if } (x,y) \in \{(3,2),(3,5),(6,5)\} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

and

$$q_2(x,y) = \begin{cases} 1/3 & \text{if } (x,y) \in \{(3,1),(3,5),(7,5)\} \\ 0 & \text{otherwise} \end{cases} . \tag{23}$$

User 1 does not know the distribution $q(x,y)$ observed at user 2. In addition, the set $\mathcal{Q}$ is unknown to both users. The only requirement we have is that this distribution be *consistent* with $p(x,y)$. That is to say that the support of $q(x,y)$ is contained in the support of $p(x,y)$, i.e., $q(x,y)$ does not have a positive probability for a source pair whose probability is zero in $p(x,y)$, i.e., $\text{supp}(q) \subseteq \text{supp}(p)$. Note that this is side information in that users 1 and 2 can *infer* which of the $q(x,y)$ or $p(x,y)$ distributions are possible at the other party, respectively, given their own distribution.

In order to interact in this setup, users 1 and 2 may initially agree to reconcile the distribution and then use it as in the previous case. To do so, user 1 informs user 2 of the true distribution. She assigns an index "0" if $p = p_1$, and a "1" if $p \in \{p_2, p_3\}$, and sends it to user 2 by using $\log 2 = 1$ bit. User 2 can infer the true distribution by using the received index as well as his own distribution q. If the received index is "0", then it immediately follows that the true distribution is p_1. However, if the received index is "1", then user 2 needs to decide between p_2 and p_3. To do so, he utilizes q: (i) whenever $q = q_1$, he declares that the true distribution is p_2, since $\text{supp}(q_1) \nsubseteq \text{supp}(p_3)$, (ii) whenever $q = q_2$, he decides that the true distribution is p_3, since in this case $\text{supp}(q_2) \nsubseteq \text{supp}(p_2)$. After this step, both users know the true distribution, and can compute $f(x,y)$ by exchanging no more than a total number of $\lceil \log 3 \rceil + \lceil \log 3 \rceil = 4$ bits, as detailed next. The case where $p_1(x,y)$ is the true distribution requires 2 bits for interaction as noted earlier. If the true distribution is $p_2(x,y)$, user 1 can send user 2 an index "0" if $x \in \{6\}$, a "1" if $x \in \{4,5\}$, or a "2" otherwise. User 2 can compute $f(x,y)$ and send the result back to user 1 by using at most $\lceil \log 3 \rceil = 2$ bits, since in the worst-case all three function values may occur, which happens when $x = 3$. Therefore, this case requires 4 bits for communication. If instead the true distribution is $p_3(x,y)$, user 1 can send user 2 an index "0" if $x \in \{7\}$, a "1" if $x \in \{4\}$,

or a "2" otherwise. User 2 can compute $f(x,y)$ and send it back to user 1 by using $\lceil \log 3 \rceil = 2$ bits, since all three function values are again possible for $x = 3$. Hence, this scheme requires 5 bits to be communicated in total, 1 bit for reconciliation and 4 bits for communication.

An alternative scheme is one in which users 1 and 2 do not reconcile the true distribution, but instead use an encoding scheme that allows error-free communication under any distribution uncertainty. To do so, user 1 sends an index "0" if $x \in \{6,7\}$, a "1" if $x \in \{4,5\}$, and a "2" otherwise. Describing 3 indices requires user 1 to use $\lceil \log 3 \rceil = 2$ bits. After receiving the index value, user 2 can recover $f(x,y)$ perfectly, whether the true distribution p is equal to p_1, p_2, or p_3, and then send it to user 1 by using no more than $\lceil \log 3 \rceil = 2$ bits, since there are at most 3 distinct values of $f(x,y)$ for each $y \in \mathcal{Y}$. Both users can then learn $f(x,y)$. Not reconciling the partial information therefore takes 4 bits, which is less then the previous two stage reconciliation-communication protocol.

3. Communication Strategies with Asymmetric Priors

In this section, we propose three strategies for zero-error communication by mitigating the ambiguities resulting from the partial information about the true distribution.

3.1. Perfect Reconciliation

For the communication model described in Section 2.1, a natural approach to tackle the partial information is by first sending the missing information to user 2 so that both sides know the source pairs that may be realized with positive probability with respect to the true distribution, which can then be utilized for communication. This setup consists of two stages. In the first stage, user 2 learns the support set of the true distribution $p(x,y)$, or equally the bipartite graph G corresponding to $p(x,y)$, from user 1. We call this the reconciliation stage. After this stage, both parties use graph G for zero-error interactive communication. We refer to this two-stage protocol as *perfect reconciliation* in the sequel. The worst-case message length under this setup is referred to as $l_R^{(n)}$.

For the reconciliation stage, we first partition $\mathcal{Q}$ into groups of distributions with distinct support sets, and denote by $\mathcal{B}$ the set of distinct bipartite graphs that correspond to the support sets of the distributions in $\mathcal{Q}$. This process is similar to the one described for $\mathcal{P}$ in Section 2.1. Next, we find a lower bound for the minimum number of bits required for user 2 to learn the graph G, i.e., all (x,y) pairs that may occur with positive probability under the true distribution $p(x,y)$.

Definition 1. *(Reconciliation graph) Define a characteristic graph $R = (\mathcal{G}, E_R)$, in which each vertex represents a graph $G \in \mathcal{G}$. Recall that $\mathcal{G}$ is a set of bipartite graphs as we define in Section 2.1. An edge $(G, G') \in E_R$ is defined between vertices G and G' if and only if there exists a $B \in \mathcal{B}$ such that $E_B \subseteq E_G$ and $E_B \subseteq E_{G'}$.*

The minimum number of bits required for user 2 to perfectly learn G is then $\lceil \log \chi(R) \rceil$, where $\chi(\cdot)$ denotes the chromatic number of a graph. This can be observed by noting that in the reconciliation phase, any two nodes in the reconciliation graph with an edge in between has to be assigned to distinct bit streams, otherwise user 2 will not be able to distinguish them, which requires a minimum of $\lceil \log \chi(R) \rceil$ number of bits to be transmitted from user 1 to user 2. It is useful to note that perfect reconciliation incurs a negligible cost for large blocklengths.

Proposition 2. *Perfect reconciliation is an asymptotically optimal strategy.*

Proof. Since the distributions $p(x,y)$ and $q(x,y)$ are fixed once chosen, reconciliation requires at most $\lceil \log |R| \rceil$ bits for any class of graphs $\mathcal{G}$. Therefore its contribution on the codeword length per symbol is $\frac{1}{n} \lceil \log |R| \rceil$, which vanishes as $n \to \infty$. Since the communication cost for not reconciling the graphs can never be lower than reconciling them, we can conclude that reconciling the graphs first, and then using the reconciled graphs for communication, cannot perform worse than not reconciling them. We note, however, that this statement may no longer hold if the joint distribution is arbitrarily

varying over the course of n symbols, since correct recovery in this case may require the graphs to be repeatedly reconciled. □

In the following, we demonstrate a lower bound on the worst-case message length for this two-stage reconciliation-communication protocol.

Lemma 1. *A lower bound on the worst-case message length for the two-stage reconciliation-communication protocol is,*

$$l_R^{(n)} \geq \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil. \tag{24}$$

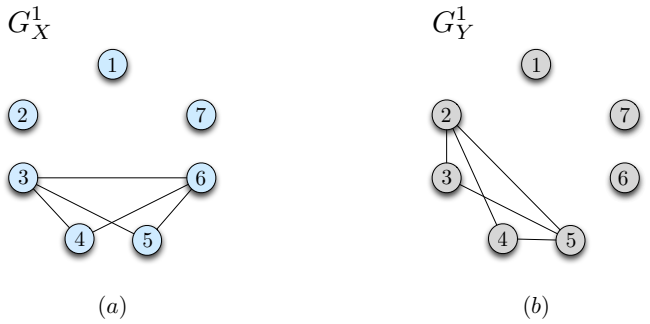
Proof. We prove Equation (24) by obtaining a lower bound on the message length for the reconciliation and communication parts separately. The lower bound for the reconciliation part is determined by bounding the minimum number of bits to be transmitted from user 1 to user 2 using Definition 1. As a result, both sides learn the support set of the true distribution $p(x, y)$. The lower bound in Equation (24) then follows from

$$\begin{aligned}
l_R^{(n)} &\geq \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \min_{\phi} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \ell(\phi(x^n, y^n)) && (25)\\
&\geq \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \min_{\phi} \frac{1}{n} \ell(\phi(x^n, y^n)) && (26)\\
&= \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \min_{\phi} \frac{1}{n} \ell([\phi_k(x^n, y^n)]_{k=1}^r) && (27)\\
&\geq \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil && (28)
\end{aligned}$$

where Equation (26) follows from the min-max inequality and Equation (28) from Proposition 1. □

We next demonstrate an upper bound for the minimum worst-case message length. Consider the distribution $p(x, y)$ and the corresponding bipartite graph $G \in \mathcal{G}$. Let $G_X^n = (\mathcal{X}^n, E_X^n)$ denote a characteristic graph for user 1 with a vertex set $\mathcal{X}^n$. Vertices of G_X^n are the n-tuples $x^n \in \mathcal{X}^n$. An edge $(x^n, \hat{x}^n) \in E_X^n$ exists between $x^n \in \mathcal{X}^n$ and $\hat{x}^n \in \mathcal{X}^n$ whenever some $y^n \in \mathcal{Y}$ exists such that $(x^n, y^n) \in \mathcal{S}_G^n$, $(\hat{x}^n, y^n) \in \mathcal{S}_G^n$ and $f^n(x^n, y^n) \neq f^n(\hat{x}^n, y^n)$. Similarly, define a characteristic graph $G_Y^n = (\mathcal{Y}^n, E_Y^n)$ for user 2 whose vertices are the n-tuples $y^n \in \mathcal{Y}^n$. An edge $(y^n, \hat{y}^n) \in E_Y^n$ exists between $y^n \in \mathcal{Y}^n$ and $\hat{y}^n \in \mathcal{Y}^n$ whenever some $x^n \in \mathcal{X}^n$ exists such that $(x^n, y^n) \in \mathcal{S}_G^n$, $(x^n, \hat{y}^n) \in \mathcal{S}_G^n$, and $f^n(x^n, y^n) \neq f^n(x^n, \hat{y}^n)$.

The characteristic graphs defined above are useful in that any valid coloring over the characteristic graphs will enable the two parties to resolve the ambiguities in distinguishing the correct function values. Figure 3 illustrates the characteristic graphs G_X^1 and G_Y^1, respectively, constructed by using $p_2(x, y)$ from Equation (20) and $f(x, y)$ from Equation (19) in the example discussed in Section 2.2. In the following, we follow the notation $G_X \triangleq G_X^1$ and $G_Y \triangleq G_Y^1$.



(*a*) (*b*)

Figure 3. Characteristic graphs (**a**) G_X^1 and (**b**) G_Y^1 constructed using distribution $p_2(x,y)$ in Equation (20) and function $f(x,y)$ in Equation (19).

Theorem 1. *The worst-case message length for the two-stage separate reconciliation and communication strategy satisfies*

$$l_R^{(n)} \leq \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \frac{1}{n} \left\{ \lceil n \log(\chi(G_X)) \rceil + \lceil n \log(\chi(G_Y)) \rceil \right\}, \tag{29}$$

Proof. Consider a minimum coloring for G_X and G_Y using $\chi(G_X)$ and $\chi(G_Y)$ colors. Note that G_X^n and G_Y^n can be colored with at most $(\chi(G_X))^n$ and $(\chi(G_Y))^n$ colors, respectively. Hence, users 1 and 2 can simultaneously send the index of the color assigned to their symbols by using at most $\lceil n \log \chi(G_X) \rceil$ and $\lceil n \log \chi(G_Y) \rceil$ bits, respectively. Then, users can utilize the received color index and their own symbols for correct recovery of the function values. □

3.2. Protocols that Do Not Explicitly Start with a Reconciliation Procedure

Instead of the reconciliation-based strategy described in Section 3.1, the two users may choose not to reconcile the distributions, but instead utilize a robust communication strategy that ensures zero-error communication under any distribution in set $\mathcal{P}$. Specifically, they can agree on a worst-case communication strategy that always ensures zero-error communication for both users. In this section, we study two specific protocols that do not explicitly start with a reconciliation procedure. We denote the worst case message length in this setting as $l_{RF}^{(n)}$.

As an example of such a robust communication strategy, consider a scenario in which user 1 enumerates each $x^n \in \mathcal{X}^n$ by using $n \log |\mathcal{X}|$ bits whereas user 2 enumerates each $y^n \in \mathcal{Y}^n$ by using $n \log |\mathcal{Y}|$ bits. Then, by using no more than $n \log |\mathcal{X}| + n \log |\mathcal{Y}|$ bits in total, the two parties can communicate their observed symbols with zero-error under any true distribution, and evaluate $f^n(X^n, Y^n)$. In that sense, this setup does not require any additional bits for learning about the distribution from the other side either perfectly or partially, but the message length for communicating the symbols is often higher. In the following, we derive an upper bound on the worst-case message length based on two achievable protocols that do not start with a reconciliation procedure.

The first achievable strategy we consider is based on graph coloring. Let $G_{X,\mathcal{G}} = (\mathcal{X}, E_X)$ be a characteristic graph for user 1 whose vertex set is $\mathcal{X}$. Define an edge $(x, \hat{x}) \in E_X$ between nodes $x \in \mathcal{X}$ and $\hat{x} \in \mathcal{X}$ whenever there exists some $y \in \mathcal{Y}$ such that $(x,y) \in \bigcup_{p \in \mathcal{P}} \text{supp}(p)$ and $(\hat{x},y) \in \bigcup_{p \in \mathcal{P}} \text{supp}(p)$ whereas $f(x,y) \neq f(\hat{x},y)$. Similarly, define a characteristic graph $G_{Y,\mathcal{G}} = (\mathcal{Y}, E_Y)$ for user 2 whose vertex set is $\mathcal{Y}$. Define an edge $(y, \hat{y}) \in E_Y$ between vertices $y \in \mathcal{Y}$ and $\hat{y} \in \mathcal{Y}$ whenever there exists some $x \in \mathcal{X}$ such that $(x,y) \in \text{supp}(p)$ and $(x,\hat{y}) \in \text{supp}(p)$ for some $p \in \mathcal{P}$ but $f(x,y) \neq f(x,\hat{y})$. We note the difference between the conditions for constructing $G_{X,\mathcal{G}}$ and $G_{Y,\mathcal{G}}$ in that the former is based on the union $\bigcup_{p \in \mathcal{P}}$ whereas the latter is based on the existence for some $p \in \mathcal{G}$. This difference results from the fact that user 2 does not know the true distribution, hence needs to distinguish the possible symbols from a group of distributions, whereas user 1 has the true distribution, and can utilize it for eliminating the ambiguities for correct function recovery. We note however that

both $G_{X,\mathcal{G}}$ and $G_{Y,\mathcal{G}}$ depend on $\mathcal{G}$. Lastly, we let $\chi(G_{X,\mathcal{G}})$ and $\chi(G_{Y,\mathcal{G}})$ denote the chromatic number of $G_{X,\mathcal{G}}$ and $G_{Y,\mathcal{G}}$, respectively.

Then, under any true distribution $p \in \mathcal{P}$, the following communication protocol ensures zero error. Suppose user 1 observes x^n and user 2 observes y^n from some distribution $p^n(x^n, y^n) = \prod_{i=1}^n p(x_i, y_i)$. For each $x_i \in \mathcal{X}$ where $i = 1 \ldots, n$, user 1 sends the color of x_i by using no more than $\lceil \log \chi(G_{X,\mathcal{G}}) \rceil$ bits. After this step, user 2 can recover $f_i(x_i, y_i)$ by using y_i as follows. Given y_i, user 2 considers the set of all $x_i \in \mathcal{X}$ such that $p(x_i, y_i) > 0$ for some $p \in \mathcal{P}$. Note that within this set, each color represents a group of $x_i \in \mathcal{X}$ for which $f_i(x_i, y_i)$ is equal. Therefore, under any true distribution $p \in \mathcal{P}$, user 2 will be able to recover the correct $f_i(x_i, y_i)$ value solely by using the received color along with y_i. Similarly, for each $y_i \in \mathcal{Y}$, user 2 sends the color of y_i by using no more than $\lceil \log \chi(G_{Y,\mathcal{G}}) \rceil$ bits, after which user 1 recovers $f_i(x_i, y_i)$ by using the received color and the true distribution $p(x, y)$. Since user 1 knows the true distribution, it can distinguish any function value correctly as long as no two $y, y' \in \mathcal{Y}$ are assigned to the same codeword for which $\exists x \in \mathcal{X}$ such that $(x, y) \in \text{supp}(p)$ and $(x, y') \in \text{supp}(p)$ when $f(x, y) \neq f(x, y')$.

We then have the following upper bound on the worst-case message length,

$$l_{RF}^{(n)} \leq \frac{1}{n} \left(n \lceil \log \chi(G_{X,\mathcal{G}}) \rceil + n \lceil \log \chi(G_{Y,\mathcal{G}}) \rceil \right) = \lceil \log \chi(G_{X,\mathcal{G}}) \rceil + \lceil \log \chi(G_{Y,\mathcal{G}}) \rceil \text{ bits/symbol}, \tag{30}$$

where user 1 sends $n \lceil \log(\chi(G_{X,\mathcal{G}})) \rceil$ bits to user 2 whereas user 2 sends $n \lceil \log(\chi(G_{Y,\mathcal{G}})) \rceil$ bits to user 1. After this step, both users can recover the correct function values $f^n(x^n, y^n)$ for any source pair $(x^n, y^n) \in \text{supp}(p^n)$ under any $p \in \mathcal{P}$.

The second achievable strategy we consider is based on perfect hash functions. A function $h : \{1, \ldots, N\} \to \{1, \ldots, k\}$ is called a *perfect hash function* for a set $\mathcal{S} \subseteq \{1, \ldots, N\}$ if for all $x, y \in \mathcal{S}$ such that $x \neq y$, one has $h(x) \neq h(y)$. Define a family of functions $\mathcal{H}$ such that $h : \{1, \ldots, N\} \to \{1, \ldots, k\}$ for all $h \in \mathcal{H}$. If

$$M \geq s(\ln N) e^{s^2/k} \tag{31}$$

for some $k \geq s$, then, there exists a family of $|\mathcal{H}| = M$ functions such that for every $\mathcal{S} \subseteq \{1, \ldots, N\}$ with $|\mathcal{S}| \leq s$, there exists a function $h \in \mathcal{H}$ that is injective (one-to-one) over $\mathcal{S}$ ([30], Section III.2.3). Perfect hash functions have been proved to be useful for constructing zero-error interactive communication protocols when the true distribution of the sources are known by both parties [7]. In the following, we extend the interactive communication framework from [7] to the setting when the true distribution is unknown by the communicating parties.

Initially, we construct a graph $\bar{G}^{(n)} = (V, E)$ for user 2 with a vertex set $V = \mathcal{Y}^n$. In that sense, each vertex of the graph is an n-tuple $y^n \in \mathcal{Y}^n$. Define an edge $(y^n, \hat{y}^n) \in E$ between vertices $y^n \in \mathcal{Y}^n$ and $\hat{y}^n \in \mathcal{Y}^n$ if for some n-tuple $x^n \in \mathcal{X}^n$ that there exists some $p \in \mathcal{P}$ for which $(x_i, y_i) \in \text{supp}(p)$ and $(x_i, \hat{y}_i) \in \text{supp}(p)$ for all $i = 1, \ldots, n$. Define a minimum coloring of this graph and let $\chi(\bar{G}^{(n)})$ denote the minimum number of required colors, i.e., the chromatic number of $\bar{G}^{(n)}$. In that sense, any valid coloring over this graph will enable user 1 to resolve the ambiguities in distinguishing the correct n-tuple observed by user 2, under any true distribution $p \in \mathcal{P}$.

We next define the following ambiguity set for each $x^n \in \mathcal{X}^n$,

$$\mathcal{I}_X(x^n) \triangleq \{ y^n \in \mathcal{Y}^n : (x_i, y_i) \in \bigcup_{p \in \mathcal{P}} \text{supp}(p) \text{ for } i = 1, \ldots, n \}, \tag{32}$$

as the set of distinct y^n sequences that may occur with respect to the support set $\bigcup_{p \in \mathcal{P}} \text{supp}(p)$ under the given sequence x^n. We denote the size of the largest single-term ambiguity set as,

$$\lambda \triangleq \max_{x \in \mathcal{X}} |\mathcal{I}_X(x)|, \tag{33}$$

and note that $\max_{x^n \in \mathcal{X}^n} |\mathcal{I}_X(x^n)| = \lambda^n$. Lastly, we define an ambiguity set for each $y^n \in \mathcal{Y}^n$,

$$\mathcal{I}_Y(y^n) \triangleq \{f^n(x^n, y^n) \in \mathcal{F}^n : x_i \in \mathcal{X} \text{ and } (x_i, y_i) \in \bigcup_{p \in \mathcal{P}} \text{supp}(p) \text{ for } i = 1, \ldots, n\}, \tag{34}$$

as the set of distinct function values that may appear for the given sequence y^n and with respect to the support set $\bigcup_{p \in \mathcal{P}} \text{supp}(p)$. We denote the size of the largest single-term ambiguity set as

$$\mu \triangleq \max_{y \in \mathcal{Y}} |\mathcal{I}_Y(y)|, \tag{35}$$

and note that $\max_{y^n \in \mathcal{Y}^n} |\mathcal{I}_Y(y^n)| \leq \mu^n$.

The interaction protocol is then given as follows. From Equation (31), there exists a family $\mathcal{H}$ of

$$|\mathcal{H}| = \lceil \lambda^n (\log \chi(\bar{G}^{(n)}))e \rceil \tag{36}$$

functions such that $h : \{1, \ldots, \chi(\bar{G}^{(n)})\} \to \{1, \ldots, \lambda^{2n}\}$ for all $h \in \mathcal{H}$ and for each $\mathcal{S} \subseteq \{1, \ldots, \chi(\bar{G}^{(n)})\}$ of size $|\mathcal{S}| \leq \lambda^n$, there exists an $h \in \mathcal{H}$ that is injective over $\mathcal{S}$. In that sense, the colors assigned to an ambiguity set $\mathcal{I}_X(x^n)$ for some $x^n \in \mathcal{X}^n$ will correspond to some $\mathcal{S}$. Both users initially agree on such a family of functions $\mathcal{H}$ and a minimum coloring of graph $\bar{G}^{(n)}$ with $\chi(\bar{G}^{(n)})$ colors. Suppose user 1 observes $x^n \in \mathcal{X}^n$ and user 2 observes $y^n \in \mathcal{Y}^n$. User 1 finds a function $h \in \mathcal{H}$ that is injective over the colors assigned to vertices $y^n \in \mathcal{I}_X(x^n)$ from Equation (32) and sends its index to user 2 by using no more than

$$\lceil \log |\mathcal{H}| \rceil = \lceil \log \lceil \lambda^n (\log \chi(\bar{G}^{(n)}))e \rceil \rceil \tag{37}$$

bits in total. After this step, user 2 evaluates the corresponding function for the assigned color of y^n and sends the evaluated value back to user 1 by using no more than $\lceil \log \lambda^{2n} \rceil$ bits. After this step, user 1 will learn the color of y^n, from which it can recover y^n by using the observed x^n. This is due to the fact that from the definition of an ambiguity set $\mathcal{I}_X(x^n)$ in Equation (32), every n-tuple $y^n \in \mathcal{I}_X(x^n)$ for a given $x^n \in \mathcal{X}^n$ will receive a different color in the minimum coloring of the graph $\bar{G}^{(n)}$. Since the selected perfect hash function is one-to-one over the colors assigned to $y^n \in \mathcal{I}_X(x^n)$, it will allow user 1 to recover the color of y^n from the evaluated hash function value. In the last step, user 1 evaluates the function $f^n(x^n, y^n)$, and sends it to user 2 by using no more than $\lceil \log \mu^n \rceil$ bits. In doing so, she assigns a distinct index for each sequence of function values in the ambiguity set $\mathcal{I}_Y(y^n)$ from Equation (34). User 2 can then recover the function $f^n(x^n, y^n)$ by using y^n and the received index. Overall, this protocol requires no more than

$$\lceil \log \lceil \lambda^n (\log \chi(\bar{G}^{(n)}))e \rceil \rceil + \lceil 2n \log \lambda \rceil + \lceil n \log \mu \rceil \tag{38}$$

bits to be transmitted in total, therefore

$$\begin{aligned}
l_{NR}^{(n)} &\leq \frac{1}{n} \left(\lceil \log \lceil \lambda^n (\log \chi(\bar{G}^{(n)}))e \rceil \rceil + \lceil 2n \log \lambda \rceil + \lceil n \log \mu \rceil \right) && (39)\\
&\leq \frac{1}{n} \left(\lceil \log (\lambda^n (\log \chi(\bar{G}^{(n)}))e + 1) \rceil + 2n \log \lambda + 1 + n \log \mu + 1 \right) && (40)\\
&\leq \frac{1}{n} \left(\log (\lambda^n (\log \chi(\bar{G}^{(n)}))e + 1) + 2n \log \lambda + n \log \mu + 3 \right) && (41)\\
&\leq \frac{1}{n} \left(\log (\lambda^n (\log \chi(\bar{G}^{(n)}))e) + \log \left(1 + \frac{1}{\lambda^n (\log \chi(\bar{G}^{(n)}))e} \right) + 2n \log \lambda + n \log \mu + 3 \right) && (42)\\
&\leq 3 \log \lambda + \log \mu + \frac{1}{n} \log \log \chi(\bar{G}^{(n)}) + \frac{4 + \log e}{n} && (43)\\
&\leq 3 \log \lambda + \log \mu + \frac{1}{n} \log n \log \chi(\bar{G}^{(1)}) + \frac{4 + \log e}{n} && (44)
\end{aligned}$$

$$\leq 3\log\lambda + \log\mu + \frac{1}{n}\log\log\chi(\bar{G}^{(1)}) + \frac{\log n}{n} + \frac{4+\log e}{n} \tag{45}$$

where Equation (42) follows from the fact that $\frac{1}{\lambda^n(\log\chi(\bar{G}^{(n)}))e} \leq 1$, and Equation (44) holds since $\chi(\bar{G}^{(n)}) \leq \chi^n(\bar{G}^{(1)})$. This is due to the fact that any coloring over the nth order strong product of $\bar{G}^{(1)}$ is also a valid coloring for $\bar{G}^{(n)}$, since by construction of $\bar{G}^{(n)}$, any edge that exists in $\bar{G}^{(n)}$ also exists in the nth order strong product of $\bar{G}^{(1)}$. Therefore, the chromatic number of $\bar{G}^{(n)}$ is no greater than the chromatic number of the nth order product of $\bar{G}^{(1)}$, which is no greater than $\chi^n(\bar{G}^{(1)})$.

Combining the bounds obtained from the two protocols from Equations (30) and (45), we have the following upper bound on the worst-case message length.

Proposition 3. *The worst-case message length for the two strategies that do not explicitly start with a reconciliation procedure can be upper bounded as,*

$$l_{RF}^{(n)} \leq \min\left\{\lceil\log\chi(G_{X,\mathcal{G}})\rceil + \lceil\log\chi(G_{Y,\mathcal{G}})\rceil, 3\log\lambda + \log\mu + \tfrac{1}{n}\log\log\chi(\bar{G}^{(1)}) + \tfrac{\log n}{n} + \tfrac{4+\log e}{n}\right\}. \tag{46}$$

Proof. The result follows from combining the two interaction strategies in Equations (30) and (45). □

3.3. Partial Reconciliation

In order to understand the impact of *level of reconciliation* on the worst-case message length, we consider a third scheme called *partial reconciliation*, which allows user 2 to distinguish the true distribution up to a class of distributions, after which the two users use a robust worst-case communication protocol that allows for zero-error communication in the presence of any distribution within the class. In that sense, partial reconciliation allows some ambiguity in the reconciled set of distributions. Accordingly, the schemes considered in Sections 3.1 and 3.2 are special cases of the partial reconciliation scheme. We denote $l_{PR}^{(n)}$ as the per-symbol worst-case message length for a finite block of n source symbols under the partial reconciliation scheme. In the following, we demonstrate two protocols for interactive communication with partial reconciliation. The first protocol is based on coloring characteristic graphs, whereas the second protocol is based on perfect hash functions. We then derive an upper bound on the worst-case message length with partial reconciliation.

For the first partial reconciliation protocol, consider the set $\mathcal{G}$ of bipartite graphs $G = (\mathcal{X}, \mathcal{Y}, E_G)$ constructed by using the distributions $p \in \mathcal{P}$ as described in Section 2.1. Define a partition of the set $\mathcal{G}$ as $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{|\mathcal{A}|}\}$ such that $\bigcup_{i=1}^{|\mathcal{A}|} \mathcal{A}_i = \mathcal{G}$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for all $i \neq j$, where $\mathcal{A}_i$ is non-empty for $i \in \{1, \ldots, |\mathcal{A}|\}$. Define $\bar{\mathcal{A}}$ as the set of all such partitions of $\mathcal{G}$.

Fix a partition $\mathcal{A} \in \bar{\mathcal{A}}$. For each $i \in \{1, \ldots, |\mathcal{A}|\}$, define a graph $G_{X,\mathcal{A}_i} = (\mathcal{X}, E_X)$ for user 1 with the vertex set $\mathcal{X}$. Define an edge $(x, \hat{x}) \in E_X$ between nodes $x \in \mathcal{X}$ and $\hat{x} \in \mathcal{X}$ if there exists some $y \in \mathcal{Y}$ such that $(x, y) \in \bigcup_{G \in \mathcal{A}_i} G$ and $(\hat{x}, y) \in \bigcup_{G \in \mathcal{A}_i} G$ whereas $f(x, y) \neq f(\hat{x}, y)$. Next, construct a graph $G_{Y,\mathcal{A}_i} = (\mathcal{Y}, E_Y)$ for user 2 with the vertex set $\mathcal{Y}$. Define an edge $(y, \hat{y})$ between nodes $y \in \mathcal{Y}$ and $\hat{y} \in \mathcal{Y}$ if there exists some $x \in \mathcal{X}$ such that $(x, y) \in E_G$ and $(x, \hat{y}) \in E_G$ for some $G \in \mathcal{A}_i$ but $f(x, y) \neq f(x, \hat{y})$. Let $\chi(G_{X,\mathcal{A}_i})$ and $\chi(G_{Y,\mathcal{A}_i})$ denote the chromatic number of $G_{X,\mathcal{A}_i}$ and $G_{Y,\mathcal{A}_i}$, respectively.

Then, under any true distribution $p \in \mathcal{P}$, the following communication protocol ensures zero error. The two users agree on a partition $\mathcal{A} \in \bar{\mathcal{A}}$ before communication starts. Suppose users 1 and 2 observe x^n and y^n, respectively, under the true distribution $p(x, y)$. Let $G = (\mathcal{X}, \mathcal{Y}, E_G)$ denote the bipartite graph corresponding to the distribution $p(x, y)$. Initially, user 1 sends the index i of the set $\mathcal{A}_i \in \mathcal{A}$ for which $G \in \mathcal{A}_i$, by using no more than $\lceil\log|\mathcal{A}|\rceil$ bits. After this step, user 1 sends the color of each symbol in x^n according to the minimum coloring of graph $G_{X,\mathcal{A}_i}$ by using no more than $n\lceil\log\chi(G_{X,\mathcal{A}_i})\rceil$ bits in total. By using the sequence of colors received from user 1, user 2 can determine the correct function values $f^n(x^n, y^n)$. In the last step, user 2 sends the color of each symbol in y^n

according to graph $G_{Y,\mathcal{A}_i}$ by using no more than $n\lceil \log \chi(G_{Y,\mathcal{A}_i})\rceil$ bits. After this step, user 1 can recover the function values $f^n(x^n, y^n)$. Overall, this protocol requires no more than

$$\lceil \log |\mathcal{A}| \rceil + \max_{i\in\{1,\ldots,|\mathcal{A}|\}} n\left(\lceil \log \chi(G_{X,\mathcal{A}_i})\rceil + \lceil \log \chi(G_{Y,\mathcal{A}_i})\rceil\right), \tag{47}$$

bits to be transmitted. Since one can leverage any partition within $\bar{\mathcal{A}}$ for constructing the communication protocol, we conclude that the worst-case message length for partial reconciliation is bounded above by,

$$l_{PR}^{(n)} \leq \min_{\mathcal{A}\in\bar{\mathcal{A}}} \left(\frac{1}{n}\lceil \log |\mathcal{A}| \rceil + \max_{i\in\{1,\ldots,|\mathcal{A}|\}} \left(\lceil \log \chi(G_{X,\mathcal{A}_i})\rceil + \lceil \log \chi(G_{Y,\mathcal{A}_i})\rceil\right) \right). \tag{48}$$

For the second partial reconciliation protocol, we again leverage perfect hash functions from Equation (31). As in the first protocol, we define a partition of the set $\mathcal{G}$ as $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{|\mathcal{A}|}\}$ such that $\bigcup_{i=1}^{|\mathcal{A}|} \mathcal{A}_i = \mathcal{G}$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for all $i \neq j$. We let $\bar{\mathcal{A}}$ be the set of all such partitions of $\mathcal{G}$.

We fix a partition $\mathcal{A} \in \bar{\mathcal{A}}$ of $\mathcal{G}$. For each $i \in \{1, \ldots, |\mathcal{A}|\}$, we define a graph $\bar{G}_i^{(n)} = (\mathcal{Y}^n, E)$ with the vertex set $\mathcal{Y}^n$. We define an edge $(y^n, \hat{y}^n) \in E$ between two vertices $y^n \in \mathcal{Y}^n$ and $\hat{y}^n \in \mathcal{Y}^n$ if there exists some $x^n \in \mathcal{X}^n$ such that $(x_j, y_j) \in \bigcup_{G\in\mathcal{A}_i} E_G$ and $(x_j, \hat{y}_j) \in \bigcup_{G\in\mathcal{A}_i} E_G$ for $j = 1, \ldots, n$. We denote the chromatic number of $\bar{G}_i^{(n)}$ by $\chi(\bar{G}_i^{(n)})$.

We define an ambiguity set for each $x^n \in \mathcal{X}^n$,

$$\mathcal{I}_i^X(x^n) \triangleq \{y^n \in \mathcal{Y}^n : (x_j, y_j) \in \bigcup_{G\in\mathcal{A}_i} E_G \text{ for } j = 1, \ldots, n\} \tag{49}$$

where the size of the largest single-term ambiguity set is given as,

$$\lambda_i \triangleq \max_{x\in\mathcal{X}} |\mathcal{I}_i^X(x)|, \tag{50}$$

and note that $\max_{x^n\in\mathcal{X}^n} |\mathcal{I}_i^X(x^n)| \leq \lambda_i^n$. Next, we define an ambiguity set for each $y^n \in \mathcal{Y}^n$,

$$\mathcal{I}_i^Y(y^n) \triangleq \{f^n(x^n, y^n) \in \mathcal{F}^n : x_j \in \mathcal{X} \text{ and } (x_j, y_j) \in \bigcup_{G\in\mathcal{A}_i} E_G \text{ for } j = 1, \ldots, n\} \tag{51}$$

and define the size of the largest single-term ambiguity set as,

$$\mu_i \triangleq \max_{y\in\mathcal{Y}} |\mathcal{I}_i^Y(y)|, \tag{52}$$

where $\max_{y^n\in\mathcal{Y}^n} |\mathcal{I}_i^Y(y^n)| \leq \mu_i^n$. Given $i \in \{1, \ldots, |\mathcal{A}|\}$, from Equation (31), there exists a family $\mathcal{H}$ of

$$|\mathcal{H}| = \lceil \lambda_i^n (\log \chi(\bar{G}_i^{(n)}))e \rceil \tag{53}$$

functions such that $h : \{1, \ldots, \chi(\bar{G}_i^{(n)})\} \to \{1, \ldots, \lambda_i^{2n}\}$ for all $h \in \mathcal{H}$ and for each $\mathcal{S} \subseteq \{1, \ldots, \chi(\bar{G}_i^{(n)})\}$ of size $|\mathcal{S}| \leq \lambda_i^n$, there exists an $h \in \mathcal{H}$ injective over $\mathcal{S}$. For each $i \in \{1, \ldots, |\mathcal{A}|\}$, the two users agree on a family of functions $\mathcal{H}$ and a coloring of graph $\bar{G}_i^{(n)}$ with $\chi(\bar{G}_i^{(n)})$ colors. Suppose user 1 observes $x^n \in \mathcal{X}^n$ and user 2 observes $y^n \in \mathcal{Y}^n$. User 1 sends the index of the partition for p to user 2 by using no more than $\lceil \log |\mathcal{A}| \rceil$ bits. User 1 then finds a function $h \in \mathcal{H}$ that is injective over the colors of the vertices $y^n \in \mathcal{I}_i^X(x^n)$ from Equation (49) and sends its index to user 2 by using no more than

$$\lceil \log |\mathcal{H}| \rceil = \lceil \log \lceil \lambda_i^n (\log \chi(\bar{G}_i^{(n)}))e \rceil \rceil \tag{54}$$

bits. User 2 then evaluates the corresponding function for the assigned color of y^n and sends it back to user 1 by using no more than $\lceil \log \lambda_i^{2n} \rceil$ bits. After this step, user 1 learns the color of y^n, from which it recovers y^n by using the observed x^n. User 1 then evaluates the function $f^n(x^n, y^n)$, and sends it to user 2 by using no more than $\lceil \log \mu_i^n \rceil$ bits. In doing so, she assigns a distinct index for each sequence of function values in the ambiguity set $\mathcal{I}_i^Y(y^n)$ from Equation (34). User 2 can then recover the function $f^n(x^n, y^n)$ by using y^n and the received index. Overall, this protocol requires no more than

$$\lceil \log \lceil \lambda_i^n (\log \chi(\bar{G}_i^{(n)}))e \rceil \rceil + \lceil 2n \log \lambda_i \rceil + \lceil n \log \mu_i \rceil \tag{55}$$

bits to be transmitted in total, therefore

$$l_{PR}^{(n)} \leq \min_{\mathcal{A} \in \bar{\mathcal{A}}} \frac{1}{n} \left(\lceil \log |\mathcal{A}| \rceil + \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(\lceil \log \lceil \lambda_i^n (\log \chi(\bar{G}_i^{(n)}))e \rceil \rceil + \lceil 2n \log \lambda_i \rceil + \lceil n \log \mu_i \rceil \right) \right) \tag{56}$$

$$\leq \min_{\mathcal{A} \in \bar{\mathcal{A}}} \left(\frac{1}{n} \lceil \log |\mathcal{A}| \rceil + \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(3 \log \lambda_i + \log \mu_i + \frac{1}{n} \log \log \chi(\bar{G}_i^{(1)}) + \frac{\log n}{n} + \frac{4 + \log e}{n} \right) \right) \tag{57}$$

Combining the bounds obtained from the two protocols in Equations (48) and (57), we have the following upper bound on the worst-case message length with partial reconciliation,

$$\begin{aligned} l_{PR}^{(n)} \leq \min_{\mathcal{A} \in \bar{\mathcal{A}}} \Bigg(\frac{1}{n} \lceil \log |\mathcal{A}| \rceil + \min \Bigg\{ & \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(\lceil \log \chi(G_{X,\mathcal{A}_i}) \rceil + \lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil \right), \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \Big(3 \log \lambda_i \\ & + \log \mu_i + \frac{1}{n} \log \log \chi(\bar{G}_i^{(1)}) + \frac{\log n}{n} + \frac{4 + \log e}{n} \Big) \Bigg\} \Bigg). \end{aligned} \tag{58}$$

At the outset, partial reconciliation characterizes the interplay between reconciliation and communication costs. In order to understand this inherent *reconciliation-communication* trade-off, we next identify the cases for which reconciling the missing information is better or worse than not reconciling them. To do so, we provide sufficient conditions under which reconciliation-based strategies can outperform the strategies that do not start with a reconciliation procedure, and vice versa, and show that either strategy can outperform the other. Finally, we demonstrate that partial reconciliation can strictly outperform both.

4. Cases in which Strategies that Do Not Start with a Reconciliation Procedure is Better than Perfect Reconciliation

In this section, we demonstrate that strategies with no explicit reconciliation step can be strictly better than perfect reconciliation.

Proposition 4. *Strategies that do not start with an explicit reconciliation procedure is better than perfect reconciliation if*

$$\begin{aligned} & \frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil \\ & > \min \left\{ \lceil \log \chi(G_{X,\mathcal{G}}) \rceil + \lceil \log \chi(G_{Y,\mathcal{G}}) \rceil, 3 \log \lambda + \log \mu + \frac{1}{n} \log \log \chi(\bar{G}^{(1)}) + \frac{\log n}{n} + \frac{4 + \log e}{n} \right\}. \end{aligned} \tag{59}$$

Proof. The result follows from comparing the lower bound on the number of bits required for the perfect reconciliation setting from Equation (24) with the upper bound from Equation (58). □

Corollary 1. *Strategies with no explicit reconciliation step can strictly outperform perfect reconciliation.*

Proof. Consider a scenario in which there exists a *parent* distribution $p^* \in \mathcal{P}$ such that $\text{supp}(p) \subseteq \text{supp}(p^*)$ for all $p \in \mathcal{P}$, then, reconciliation cannot perform better than the strategies with no explicit

reconciliation step. This immediately follows from: (i) any zero-error communication strategy for p^* is a valid strategy with no explicit reconciliation step, since $\bigcup_{p\in\mathcal{P}} \text{supp}(p) = \text{supp}(p^*)$, (ii) any perfect reconciliation scheme should ensure a valid zero-error communication strategy for p^*, as it may appear as the true distribution. Therefore, reconciling distributions cannot decrease the overall message length. Suppose that there exists some $q \in \mathcal{Q}$ for which $\text{supp}(q) \subseteq \text{supp}(p)$ for all $p \in \mathcal{P}$. Then, Corollary 1 holds whenever $|\mathcal{P}| > 1$. □

We next consider the following example to elaborate on the impact of overlap between the edges of bipartite graphs on the worst-case message length. To do so, we let $n = 1$ and investigate the following class of graphs.

Definition 2. *(Z-Graph) Consider a class of graphs $\mathcal{G}$ for which there exists a single $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $(x, y) \in E_G$ for all $G \in \mathcal{G}$. Additionally, assume that for any $(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$ such that $(\hat{x}, \hat{y}) \in E_G$ for some $G \in \mathcal{G}$, then either $x = \hat{x}$ or $y = \hat{y}$. In that sense, the structure of these graphs resemble a Z shape, hence we refer to them as Z-graphs. For this class of graphs, $\lambda_G = \lambda_G(x)$ and $\mu_G = \mu_G(y)$ for any $G \in \mathcal{G}$.*

Lemma 2. *Consider the class of graphs defined in Definition 2. For this class of graphs, the worst-case message length for strategies with no explicit reconciliation step satisfies,*

$$l_{RF}^{(1)} \leq \lceil \log \chi(G_{Y,\mathcal{G}}) \rceil + \lceil \log \mu_{\mathcal{G}} \rceil \tag{60}$$

where $\lceil \log \chi(G_{Y,\mathcal{G}}) \rceil$ is defined in Section 3.2 and $\mu_{\mathcal{G}} = \max_{y\in\mathcal{Y}} |\cup_{G\in\mathcal{G}} \mathcal{I}_{Y,G}(y)|$ such that $\mathcal{I}_{Y,G}(y)$ is as given in Equation (12).

Proof. Consider the following encoding scheme. Group all the neighbors $x' \in \mathcal{X}$ of y in $\cup_{G\in\mathcal{G}} G$ that lead to the same function value $f(x', y)$. Assign a single distinct codeword to each of these groups. User 1 sends the corresponding codeword to user 2, which requires no more than $\lceil \log \mu_{\mathcal{G}} \rceil$ bits, after which user 2 can recover the correct function value. Next, construct the graph $G_{Y,\mathcal{G}}$ as defined in Section 3.2. Find the minimum coloring of $G_{Y,\mathcal{G}}$, and assign a distinct codeword to each of the colors. User 2 then sends the corresponding codeword to user 1, by using no more than $\lceil \log \chi(G_{Y,\mathcal{G}}) \rceil$ bits. Note that user 1 can infer the correct function value after this step, as she already knows the bipartite graph G that corresponds to the true distribution and given x and G, each color represents a distinct function value. □

Example 1. *Consider the framework of Section 3.1 along with a class of Z-graphs $\mathcal{G} = \{G_1, G_2\}$ and $\mathcal{B} = \{B_1, B_2\}$. That is, G_1, G_2, B_1, B_2 share an edge $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $(x, y) \in E_{G_1}, E_{G_2}, E_{B_1}, E_{B_2}$. Moreover, for any other edge $(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$, either $\hat{x} = x$ or $\hat{y} = y$. Assume that $f(x, y)$ is distinct for each edge (x, y) in $\mathcal{G}$. Let*

$$\omega \triangleq |\{(\hat{x}, y) : (\hat{x}, y) \in E_{G_1}, E_{G_2}, \hat{x} \in \mathcal{X}\}| \tag{61}$$

represent the number of common edges, i.e., overlap, between G_1 and G_2, where $1 \leq \omega \leq \min\{\mu_{G_1}, \mu_{G_2}\}$. Note that the overlap between G_1 and G_2 can only consist of the edges that share the endpoint y. We consider the following four cases that may occur for the relations between the structures of the graphs G_1, G_2, B_1, B_2.

1. *$E_{B_1} \subseteq E_{G_1}$, $E_{B_2} \nsubseteq E_{G_1}$, $E_{B_1} \nsubseteq E_{G_2}$, $E_{B_2} \subseteq E_{G_2}$. In this case, no reconciliation is always better than reconciliation, because whenever user 2 observes B_1 (respectively B_2), he can infer that user 1 knows G_1 (respectively G_2).*
2. *$E_{B_1} \subseteq E_{G_1}$, $E_{B_2} \subseteq E_{G_1}$, $E_{B_1} \nsubseteq E_{G_2}$, $E_{B_2} \nsubseteq E_{G_2}$. In this case, no reconciliation is again optimal as user 2 can infer that user 1 is knows G_1 whenever he observes B_1 or B_2.*
3. *$E_{B_1} \nsubseteq E_{G_1}$, $E_{B_2} \nsubseteq E_{G_1}$, $E_{B_1} \subseteq E_{G_2}$, $E_{B_2} \subseteq E_{G_2}$. Then, no reconciliation is again optimal as user 2 can infer that user 1 knows G_2 if she observes B_2 or B_1.*

4. $E_{B_1} \nsubseteq E_{G_1}$, $E_{B_2} \subseteq E_{G_1}$, $E_{B_1} \subseteq E_{G_2}$, $E_{B_2} \subseteq E_{G_2}$. *In this case, the chromatic number of the reconciliation graph is given by* $\chi(R) = 2$ *from Definition 1. Then the worst-case message length for the perfect reconciliation scheme satisfies,*

$$l_R^{(1)} \geq 1 + \max\{\lceil \log(\lambda_{G_1} + \mu_{G_1} - 1) \rceil, \lceil \log \lambda_{G_2} + \mu_{G_2} - 1 \rceil\}. \tag{62}$$

which follows from Lemma 1. On the other hand, we find that the worst-case message length for the no reconciliation scheme satisfies

$$l_{RF}^{(1)} \leq \max\{\lceil \log(\lambda_{G_1}) \rceil, \lceil \log(\lambda_{G_2}) \rceil\} + \lceil \log(\mu_{G_1} + \mu_{G_2} - \omega) \rceil, \tag{63}$$

which follows from Lemma 2 and the following coloring scheme. Suppose $\max\{\lambda_{G_1}, \lambda_{G_2}\} = \lambda_{G_1}$. *Using* λ_{G_1} *colors, assign each* $\hat{y} \in \mathcal{Y}$ *that is connected to* x *in* G_1 *a distinct color. Next, take* λ_{G_2} *of these colors excluding the color assigned to node* y*, and color each* $\hat{y} \neq y$ *that is connected to* x *in* G_2 *with a distinct color. Note that this is a valid coloring since there are only two bipartite graphs* G_1 *and* G_2*, corresponding to two cliques whose sizes are* λ_{G_1} *and* λ_{G_2} *in the characteristic graph and the only common node between these two cliques is* y*. Furthermore, no edge exists across the two cliques. Hence, no reconciliation is better than perfect reconciliation whenever*

$$\begin{aligned} \max\{\lceil \log \lambda_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil\} + \lceil \log(\mu_{G_1} + \mu_{G_2} - \omega) \rceil \\ < 1 + \max\{\lceil \log(\lambda_{G_1} + \mu_{G_1} - 1) \rceil, \lceil \log(\lambda_{G_2} + \mu_{G_2} - 1) \rceil\}. \end{aligned} \tag{64}$$

As an example, consider the graphs illustrated in Figure 4 for which $\lambda_{G_1} = \lambda_{G_2} = \mu_{G_1} = \mu_{G_2} = 2$ *and* $\omega = \mu_{G_1} = \mu_{G_2}$*. The corresponding characteristic graph and coloring of* $G_{Y,\mathcal{G}}$ *is illustrated in Figure 5. For this case, we observe that no reconciliation is always better than reconciliation.*

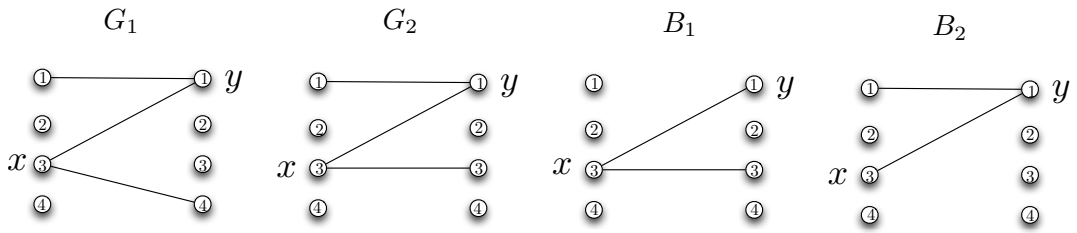


Figure 4. Example graphs $\mathcal{G} = \{G_1, G_2\}$ and $\mathcal{B} = \{B_1, B_2\}$, where $\lambda_{G_1} = \lambda_{G_2} = \mu_{G_1} = \mu_{G_2} = 2$.

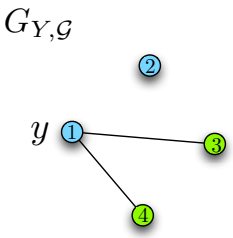


Figure 5. Coloring of the characteristic graph $G_{Y,\mathcal{G}}$.

We note that the performance of a particular communication strategy with respect to others greatly depends on the structure of the partial information as well as the true probability distribution of the observed symbols. In the following section, we show that there exist cases for which reconciling the true distribution only partially can lead to better worst-case message length then both the strategies

from Sections 3.1 and 3.2, indicating that the best communication strategy under partial information may result from a balance between reconciliation and communication costs.

5. Cases in Which Partial Reconciliation is Better

We now investigate the conditions under which partially reconciling the graph information is better than perfect reconciliation. To do so, we initially compare the perfect and partial reconciliation strategies.

Proposition 5. *Partial reconciliation is better than perfect reconciliation if*

$$\begin{aligned}\frac{\lceil \log \chi(R) \rceil}{n} + \max_{G \in \mathcal{G}} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil \\ > \min_{\mathcal{A} \in \bar{\mathcal{A}}} \left(\frac{1}{n} \lceil \log |\mathcal{A}| \rceil + \min \left\{ \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(\lceil \log \chi(G_{X,\mathcal{A}_i}) \rceil + \lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil \right), \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(3 \log \lambda_i \right. \right. \right. \\ \left. \left. \left. + \log \mu_i + \frac{1}{n} \log \log \chi(\bar{G}_i^{(1)}) + \frac{\log n}{n} + \frac{4 + \log e}{n} \right) \right\} \right).\end{aligned} \tag{65}$$

Proof. The right-hand side of Equation (65) is an upper bound on the zero-error message length with partial reconciliation from Equation (58), whereas the left-hand side lower bounds the zero-error codeword length for perfect reconciliation via Equation (24), from which Equation (65) follows. □

We next show that there exist cases for which partial reconciliation strictly outperforms the strategies from Sections 3.1 and 3.2. To do so, we let $n = 1$ and again focus on the class of graphs introduced in Definition 2. First, we present an upper bound on the worst-case message length with partial reconciliation for Z-graphs.

Lemma 3. *The worst-case message length with partial reconciliation for the class of graphs from Definition 2 can be upper bounded by,*

$$l_{PR}^{(1)} \leq \min_{\mathcal{A} \in \bar{\mathcal{A}}} \left(\lceil \log |\mathcal{A}| \rceil + \max_{i \in \{1,\ldots,|\mathcal{A}|\}} \left(\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil + \lceil \log \mu_{\mathcal{A}_i} \rceil \right) \right) \tag{66}$$

where $\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil$ *is as defined in Section 3.3 and* $\mu_{\mathcal{A}_i} = \max_{y \in \mathcal{Y}} |\cup_{G \in \mathcal{A}_i} \mathcal{I}_{Y,G}(y)|$ *with* $\mathcal{I}_{Y,G}(y)$ *as described in Equation (12).*

Proof. To prove achievability, note that for a given partition $\mathcal{A}$, at least $\lceil \log |\mathcal{A}| \rceil$ bits are necessary for sending the partition index, which reconciles each graph up to the class of graphs in the partition it is assigned to. After reconciliation, zero-error communication requires no more than $\max_{i:\mathcal{A}_i \in \mathcal{A}} \left(\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil + \lceil \log \mu_{\mathcal{A}_i} \rceil \right)$ in the worst-case. We show this by considering an encoding scheme that ensures zero-error communication for any graph in $\mathcal{A}_i$ by using $\left(\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil + \lceil \log \mu_{\mathcal{A}_i} \rceil \right)$ bits. Group all the neighbors $x' \in \mathcal{X}$ of y in $\cup_{G \in \mathcal{A}_i} G$ that lead to the same function value $f(x', y)$. Assign a single distinct codeword to each of these groups. Note that this requires no more than $\lceil \log \mu_{\mathcal{A}_i} \rceil$ bits in total. Next, for a given partition $\mathcal{A}_i$, construct the graph $G_{Y,\mathcal{A}_i}$ as defined in Section 3.3. Find the minimum coloring of $G_{Y,\mathcal{A}_i}$, and assign a distinct codeword to each of the colors, which requires no more than $\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil$ bits in total. Then, fix a partitioning $\mathcal{A}$ of $\mathcal{G}$ and use the following communication scheme. User 1, using the partition $\mathcal{A}$, sends the index of the group in which her graph G resides. Then, users 1 and 2 use a robust communication scheme for all the graphs contained in this group. To do so, user 1 sends the codeword assigned to $x' \in \mathcal{X}$ by using no more than $\lceil \log \mu_{\mathcal{A}_i} \rceil$ bits, after which user 2 can recover the correct function value. Then, user 2 sends the color assigned to $y' \in \mathcal{Y}$ using no more than $\lceil \log \chi(G_{Y,\mathcal{A}_i}) \rceil$ bits, after which user 1 can learn the correct function value. □

Proposition 6. *Partial reconciliation can strictly outperform the strategies from Sections 3.1 and 3.2.*

Proof. Consider the set of Z-graphs $\mathcal{G} = \{G_1, G_2, G_3\}$ and $\mathcal{B} = \{B\}$ in Figure 6. The edge sets satisfy $E_{G_3} \subset E_{G_1}$, $E_{G_2} \cap E_{G_1} = \{(x, y)\}$, and $E_B = \{(x, y)\}$.

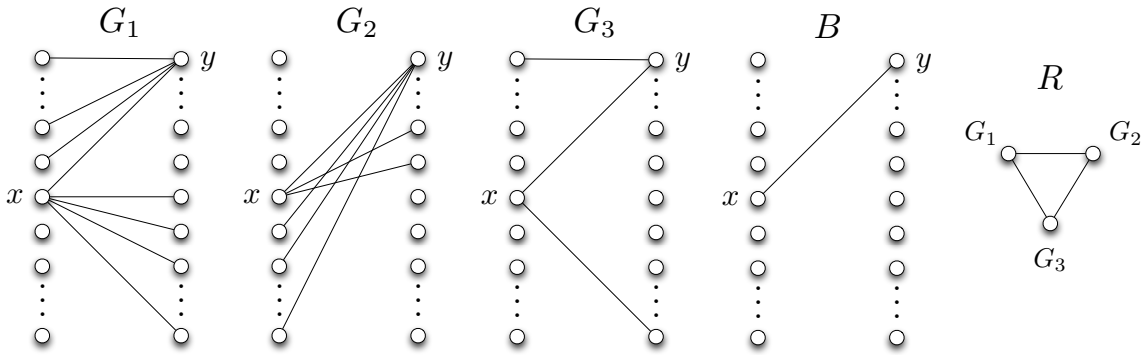


Figure 6. Bipartite graphs $\mathcal{G} = \{G_1, G_2, G_3\}$, $\mathcal{B} = \{B\}$, and the corresponding reconciliation graph R.

Let $f(x, y)$ be distinct for each edge (x, y) in $\mathcal{G}$, and that $\lambda_{G_i} \geq 2$ for some $i \in \{1, 2\}$. First, consider the protocol from Section 3.2. It can be shown that this protocol satisfies

$$l_{RF}^{(1)} \geq 1 + \lceil \log(\mu_{G_1} + \mu_{G_2} - 1) \rceil \tag{67}$$

which results from the following observation. From Proposition 1, it follows that if $(x', y), (x'', y) \in E_{G_i}$ for some $i \in \{1, 2, 3\}$, then $[\phi_k^X(x', \phi^{k-1}(x', y))]_{k=1}^r$ cannot be a prefix of $[\phi_k^X(x'', \phi^{k-1}(x'', y))]_{k=1}^r$, where $[\phi_k^X(x', \phi^{k-1}(x', y))]_{k=1}^r$ is the sequence of bits sent by user 1 in r rounds. Next, suppose that for some $(x', y) \in E_{G_i}$ and $(x'', y) \in E_{G_j}$ where $i \neq j$, and $[\phi_k^X(x', \phi^{k-1}(x', y))]_{k=1}^r$ is a prefix of $[\phi_k^X(x'', \phi^{k-1}(x'', y))]_{k=1}^r$. Since user 2 does not know the true distribution, she cannot distinguish between x' and x'', causing an error since (x', y) and (x'', y) lead to different function values. This in turn violates the zero-error condition. As a result, $[\phi_k^X(x, \phi^{k-1}(x, y))]_{k=1}^r$ should be prefix free for all $x \in \mathcal{I}_Y(y)$ defined in Equation (34) whose size is $\mu_{G_1} + \mu_{G_2} - 1$. Therefore, user 1 needs to send at least $\lceil \log(\mu_{G_1} + \mu_{G_2} - 1) \rceil$ bits to user 2. Next, we demonstrate that user 2 needs to send at least 1 bit to user 1. Suppose that this is not true, i.e., user 2 does not send anything. Since $\lambda_{G_i} \geq 2$ for some $i \in \{1, 2\}$, in this case user 1 will not be able to distinguish between two distinct function values for at least one graph that may occur at user 1. Therefore, by contradiction, Equation (67) provides a lower bound for Z-graphs for the protocols that do not start with a reconciliation strategy considered in Section 3.2.

Next, consider the perfect reconciliation protocol. For this scheme, we construct the reconciliation graph R as given in Figure 6, and observe that any encoding strategy that allows user 2 to distinguish the graph of user 1 requires 3 colors (distinct codewords). After this step, both users consider one of G_1, G_2, or G_3. Then,

$$l_R^{(1)} \geq \lceil \log 3 \rceil + \max_{G_i \in \mathcal{G}} \lceil \log(\lambda_{G_i} + \mu_{G_i} - 1) \rceil \tag{68}$$

which follows from Lemma 1 with the observation that $|\mathcal{I}_{G_i}(x, y)| = \lambda_{G_i} + \mu_{G_i} - 1$ for $i \in \{1, 2, 3\}$.

Lastly, consider the partial reconciliation protocol. In particular, consider a partial reconciliation scheme achieved by the partitioning $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ such that $\mathcal{A}_1 = \{G_1, G_3\}$, and $\mathcal{A}_2 = \{G_2\}$. Then, from Equation (66), we obtain

$$l_{PR}^{(1)} \leq \log 2 + \max\{\lceil \log \lambda_{G_1} \rceil + \lceil \log \mu_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil + \lceil \log \mu_{G_2} \rceil\}. \tag{69}$$

Therefore, whenever $\lambda_{G_1}, \lambda_{G_2}, \mu_{G_1}, \mu_{G_2}$ satisfy

$$\log 2 + \max\{\lceil \log \lambda_{G_1} \rceil + \lceil \log \mu_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil + \lceil \log \mu_{G_2} \rceil\} < 1 + \lceil \log(\mu_{G_1} + \mu_{G_2} - 1) \rceil, \tag{70}$$

then, partial reconciliation outperforms the strategies from Section 3.2. On the other hand, whenever $\lambda_{G_1}, \lambda_{G_2}, \mu_{G_1}, \mu_{G_2}$ satisfy

$$\begin{aligned} \log 2 + \max\{\lceil \log \lambda_{G_1} \rceil + \lceil \log \mu_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil + \lceil \log \mu_{G_2} \rceil\} \\ < \lceil \log 3 \rceil + \max\{\lceil \log(\lambda_{G_1} + \mu_{G_1} - 1) \rceil, \lceil \log(\lambda_{G_2} + \mu_{G_2} - 1) \rceil\}, \end{aligned} \tag{71}$$

then, partial reconciliation outperforms the perfect reconciliation scheme. By setting $\lambda_{G_1} = 2$, $\mu_{G_1} = 8$, $\lambda_{G_2} = 1$, $\mu_{G_2} = 16$, we observe that $l_{PR}^{(1)} \leq 5$ whereas $l_{R}^{(1)} \geq 6$ and $l_{RF}^{(1)} \geq 6$ and both Equations (70) and (71) are satisfied, from which Proposition 6 follows. □

Therefore, under certain settings, it is strictly better to design the interaction protocols to allow the communicating parties to agree on the true source distribution only partially, than to learn it perfectly or not learn it at all, pointing to an inherent reconciliation-communication tradeoff.

6. Communication Strategies with Symmetric Priors

In this section we let $\mathcal{P} = \mathcal{Q}$ and $|\mathcal{P}| = 1$ and specialize the communication model to the conventional function computation scenario where the true distribution $p(x, y)$ of the sources is known by both users. Users thus share a common bipartite graph $G = (\mathcal{X}, \mathcal{Y}, E)$ which they can leverage for interactive communication. We first state a simple lower bound on the worst-case message length.

Proposition 7. *A lower bound on the worst-case message length when the true distribution is known by both parties is,*

$$l^{(n)} \geq \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil. \tag{72}$$

Proof. For the worst-case codeword length, we have

$$l^{(n)} = \min_{\phi} \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \ell(\phi(x^n, y^n)) \tag{73}$$

$$\geq \max_{(x^n, y^n) \in \mathcal{S}_G^n} \min_{\phi} \frac{1}{n} \ell(\phi(x^n, y^n)) \tag{74}$$

$$\geq \max_{(x^n, y^n) \in \mathcal{S}_G^n} \frac{1}{n} \lceil \log |\mathcal{I}_G(x^n, y^n)| \rceil \tag{75}$$

where Equation (74) follows from the min-max inequality whereas Equation (75) follows from Proposition 1. □

We next consider the upper bounds on the worst-case message length for this scenario. A simple upper bound can be obtained via the graph coloring approach in Theorem 1,

$$l^{(n)} \leq \frac{1}{n} \left\{ \lceil n \log(\chi(G_X)) \rceil + \lceil n \log(\chi(G_Y)) \rceil \right\}, \tag{76}$$

where the characteristic graphs G_X and G_Y are constructed as in Theorem 1 using the bipartite graph corresponding to the true distribution $p(x, y)$. We note that Equation (76) implies that

$$\lim_{n \to \infty} l^{(n)} \leq \log(\chi(G_X)) + \log(\chi(G_Y)). \tag{77}$$

The above approach may yield limited gains for compression for large values of $\chi(G_X)$ and $\chi(G_Y)$, and another round of interaction may help reduce the compression rate. We next provide another upper bound that combines graph coloring and hypergraph partitioning. To do so, we first review the following notable results. The first one is a technical result regarding partitioning hypergraphs.

Lemma 4 ([1]). *Define $\Gamma = (V, E)$ to be a hypergraph with a vertex set of size $|V|$, and the hyperedges $E_m \subseteq V$ with $m = 1, \ldots, |E|$. Assume that each hyperedge has at most κ elements, i.e., $|E_m| \leq \kappa$. Then for any given $\epsilon > 0$, there exists a constant $\rho(\epsilon)$ such that $\forall s \geq (\ln \sqrt{|V||E|})^{1+\epsilon}$ and $s > 1$, a partition $V_1, V_2, \ldots V_{\lceil \frac{\kappa}{s}\rho(\epsilon) \rceil}$ of V can be found with $|V_k \cap E_m| < s$ for all $m = 1, \ldots, |E|$ and $k = 1, \ldots, \lceil \frac{\kappa}{s}\rho(\epsilon) \rceil$.*

We can now state the second useful result.

Lemma 5 ([1]). *The following worst-case codeword length can be achieved in three rounds for $n = 1$,*

$$l^{(1)} \leq \log \Delta_X + \log \Delta_Y + (1+\epsilon) \log \left(\log \sqrt{|\mathcal{X}||\mathcal{Y}|} \right) + 2 \log \rho(\epsilon) + 5. \tag{78}$$

where each person makes two non-empty transmissions.

We next derive an upper bound based on Lemma 5 by increasing the number of interaction rounds and following a sequential hypergraph partitioning approach. This allows the proposed scheme to work in low-rate communication environments when parties do not mind having extra rounds of interaction.

Theorem 2. *Given a joint probability distribution $p(x, y)$, consider the corresponding bipartite graph $G = (\mathcal{X}, \mathcal{Y}, E)$. Consider a partition of $\mathcal{X}^n$ into $\lceil \frac{\Delta_Y^n}{(\ln \sqrt{|\mathcal{X}^n||\mathcal{Y}^n|})^{1+\epsilon}} \rho(\epsilon) \rceil$ groups such that for each group $\mathcal{X}_u^n$,*

$$|\mathcal{X}_u^n \cap \{x^n : (x^n, y^n) \in \mathcal{S}^n,\ x^n \in \mathcal{X}^n\}| \leq (\ln \sqrt{|\mathcal{X}^n||\mathcal{Y}^n|})^{1+\epsilon}, \qquad \forall y^n \in \mathcal{Y}^n, \tag{79}$$

where $u = 1, \ldots, \left\lceil \frac{\min\{\Delta_X^n, \Delta_Y^n\}}{(n \ln \sqrt{|\mathcal{X}||\mathcal{Y}|})^{1+\epsilon}} \rho(\epsilon) \right\rceil$. Then, the worst-case codeword length with four total rounds can be bounded by using sequential hypergraph partitioning as

$$l^{(n)} \leq \log \Delta_X + \log \Delta_Y + \frac{(1+\epsilon)}{n} \log (\log \sqrt{\gamma_n}) + \frac{2}{n} \log \rho(\epsilon) + \frac{5}{n} \tag{80}$$

where $\gamma_n = \max_u \left(|\mathcal{X}_u^n| \times \min\{\Delta_X^n |\mathcal{X}_u^n|, |\mathcal{Y}^n|\} \right) \leq |\mathcal{X}|^n |\mathcal{Y}|^n$.

Proof. Our proof builds upon [1] as follows. The set of symbols in the first round are from $\mathcal{X}^n$ and $\mathcal{Y}^n$ for users 1 and 2, respectively. We assume $\Delta_X > \Delta_Y$ without loss of generality. Let $s_1 = (\ln \sqrt{|\mathcal{X}^n||\mathcal{Y}^n|})^{1+\epsilon}$. From Lemma 4, $\mathcal{X}^n$ can be partitioned into $\lceil \frac{\Delta_Y^n}{s_1} \rho(\epsilon) \rceil$ groups such that for each group $\mathcal{X}_u^n$,

$$|\mathcal{X}_u^n \cap \{x^n : (x^n, y^n) \in \mathcal{S}^n,\ x^n \in \mathcal{X}^n\}| \leq s_1, \qquad \forall y^n \in \mathcal{Y}^n, \tag{81}$$

where $u = 1, \ldots, \lceil \frac{\Delta_Y^n}{s_1} \rho(\epsilon) \rceil$. In this round, user 1 sends the index of the group that her symbol resides in by using no more than $\lceil \log (\frac{\Delta_Y^n}{s_1} \rho(\epsilon)) \rceil$ bits, and user 2 makes a null transmission. Let $\hat{u}$ be the index of the group sent by user 1 in the first round. In the second round, the following set is considered by user 2 after receiving the index from user 1

$$\mathcal{Y}_{\hat{u}}^n = \{y^n : (x^n, y^n) \in \mathcal{S}^n,\ x^n \in \mathcal{X}_{\hat{u}}^n,\ y^n \in \mathcal{Y}^n\}. \tag{82}$$

Note that $|\mathcal{Y}_{\hat{u}}^n| \leq \min\{\Delta_X^n |\mathcal{X}_{\hat{u}}^n|, |\mathcal{Y}^n|\}$. Next, consider a hypergraph $\Gamma = (V, E)$ with the vertex set $V = \mathcal{Y}_{\hat{u}}^n$ and define a hyperedge for each $x^n \in \mathcal{X}_{\hat{u}}^n$ as follows.

$$E_{x^n} = \{y^n : (x^n, y^n) \in \mathcal{S}^n,\ y^n \in \mathcal{Y}_{\hat{u}}^n\}, \tag{83}$$

where $|E| = |\mathcal{X}_{\hat{u}}^n|$, and $|E_{x^n}| \leq \Delta_X^n,\ x^n \in \mathcal{X}_{\hat{u}}^n$. User 1 can also determine this set by using the group index for her symbol and the relations between the function values of both parties. Let

$$s_{2\hat{u}} = \left(\ln \sqrt{|V||E|}\right)^{1+\epsilon} = \left(\ln \sqrt{|\mathcal{Y}_{\hat{u}}^n||\mathcal{X}_{\hat{u}}^n|}\right)^{1+\epsilon} \leq s_1. \tag{84}$$

User 2 then partitions $\mathcal{Y}_{\hat{u}}^n$ into $\lceil \frac{\Delta_X^n}{s_{2\hat{u}}} \rho(\epsilon) \rceil$ groups and sends the group index for his symbol, which requires no more than $\lceil \log (\frac{\Delta_X^n}{s_{2\hat{u}}} \rho(\epsilon)) \rceil$ bits. After receiving the group index, user 1 has to decide from at most $s_{2\hat{u}}$ possible symbols from user 2.

In the third round, symbols are now restricted to a subspace of $\mathcal{X}^n \times \mathcal{Y}^n$ with at most s_1 possible symbols from user 1 for each symbol from user 2 and at most $s_{2\hat{u}}$ possible symbols from user 2 for each one of the symbols of user 1. Then, by using ([1], Lemma 2), one can find that no more than $\lceil \log(s_1) \rceil + 2\lceil \log(s_{2\hat{u}}) \rceil$ bits are required.

This scheme requires four rounds of interaction in total; each person makes two non-empty transmissions. The total number of bits required in the worst-case satisfies

$$l^{(n)} \leq \frac{1}{n} \max_u \left(\left\lceil \log \left(\frac{\Delta_Y^n}{s_1} \rho(\epsilon) \right) \right\rceil + \left\lceil \log \left(\frac{\Delta_X^n}{s_{2u}} \rho(\epsilon) \right) \right\rceil + \lceil \log s_1 \rceil + 2 \lceil \log s_{2u} \rceil \right) \tag{85}$$

$$\leq \frac{1}{n} \max_u \left(\log \Delta_Y^n + \log \Delta_X^n + \log s_{2u} + 2 \log \rho(\epsilon) + 5 \right) \tag{86}$$

$$\leq \log \Delta_X + \log \Delta_Y + \frac{1}{n} \left((1+\epsilon) \log \left(\log \sqrt{\gamma_n} \right) + 2 \log \rho(\epsilon) + 5 \right), \tag{87}$$

where $\gamma_n = \max_u \left(|\mathcal{X}_u^n||\mathcal{Y}_u^n| \right) \leq |\mathcal{X}|^n |\mathcal{Y}|^n$. □

Corollary 2. *In the limit of large block lengths, the upper bound of Theorem 2 satisfies*

$$\lim_{n \to \infty} l^{(n)} = \log \Delta_X + \log \Delta_Y. \tag{88}$$

Proof. As $|\mathcal{X}_u^n| \leq |\mathcal{X}^n|$ and $|\mathcal{Y}_u^n| \leq |\mathcal{Y}^n|$ for all partitions u of $\mathcal{X}^n$ and $\mathcal{Y}^n$, from Theorem 2,

$$\begin{aligned} \lim_{n \to \infty} l^{(n)} &\leq \lim_{n \to \infty} \left(\log \Delta_X + \log \Delta_Y + \frac{(1+\epsilon)}{n} \log \left(\log \sqrt{|\mathcal{X}||\mathcal{Y}|} \right) + (1+\epsilon) \frac{\log n}{n} + \frac{2}{n} \log \rho(\epsilon) + \frac{5}{n} \right) \\ &= \log \Delta_X + \log \Delta_Y. \end{aligned} \tag{89}$$

□

Lemma 5 and Theorem 2 apply the hypergraph partitioning technique to the bipartite graph of the joint distribution $p(x, y)$, but provide achievable rates by first performing source reconstruction at the two ends, after which both users can compute the correct function value. The next theorem takes the function values into account while constructing the hypergraph partitioning algorithm, with the use of characteristic graphs.

Consider any valid coloring of the characteristic graphs G_X^n and G_Y^n defined in Section 3.1. Note that by using their own symbols, each user can recover the correct function values upon receiving the color from the other user. The problem now reduces to sharing the colors between the two parties correctly, for which we apply sequential hypergraph partitioning to the *colors* of the graphs G_X^n and G_Y^n.

Theorem 3. *Define a coloring $\alpha : G_X^n \to \mathcal{C}_\alpha$ for G_X^n with $|\mathcal{C}_\alpha|$ colors, and a coloring $\beta : G_Y^n \to \mathcal{C}_\beta$ for G_Y^n with $|\mathcal{C}_\beta|$ colors. Let $c(x^n)$ and $c(y^n)$ denote the colors assigned to x^n and y^n by the colorings α and β, respectively. Define the ambiguity set for color $c_X \in \mathcal{C}_\alpha$ as*

$$\mathcal{J}_X(c_X) \triangleq \{c_Y \in \mathcal{C}_\beta : (x^n, y^n) \in \mathcal{S}^n, c(x^n) = c_X, c(y^n) = c_Y\} \tag{90}$$

with the size bound $\Delta_{X\alpha}^{(n)} \triangleq \max_{c_X \in \mathcal{C}_\alpha} |\mathcal{J}_X(c_X)|$, and the ambiguity set for color $c_Y \in \mathcal{C}_\beta$ as

$$\mathcal{J}_Y(c_Y) \triangleq \{c_X \in \mathcal{C}_\alpha : (x^n, y^n) \in \mathcal{S}^n, c(x^n) = c_X, c(y^n) = c_Y\} \tag{91}$$

with the size bound $\Delta_{Y\beta}^{(n)} \triangleq \max_{c_Y \in \mathcal{C}_\beta} |\mathcal{J}_Y(c_Y)|$. Consider a partition of $\mathcal{C}_\alpha$ into $\left\lceil \frac{\min\{\Delta_{X\alpha}^{(n)}, \Delta_{Y\beta}^{(n)}\}}{(\ln \sqrt{|\mathcal{C}_\alpha||\mathcal{C}_\beta|})^{1+\epsilon}} \rho(\epsilon) \right\rceil$ groups such that for each group $\mathcal{C}_{\alpha u}$,

$$|\mathcal{C}_{\alpha u} \cap \mathcal{J}_Y(c_Y)| \leq \left(\ln \sqrt{|\mathcal{C}_\alpha||\mathcal{C}_\beta|}\right)^{1+\epsilon} \qquad \forall c_Y \in \mathcal{C}_\beta, \tag{92}$$

and

$$\mathcal{C}_{\beta u} \triangleq \{c_Y \in \mathcal{C}_\beta : (x^n, y^n) \in \mathcal{S}^n,\ c(x^n) = c_X \in \mathcal{C}_{\alpha u},\ c(y^n) = c_Y\}. \tag{93}$$

where $u = 1, \ldots, \left\lceil \frac{\min\{\Delta_{X\alpha}^{(n)}, \Delta_{Y\beta}^{(n)}\}}{(\ln \sqrt{|\mathcal{C}_\alpha||\mathcal{C}_\beta|})^{1+\epsilon}} \rho(\epsilon) \right\rceil$. Then, the worst-case message length can be upper bounded as,

$$l^{(n)} \leq \min_{\alpha,\beta} \left(\frac{\log \Delta_{X\alpha}^{(n)}}{n} + \frac{\log \Delta_{Y\beta}^{(n)}}{n} + \frac{(1+\epsilon)}{n} \log \left(\log \sqrt{\gamma_{\alpha,\beta}} \right) + \frac{2}{n} \log \rho(\epsilon) + \frac{5}{n} \right), \tag{94}$$

where $\gamma_{\alpha,\beta} = \max_u \left(|\mathcal{C}_{\alpha u}||\mathcal{C}_{\beta u}| \right)$.

Proof. Assume $\Delta_{X\alpha}^{(n)} > \Delta_{Y\beta}^{(n)}$ without loss of generality. Choose $s_1 = (\ln \sqrt{|\mathcal{C}_\alpha||\mathcal{C}_\beta|})^{1+\epsilon}$. Partition $\mathcal{C}_\alpha$ into $\lceil \frac{\Delta_{Y\beta}^{(n)}}{s_1} \rho(\epsilon) \rceil$ groups such that in each partition the number of colors from the ambiguity set is no greater than s_1. Hence, for any $c_Y \in \mathcal{C}_\beta$,

$$|\mathcal{C}_{\alpha u} \cap \{c_X : (x^n, y^n) \in \mathcal{S}^n,\ c(x^n) = c_X \in \mathcal{C}_\alpha,\ c(y^n) = c_Y\}| \leq s_1, \tag{95}$$

for $u = 1, \ldots, \lceil \frac{\Delta_{Y\alpha}^{(n)}}{s_1} \rho(\epsilon) \rceil$. In the first round, user 1 sends the index of the partition the color of her symbols lies in. This requires at most $\lceil \log (\frac{\Delta_{Y\beta}^{(n)}}{s_1} \rho(\epsilon)) \rceil$ bits, whereas user 2 makes an empty transmission. Denote $\hat{u}$ as the index of the partition sent from user 1. In the second round, upon receiving $\hat{u}$ from user 1, user 2 considers a set $\mathcal{C}_{\beta\hat{u}}$ given as $u = \hat{u}$ in Equation (93), where $|\mathcal{C}_{\beta,\hat{u}}| \leq \min\{\tilde{\lambda}_{max}|\mathcal{C}_{\alpha\hat{u}}|, |\mathcal{C}_\beta|\}$. Define a hypergraph $\Gamma = (V, E)$ with a vertex set $V = \mathcal{C}_{\beta\hat{u}}$ and a hyperedge $E_{c_X} = \{c_Y : (x^n, y^n) \in \mathcal{S}^n,\ c(y^n) = c_Y \in \mathcal{C}_{\beta\hat{u}},\ c(x^n) = c_X\}$ for each $c_X \in \mathcal{C}_{\alpha\hat{u}}$ such that $|E| = |\mathcal{C}_{\alpha\hat{u}}|$, and $|E_{c_X}| \leq \Delta_{X\alpha}^{(n)}$ for every $c_X \in \mathcal{C}_{\alpha\hat{u}}$. Define $s_{2\hat{u}} = (\ln \sqrt{|V||E|})^{1+\epsilon} = (\ln \sqrt{|\mathcal{C}_{\beta\hat{u}}||\mathcal{C}_{\alpha\hat{u}}|})^{1+\epsilon} < s_1$ and partition $\mathcal{C}_{\beta\hat{u}}$ into $\lceil \frac{\Delta_{X\alpha}^{(n)}}{s_{2\hat{u}}} \rho(\epsilon) \rceil$ to groups so that user 2 can send the index of his symbols with at most $\lceil \log (\frac{\Delta_{X\alpha}^{(n)}}{s_{2\hat{u}}} \rho(\epsilon)) \rceil$ bits. Upon receiving the index, user 1 can reduce the number of possible symbols from user 2 to at most $s_{2\hat{u}}$. Colors in the third round are restricted to a subset of $\mathcal{C}_\alpha \times \mathcal{C}_\beta$ such that for every color from user 1 (user 2), there are at most $s_{2\hat{u}}$ (s_1) possible colors exist from user 2 (user 1). Then Equation (94) follows from ([1], Lemma 2). □

It can be observed from Equation (94) that different codeword lengths are obtained by different colorings, since they lead to different color and ambiguity set sizes. In general, there exists a trade-off

between the ambiguity set sizes and the number of colors, such that using a smaller number of colors may in turn increase the ambiguity set sizes. The exact nature of the bound depends on the graphical structures such as degree and connectivity, however, any valid coloring allows error-free recovery. For instance, assigning a distinct color to each element of $\mathcal{X}^n$ and $\mathcal{Y}^n$ is a valid coloring scheme. If one restricts oneself to such set of colorings, the coding scheme of Theorem 3 will reduce to that of Theorem 2, hence the bound in Theorem 3 generalizes the achievable protocols in Theorem 2.

7. Conclusions

In this paper, we have considered a communication scenario in which two parties interact to compute a function of two correlated sources with zero error. The prior distribution available at one of the communicating parties is possibly different from the true distribution of the sources. In this setting, we have studied the impact of reconciling the missing information about the true distribution prior to communication on the worst-case message length. We have identified sufficient conditions under which reconciling the partial information is better or worse than not reconciling it but instead using a robust communication protocol that ensures zero-error recovery despite the asymmetry in the knowledge of the distribution. Accordingly, we have provided upper and lower bounds on the worst-case message length for computing multiple descriptions of the given function. Our results point to an inherent reconciliation-communication tradeoff, in that an increased reconciliation cost often leads to a lower communication cost. A number of interesting future directions remain. In this paper, we do not consider additional strategies which consider further information that may be revealed by the function realizations on the support set. Developing interaction strategies that leverage this information is another interesting future direction. A second one is finding the optimal joint reconciliation-communication strategy in general and the study of alternative upper bounds that take into account the specific structure of the function and input distributions. Another interesting direction is to model the case where knowledge asymmetry is due to one party having superfluous information.

Acknowledgments: This research was sponsored by the U.S. Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. Earlier versions of this work have partially appeared at the IEEE GlobalSIP Symposium on Network Theory, December 2013, IEEE Data Compression Conference (DCC'14), March 2014, and IEEE Data Compression Conference (DCC'16), March 2016. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

Author Contributions: The ideas in this work were formed by the discussions between Basak Guler and Aylin Yener with Prithwish Basu and Ananthram Swami. Basak Guler is the main author of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. El Gamal, A.; Orlitsky, A. Interactive data compression. In Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'84), West Palm Beach, FL, USA, 24–26 October 1984; pp. 100–108.
2. Orlitsky, A. Worst-case interactive communication I: Two messages are almost optimal. *IEEE Trans. Inf. Theory* **1990**, *36*, 1111–1126.
3. Guler, B.; Yener, A.; Basu, P. A study of semantic data compression. In Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP'13), Austin, TX, USA, 3–5 December 2013; pp. 887–890.
4. Guler, B.; Yener, A. Compressing semantic information with varying priorities. In Proceedings of the IEEE Data Compression Conference (DCC'14), Snowbird, UT, USA, 26–28 March 2014; pp. 213–222.
5. Yao, A.C. Some complexity questions related to distributed computing. In Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC'79), Atlanta, GA, USA, 30 April–2 May 1979; pp. 209–213.

6. Feder, T.; Kushilevitz, E.; Naor, M.; Nisan, N. Amortized communication complexity. *SIAM J. Comput.* **1995**, *24*, 736–750.
7. Kushilevitz, E.; Nisan, N. *Communication Complexity*; Cambridge University Press: Cambridge, UK, 1997.
8. Orlitsky, A.; Roche, J.R. Coding for computing. *IEEE Trans. Inf. Theory* **2001**, *47*, 903–917.
9. Ma, N.; Ishwar, P. Some results on distributed source coding for interactive function Computation. *IEEE Trans. Inf. Theory* **2011**, *57*, 6180–6195.
10. Ma, N.; Ishwar, P.; Gupta, P. Interactive source coding for function computation in collocated networks. *IEEE Trans. Inf. Theory* **2012**, *58*, 4289–4305.
11. Yang, E.H.; He, D.K. Interactive encoding and decoding for one way learning: Near lossless recovery with side information at the decoder. *IEEE Trans. Inf. Theory* **2010**, *56*, 1808–1824.
12. Shannon, C. The zero error capacity of a noisy channel. *IRE Trans. Inf. Theory* **1956**, *2*, 8–19.
13. Witsenhausen, H.S. The zero-error side information problem and chromatic numbers. *IEEE Trans. Inf. Theory* **1976**, *22*, 592–593.
14. Simonyi, G. On Witsenhausen's zero-error rate for multiple sources. *IEEE Trans. Inf. Theory* **2003**, *49*, 3258–3260.
15. Körner, J. Coding of an information source having ambiguous alphabet and the entropy of graphs. In Proceedings of the Sixth Prague Conference on Information Theory, Prague, Czech Republic, 19–25 September 1973; pp. 411–425.
16. Alon, N.; Orlitsky, A. Source coding and graph entropies. *IEEE Trans. Inf. Theory* **1995**, *42*, 1329–1339.
17. Doshi, V.; Shah, D.; Médard, M.; Effros, M. Functional compression through graph coloring. *IEEE Trans. Inf. Theory* **2010**, *56*, 3901–3917.
18. Minsky, Y.; Trachtenberg, A.; Zippel, R. Set reconciliation with nearly optimal communication complexity. *IEEE Trans. Inf. Theory* **2003**, *49*, 2213–2218.
19. Nayak, J.; Rose, K. Graph capacities and zero-error transmission over compound channels. *IEEE Trans. Inf. Theory* **2005**, *51*, 4374–4378.
20. Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic Web. *Sci. Am.* **2001**, *284*, 28–37.
21. Sheth, A.; Bertram, C.; Avant, D.; Hammond, B.; Kochut, K.; Warke, Y. Managing semantic content for the Web. *IEEE Internet Comput.* **2002**, *6*, 80–87.
22. Lee, E.A. Cyber physical systems: Design challenges. In Proceedings of the IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC'08), Orlando, FL, USA, 5–7 May 2008; pp. 363–369.
23. Sheth, A.; Henson, C.; Sahoo, S. Semantic sensor Web. *IEEE Internet Comput.* **2008**, *12*, 78–83.
24. Chen, J.; He, D.K.; Jagmohan, A. On the duality between Slepian–Wolf coding and channel coding under mismatched decoding. *IEEE Trans. Inf. Theory* **2009**, *55*, 4006–4018.
25. Juba, B.; Kalai, A.T.; Khanna, S.; Sudan, M. Compression without a common prior: An information-theoretic justification for ambiguity in language. In Proceedings of the Second Symposium on Innovations in Computer Science (ICS 2011), Beijing, China, 7–9 January 2011.
26. Haramaty, E.; Sudan, M. Deterministic compression with uncertain priors. *Algorithmica* **2016**, *76*, 630–653.
27. Guler, B.; Yener, A.; MolavianJazi, E.; Basu, P.; Swami, A.; Andersen, C. Interactive Function Compression with Asymmetric Priors. In Proceedings of the IEEE Data Compression Conference (DCC'16), Snowbird, UT, USA, 30 March–1 April 2016; pp. 379–388.
28. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, 2012.
29. Gács, P.; Körner, J. Common information is far less than mutual information. *Probl. Control Inf. Theory* **1973**, *2*, 149–162.
30. Mehlhorn, K. *Data Structures and Algorithms 1: Sorting and Searching*; Springer Science & Business Media: Berlin, Germany, 2013.