Interactive Function Compression with Asymmetric Priors

Basak Guler* Aylin Yener* Ebrahim MolavianJazi* Prithwish Basu†
Ananthram Swami** Carl Andersen‡

*The Pennsylvania State University
University Park, PA
{basak,ebrahim}@psu.edu yener@ee.psu.edu

†Raytheon BBN Technologies
Cambridge, MA
pbasu@bbn.com

**Army Research Laboratory
Adelphi, MD
a.swami@ieee.org

‡Raytheon BBN Technologies
Rosslyn, VA
canderse@bbn.com

Abstract

We study the interactive compression of an arbitrary function of two discrete sources with zero-error. The information on the joint distribution of the sources available at the two sides is asymmetric, in that one user knows the true distribution, whereas the other user observes a different distribution. This paper considers the minimum worst-case zero-error codeword length under such asymmetric prior distributions. We investigate the cases for which reconciling the information mismatch is better or worse than not reconciling it, but instead using an encoding scheme that ensures zero-error with possibly increased communication rate. Our results indicate a reconciliation-communication tradeoff and that there exist cases for which partially reconciling the mismatched information is better than both perfect reconciliation and no reconciliation.

1 Introduction

In networked systems such as the social media, human-computer interaction, and cyber-physical systems, interaction often takes place between sources with different backgrounds, characteristics, and knowledge bases. These differences lead to different interpretations of the same information. Motivated by such scenarios, we consider zero-error interactive function computation of two correlated sources when the communicating parties may have mismatched information about the true source distribution.

The impact of interaction on the data compression performance has been investigated when both parties know the true distribution of the sources. Interactive compression of correlated sources has been studied in [1] for recovering the sources with zero-error. In [2] and [3], interactive communication has been leveraged to enable zero-error and non-zero error recovery of the information known by one party at the other side, respectively. Interactive communication has been utilized in [4] for computing a function and in [5] for computing functions of multiple source realizations, where it has been shown that computing multiple instances of a function simultaneously could be better than computing each instance separately, and a lower bound on the amount of communication required for computing multiple instances has been established in terms of the communication needed for

This research is sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053.

1068-0314/16 $31.00 © 2016 IEEE
DOI 10.1109/DCC.2016.109

computing a single instance. Computing a function of two correlated sources at one or both parties is investigated in [6] and [7], respectively, for the case of vanishing error probability. Reference [8] has studied the impact of mismatched decoding on the source coding performance in the presence of decoder side information. Compressing a source when two parties have asymmetric information about its distribution is considered in [9] and [10] for one-way (non-interactive) communication.

Zero-error communication protocols often utilize the notion of characteristic graphs to construct a graphical representation of the confusable or ambiguous source instances. As such, characteristic graphs has been leveraged in [11] for compressing a source with zero-error in the presence of decoder side information. They have since found applications in one-way set reconciliation [12], in zero-error compression with compound decoder side information [13], and in function compression with zero [6] and non-zero distortion [14].

We study the worst-case zero-error interactive function computation when the two terminals have *mismatched information* about the true distribution of the sources. We aim at identifying the impact of this mismatch on the worst-case message length. To do so, we investigate the conditions under which reconciliation, i.e., reconciling the true distribution between the two users first and then using it for communication, is better or worse than no reconciliation, i.e., not reconciling the true distribution but instead using a zero-error encoding protocol with increased codeword length. We next construct a communication protocol to create partial levels of agreement about the true distribution and utilize it to identify a reconciliation-communication tradeoff. This tradeoff results from the interplay between the cost of reconciliation and the resulting improvement on the compression performance. We demonstrate that, reconciling the information mismatch may improve the compression performance, but may also fail to do so. In fact, we show that partial reconciliation can strictly outperform both perfect reconciliation or no reconciliation.

In the remainder of the paper, $\mathcal{X}$ denotes a set with cardinality $|\mathcal{X}|$ and $\text{supp}(p) = \{x \in \mathcal{X} : p(x) > 0\}$ is the support set of a probability distribution $p(x)$ over $\mathcal{X}$. We let $x^n = (x_1, \ldots, x_n)$ and $\{0,1\}^* = \cup_{n=1}^{\infty} \{0,1\}^n$. The chromatic number of graph G is $\chi(G)$.

2 System Model and Preliminaries

We consider two discrete correlated sources (X, Y) over a finite set $\mathcal{X} \times \mathcal{Y}$. User 1 observes X, whereas user 2 observes Y. The two sources are generated by a probability distribution $p(x, y)$ from a finite set of distributions $\mathcal{P}$. User 1 observes a sequence $x^n \in \mathcal{X}^n$, whereas user 2 observes a sequence $y^n \in \mathcal{Y}^n$, with probability $p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i)$. Both users know the set $\mathcal{P}$, however, only user 1 knows the nature's selection for $p(x, y) \in \mathcal{P}$. User 2 instead assumes a probability distribution $q(x, y)$ from a finite set of distributions $\mathcal{Q}$ such that $\text{supp}(q) \subseteq \text{supp}(p)$. In essence, this provides some side information to user 2 about the possible candidates for $p(x, y)$, which is a subset of $\mathcal{P}$. Again, both users know $\mathcal{Q}$, however, only user 2 knows the actual $q(x, y)$.

The two parties want to compute a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{F}$ where $\mathcal{F}$ is a finite set. To do so, user 1 recovers some $\hat{y}^n \in \mathcal{Y}^n$ and user 2 recovers some $\hat{x}^n \in \mathcal{X}^n$ such that the function values for the recovered sequences are the same as the original source sequence, leading to the following *zero-error* condition,

$$\Pr\left[f^n(X^n, \hat{Y}^n) \neq f^n(X^n, Y^n) \bigcup f^n(\hat{X}^n, Y^n) \neq f^n(X^n, Y^n)\right] = 0, \tag{1}$$

where $f^n(x^n, y^n) := (f(x_1, y_1), \ldots, f(x_n, y_n))$. In essence, (1) states that

$$f(x_i, \hat{y}_i) = f(\hat{x}_i, y_i) = f(x_i, y_i) \text{ for all } p(x_i, y_i) > 0 \text{ for } i = 1, \ldots, n. \tag{2}$$

Lastly, we let $\mathcal{S}^n := \text{supp}(p(x^n, y^n))$.

We consider interactive encoding strategies in which users take turns to send binary sequences that we refer to as *messages*. We assume that the communication takes place in r-rounds where r is finite. The encoding function is defined as $\phi : \mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}^*$ for which the codeword $\phi(x^n, y^n) = [\phi_i(x^n, y^n)]_{i=1}^r$ is the sequence of messages exchanged for $(x^n, y^n) \in \mathcal{S}^n$, where $\phi_i(x^n, y^n)$ is the message transmitted by the two parties at round i and $\phi^i(x^n, y^n) = [\phi_k(x^n, y^n)]_{k=1}^i$ is the sequence of messages exchanged in the first i rounds for $i \in \{1, \ldots, r\}$. The encoding at a given user is based on the user's own symbols as well as the previous messages

$$\phi_i(x^n, y^n) = [\phi_i^X(x^n, \phi^{i-1}(x^n, y^n)), \phi_i^Y(y^n, \phi^{i-1}(x^n, y^n))], \tag{3}$$

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 sent at round i from user 1 and user 2, respectively. To ensure that users know when a received message ends, we have the following condition. For all $(x^n, y^n), (x^n, \hat{y}^n) \in \mathcal{S}^n$ and $f(x^n, y^n) \neq f(x^n, \hat{y}^n)$, if $[\phi_k(x^n, y^n)]_{k=1}^{i-1} = [\phi_k(x^n, \hat{y}^n)]_{k=1}^{i-1}$ for some $i \in \{2, \cdots, r\}$ then $\phi_i^Y(x^n, \phi^{i-1}(x^n, y^n))$ is not a proper prefix of $\phi_i^Y(x^n, \phi^{i-1}(x^n, \hat{y}^n))$. Same applies when we interchange X and Y, i.e., for user 2. The encoding function is a deterministic mapping known by both parties in advance. The worst-case codeword length for ϕ is

$$l_{n,\phi} = \max_{(x^n, y^n) \in \mathcal{S}^n} \frac{1}{n} |\phi(x^n, y^n)| \quad \text{bits/symbol} \tag{4}$$

where $|\cdot|$ is the length, i.e., number of bits, in a binary sequence. We define the minimum worst-case codeword length as

$$l_n = \min_{\phi} \; l_{n,\phi}. \tag{5}$$

We observe from (1) that the minimum worst-case codeword length is equal for all $p, p' \in \mathcal{P}$ with $\text{supp}(p) = \text{supp}(p')$. We leverage this property for constructing our communication protocols as follows. 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 $p(x, y) > 0$.

We note that $G_p = G_{p'}$ for any $p(x, y), p'(x, y) \in \mathcal{P}$ with $\text{supp}(p) = \text{supp}(p')$. Accordingly, $\mathcal{P}$ can be partitioned into groups of distributions that have the same support set, such that the set of graphs in each partition corresponds to a distinct bipartite graph. We represent this set of distinct bipartite graphs by $\mathcal{G}$, and each $G \in \mathcal{G}$ as $G = (\mathcal{X}, \mathcal{Y}, E_G)$.

Definition 1. *(Ambiguity sets) Given $G \in \mathcal{G}$, define for each $x^n \in \mathcal{X}^n$ a set*

$$\mathcal{I}_{X,G}(x^n) = \{f^n(x^n, y^n) \in \mathcal{F}^n : (x_i, y_i) \in E_G, \; y_i \in \mathcal{Y}, \; i = 1, \ldots, n\}, \tag{6}$$

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

$$\mathcal{I}_{Y,G}(y^n) = \{f^n(x^n, y^n) \in \mathcal{F}^n : (x_i, y_i) \in E_G, \; x_i \in \mathcal{X}, \; i = 1, \ldots, n\}, \tag{7}$$

with $\mu_G(y^n) := |\mathcal{I}_{Y,G}(y^n)|$. We let $\lambda_G := \max_{x \in \mathcal{X}} \lambda_G(x)$, and $\mu_G := \max_{y \in \mathcal{Y}} \mu_G(y)$. Lastly, for each $G \in \mathcal{G}$, we construct a set

$$\mathcal{S}_G^n = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : (x_i, y_i) \in E_G, \; i = 1, \ldots, n\}. \tag{8}$$

2.1 Function Computation When Both Users Observe the True Distribution

The following known results for the case $p(x,y) = q(x,y)$ are key to our analysis.

Proposition 1. *[2] Given a bipartite graph G constructed from $p(x,y)$ and $f(x,y)$, zero-error decoding requires that: i) the set of codewords for the sequences in $\mathcal{I}_{X,G}(x^n)$ for each $x^n \in \mathcal{X}$ is prefix-free, ii) the set of codewords in $\mathcal{I}_{Y,G}(y^n)$ for each $y^n \in \mathcal{Y}$ is prefix-free.*

The following result provides a lower bound on the worst-case codeword length.

Theorem 1. *[15] Let $x_{\max} \in \mathcal{X}$ be a symbol with $\lambda_G(x_{\max}) = \lambda_G$ and $y_{\max} \in \mathcal{Y}$ with $\mu_G(y_{\max}) = \mu_G$. Denote $x^n_{\max} := (x_{\max}, \ldots, x_{\max})$ and $y^n_{max} := (y_{\max}, \ldots, y_{\max})$. Then,*

$$l_n \geq \max_{\substack{(x^n_{\max}, y^n) \in \mathcal{S}^n \\ (x^n, y^n_{\max}) \in \mathcal{S}^n}} \left\{ \log \lambda_G + \frac{1}{n} \log \mu(y^n),\ \frac{1}{n} \log \lambda(x^n) + \log \mu_G \right\}. \tag{9}$$

Definition 2. *(Characteristic graphs) Define a graph $G_X = (\mathcal{X}, E_X)$ for user 1 with a vertex set $\mathcal{X}$. An edge $(x, \hat{x}) \in E_X$ exists between $x \in \mathcal{X}$ and $\hat{x} \in \mathcal{X}$ whenever some $y \in \mathcal{Y}$ exists such that $(x,y) \in \mathcal{S}^{(1)}$, $(\hat{x},y) \in \mathcal{S}^{(1)}$ and $f(x,y) \neq f(\hat{x},y)$. Similarly, define a characteristic graph $G_Y = (\mathcal{Y}, E_Y)$ for user 2 whose vertices are $y \in \mathcal{Y}$. An edge $(y, \hat{y}) \in E_Y$ exists between $y \in \mathcal{Y}$ and $\hat{y} \in \mathcal{Y}$ whenever some $x \in \mathcal{X}$ exists such that $(x,y) \in \mathcal{S}^{(1)}$, $(x,\hat{y}) \in \mathcal{S}^{(1)}$, and $f(x,y) \neq f(x,\hat{y})$.*

The following result provides an upper bound on the worst-case codeword length.

Theorem 2. *[16] For the worst-case codeword length with one round of interaction,*

$$l_n \leq \frac{1}{n} \left\{ \lceil n \log(\chi(G_X)) \rceil + \lceil n \log(\chi(G_Y)) \rceil \right\}. \tag{10}$$

Although this work focuses on protocols with at most two rounds, our analysis in the sequel can be extended to achievable schemes with a larger number of rounds.

In the context of deterministic and non-deterministic communication complexity, there exist various lower bounds for function computation based on matrix partitions and matrix coverings [17]. Our choice of (9) is motivated by its immediate connection to the graphical structure of the corresponding bipartite graph, which we frequently utilize in our analysis in the sequel.

2.2 Different Approaches to Function Computation with Asymmetric Priors

For zero-error function computation with mismatched information, i.e., when $p(x,y) \neq q(x,y)$, the communication scheme must prevent any inference errors that may occur due to the mismatch. One solution is for user 2 to learn the true distribution from user 1, after which both users can use it for communication. We refer to this strategy as *perfect reconciliation*. Perfect reconciliation allows user 2 to distinguish the true distribution uniquely and the transmission of this information may require additional resources. After reconciliation, however, both users will know the true distribution, which has the potential of reducing the worst-case codeword length. Alternatively, users 1 and 2 may choose not to reconcile the distributions, but to utilize a *worst-case graph* that is agreed upon before communication starts. This graph should always enable zero-error communication for both users. We refer to this scheme as *no reconciliation*. No reconciliation requires no additional bits for learning the graph as users utilize a predetermined graph, but the minimum codeword length for

this graph can be higher than any of the individual graphs in $\mathcal{G}$. Our focus is on identifying this *reconciliation-communication* trade-off, by investigating the conditions under which reconciliation is strictly better than no reconcilation.

Perfect reconciliation and no reconciliation are two extreme cases for tackling the mismatched distribution information. As such, we propose a third scheme called *partial reconciliation*, which allows user 2 to distinguish user 1's graph up to a class of graphs, after which the two users use a worst-case graph that allows for zero-error communication for any confusable graph within that class. Accordingly, partial reconciliation allows some ambiguity in the reconciled set of graphs. By doing so, our goal is to identify the impact of *level of reconciliation* on the worst-case codeword length. We note that perfect reconciliation and no reconciliation schemes are both special cases of the partial reconciliation scheme. The per-symbol worst-case codeword length for a finite block of n source symbols is denoted as $l_{n,\text{R}}$, $l_{n,\text{NR}}$, and $l_{n,\text{PR}}$ for the perfect reconciliation, no reconciliation, and partial reconciliation schemes, respectively. In the remainder of our analysis, we partition $\mathcal{Q}$ into groups of distributions with distinct support sets, and denote by $\mathcal{H}$ the set of distinct bipartite graphs corresponding to each support set as done for $\mathcal{P}$.

Definition 3. *(Reconciliation graph) Consider a characteristic graph $R = (\mathcal{G}, E_R)$, such that each vertex represents a graph $G \in \mathcal{G}$. Define an edge $(G, G') \in E_R$ between vertices G and G' if and only if there exists an $H \in \mathcal{H}$ such that $E_H \subseteq E_G$ and $E_H \subseteq E_{G'}$.*

Then, $\lceil \log \chi(R) \rceil$ is the minimum number of bits required for user 2 to perfectly learn G. We state in the sequel that perfect reconciliation incurs a negligible cost for large blocklengths, which no longer holds for finite blocklengths.

Proposition 2. *Perfect reconciliation is asymptotically optimal.*

Proof. Since $p(x, y)$ and $q(x, y)$ are fixed once selected, reconciliation requires at most $\lceil \log |R| \rceil$ bits for any $\mathcal{G}$. As as result, 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 no reconciliation can never be lower than reconciliation, we observe that reconciling the graphs first, and then using the reconciled graphs for communication, is always better than not reconciling them, i.e.,

$$\lim_{n\to\infty} l_{n,\text{R}} \leq \lim_{n\to\infty} l_{n,\text{NR}}. \tag{11}$$

□

We now study the impact of reconciling the information mismatch on interactive function computation with finite blocklength.

3 Cases in which no reconciliation is better

Let $G_{X,\mathcal{G}} = (\mathcal{X}, E_X)$ be a characteristic graph with a vertex set $\mathcal{X}$, for which an edge $(x, \hat{x}) \in E_X$ exists whenever some $y \in \mathcal{Y}$ exists such that $(x, y) \in \cup_{G\in\mathcal{G}} E_G$, $(\hat{x}, y) \in \cup_{G\in\mathcal{G}} E_G$ and $f(x, y) \neq f(\hat{x}, y)$. On the other hand, let $G_{Y,\mathcal{G}} = (\mathcal{Y}, E_Y)$ be a characteristic graph with a vertex set $\mathcal{Y}$, for which $(y, \hat{y}) \in E_Y$ whenever some $x \in \mathcal{X}$ exists such that $(x, y) \in E_G$, $(x, \hat{y}) \in E_G$ *for some* $G \in \mathcal{G}$ and $f(x, y) \neq f(x, \hat{y})$. We note the difference between the conditions for constructing $G_{X,\mathcal{G}}$ and $G_{Y,\mathcal{G}}$: the former is based on a union of

graphs $\cup_G \in \mathcal{G}$ whereas the latter is based on the existence in some graph $G \in \mathcal{G}$. 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 graphs, whereas user 1 has the true distribution, and can utilize it for eliminating the ambiguities for correct recovery.

Theorem 3. *No reconciliation is always better than perfect reconciliation whenever*

$$\lceil \log \chi(R) \rceil + \max_{G \in \mathcal{G}} \max_{\substack{(x^n_{\max}, y^n) \in \mathcal{S}^n_G \\ (x^n, y^n_{\max}) \in \mathcal{S}^n_G}} \{ n \log \lambda_G + \log \mu_G(y^n),\ \log \lambda_G(x^n) + n \log \mu_G \} \\ > \lceil n \log \chi(G_{X,\mathcal{G}}) \rceil + \lceil n \log \chi(G_{Y,\mathcal{G}}) \rceil, \quad (12)$$

where $x_{\max}$ is a symbol with $|\mathcal{I}_{X,G}(x_{\max})| = \lambda_G$ and $y_{\max}$ is a symbol with $|\mathcal{I}_{Y,G}(y_{\max})| = \mu_G$ that can be obtained from (6) *and* (7)*, respectively, and we use the shorthands $x^n_{\max} = (x_{\max}, \ldots, x_{\max})$ and $y^n_{\max} = (y_{\max}, \ldots, y_{\max})$.*

Proof. We first obtain a lower bound for perfect reconciliation,

$$l_{n,\mathrm{R}} \geq \frac{\lceil \log \chi(R) \rceil}{n} + \frac{1}{n} \max_{G \in \mathcal{G}} \max_{\substack{(x^n_{\max}, y^n) \in \mathcal{S}^n_G \\ (x^n, y^n_{\max}) \in \mathcal{S}^n_G}} \{ n \log \lambda_G + \log \mu_G(y^n),\ \log \lambda_G(x^n) + n \log \mu_G \}. \quad (13)$$

This follows from the fact that at least $\lceil \log \chi(R) \rceil$ bits are required for reconciling any graph in $\mathcal{G}$, and then applying Theorem 1 to the reconciled graph $G \in \mathcal{G}$. For the no reconciliation case, observe that using a union graph $\cup_{G \in \mathcal{G}} G$ as the worst-case graph always enables zero-error communication for user 2. User 1 knows the true distribution and can distinguish any function value as long as no two $y, y' \in \mathcal{Y}$ are assigned to the same codeword such that there exists a $x \in \mathcal{X}$ for which both (x, y) and (x, y') are edges in the bipartite graph known by user 1 with $f(x, y) \neq f(x, y')$. Then, from (10) we obtain the upper bound

$$l_{n,\mathrm{NR}} \leq \frac{1}{n} \left(\lceil n \log(\chi(G_{X,\mathcal{G}})) \rceil + \lceil n \log(\chi(G_{Y,\mathcal{G}})) \rceil \right). \quad (14)$$

where user 1 sends $\lceil n \log(\chi(G_{X,\mathcal{G}})) \rceil$ bits to user 2 whereas user 2 sends $\lceil n \log(\chi(G_{Y,\mathcal{G}})) \rceil$ bits to user 1. □

We note that whenever there exists a "parent" graph that subsumes any other graph in $\mathcal{G}$, reconciliation is strictly suboptimal. To this end, consider a class of graphs for which there exists some $G^* \in \mathcal{G}$ such that $E_G \subseteq E_{G^*}$ for all $G \in \mathcal{G}$. Then, reconciliation is always worse than no reconciliation. This immediately follows from: i) any zero-error communication strategy for G^* is a valid no reconciliation scheme, since $\cup_{G \in \mathcal{G}} G = G^*$, ii) any perfect reconciliation scheme should use a valid zero-error communication strategy for G^*, as it may appear at user 1. Therefore, reconciling graphs could only increase the codeword length, and is useless. An example to this case is when G^* is a complete graph.

4 Cases in which partial reconciliation is (strictly) better

We now investigate the conditions under which partially reconciling the graph information is better than perfect reconciliation. To do so, we compare the performance of a given partial reconciliation scheme with the perfect reconciliation scheme. Let $\mathcal{V}$ define the set of all possible partitions of $\mathcal{G}$, and $V \in \mathcal{V}$ be a partition of $\mathcal{G}$ such that $\cup_{V_i \in V} V_i = \mathcal{G}$ and

$V_i \cap V_j = \emptyset$ for any $V_i, V_j \in V$ such that $i \neq j$. Let G_{X,V_i} be a characteristic graph defined in the same way as $G_{X,\mathcal{G}}$ but by using $\cup_{G\in V_i} G$ instead of $\cup_{G\in\mathcal{G}} G$. Moreover, let G_{Y,V_i} be a characteristic graph with a vertex set $\mathcal{Y}$, for which $(y,\hat{y}) \in E_Y$ whenever some $x \in \mathcal{X}$ exists such that $(x,y) \in E_G$, $(x,\hat{y}) \in E_G$ *for some* $G \in V_i$ with $f(x,y) \neq f(x,\hat{y})$. We note the procedure for constructing G_{Y,V_i} is along the lines of the procedure for $G_{Y,\mathcal{G}}$, but by using the group of graphs $G \in V_i$ instead of the set of all possible graphs $G \in \mathcal{G}$.

Theorem 4. *The partial reconciliation strategy $V \in \mathcal{V}$ is always better than perfect reconciliation whenever*

$$\lceil \log \chi(R) \rceil + \max_{G\in\mathcal{G}} \max_{\substack{(x^n_{\max},y^n)\in\mathcal{S}^n_G \\ (x^n,y^n_{\max})\in\mathcal{S}^n_G}} \{n \log \lambda_G + \log \mu_G(y^n),\ \log \lambda_G(x^n) + n \log \mu_G\} \\ > \lceil \log |V| \rceil + \max_{i:V_i\in V} \{\lceil n \log(\chi(G_{X,V_i})) \rceil + \lceil n \log(\chi(G_{Y,V_i})) \rceil\}. \quad (15)$$

Proof. The right-hand side of (15) holds since $\lceil \log |V| \rceil$ bits are sufficient for sending each group index, and for any $V_i \in V$, zero-error recovery can be satisfied by using the graphs in V_i, instead of $\mathcal{G}$ in (14), by using no more than $\lceil n \log(\chi(G_{X,V_i})) \rceil + \lceil n \log(\chi(G_{Y,V_i})) \rceil$ bits. As a result, the right-hand side of (15) is an upper bound on the zero-error codeword length when the graphs are partially reconciled between the two users, whereas the left-hand side lower bounds the zero-error codeword length for perfect reconciliation as in (13), from which (15) follows. □

5 Numerical examples via z-graphs

In this section, we investigate a special class of graphs for which upper and lower bounds on the zero-error codeword length are tight. We then leverage this property to determine the optimal two-stage reconciliation and communication strategies for $n = 1$.

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, this class of graphs has a structure that resembles a z-shape, hence we refer to them as z-graphs.

Theorem 5. *Consider a class of z-graphs for which some $H \in \mathcal{H}$ exists such that $E_H \subseteq E_G$ for all $G \in \mathcal{G}$. Then,*

$$l_{1,PR} = \min_{V\in\mathcal{V}} \left(\lceil \log |V| \rceil + \max_{i:V_i\in V} \left(\lceil \log \chi(G_{Y,V_i}) \rceil + \lceil \log \mu_{V_i} \rceil \right) \right) \quad (16)$$

where $\lceil \log \chi(G_{Y,V_i}) \rceil$ is defined in Section 4 and $\mu_{V_i} = \max_{y\in\mathcal{Y}} |\cup_{G\in V_i} \mathcal{I}_{Y,G}(y)|$ with notations as in (6) *and* (7).

Proof. To prove achievability, note that for a given partition V, at least $\lceil \log |V| \rceil$ bits are necessary for sending the group index, which reconciles each graph up to the class of graphs in the group it is assigned to. After reconciliation, zero-error communication requires $\max_{i:V_i\in V} (\lceil \log \chi(G_{Y,V_i}) \rceil + \lceil \log \mu_{V_i} \rceil)$ in the worst-case.

We show this is sufficient by first considering an encoding that ensures zero-error communication for any graph in V_i by using $(\lceil \log \chi(G_{Y,V_i}) \rceil + \lceil \log \mu_{V_i} \rceil)$ bits. Group all the neighbors $x' \in \mathcal{X}$ of y in $\cup_{G\in V_i} G$ that lead to the same function value $f(x',y)$. Assign

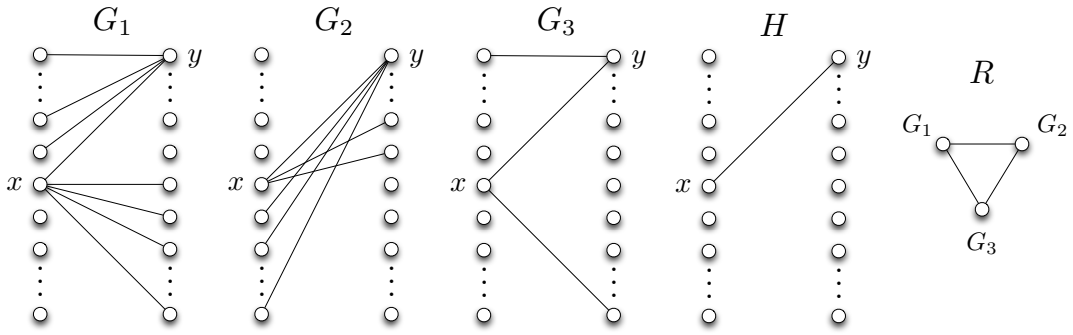

Figure 1: Mismatched graphs $\mathcal{G} = \{G_1, G_2, G_3\}$, $\mathcal{H} = \{H\}$, and the reconciliation graph R.

a single distinct codeword to each of these groups. Note that this requires $\lceil \log \mu_{V_i} \rceil$ bits in total. Next, for a given V_i, construct the graph G_{Y,V_i} as defined in Section 4. Find the minimum coloring of G_{Y,V_i}, and assign a distinct codeword to each of the colors, which requires $\lceil \log \chi(G_{Y,V_i}) \rceil$ bits in total. Finally, we show that any such partition corresponds to a valid encoding scheme. Suppose that this is not the case, and two symbols x', x'' with the same neighbor y in $\cup_{G \in V_i} G$ such that $f(x', y) \neq f(x'', y)$ are assigned to the same codeword. User 2 then can not distinguish between these two symbols that lead to different function values for at least some pairs of possible graphs, since user 2 does not know the true distribution, causing an error and violating the zero-error condition. On the other hand, assume that two symbols y', y'' with the same neighbor x in some $G \in V_i$ for which $f(x, y') \neq f(x, y'')$ are assigned to the same codeword. Then, user 1 will not be able to distinguish between $f(x, y') \neq f(x, y'')$, i.e., the correct function values, for at least some graphs that may occur at user 1. Therefore, by contradiction, (16) provides the optimal strategy for communicating z-graphs. □

Example 1. Consider the set of z-graphs $\mathcal{G} = \{G_1, G_2, G_3\}$ and $\mathcal{H} = \{H\}$ in Fig. 1. The edge sets satisfy $E_{G_3} \subset E_{G_1}$, $E_{G_2} \cap E_{G_1} = \{(x, y)\}$, and $E_H = \{(x, y)\}$. Assume that $f(x, y)$ is distinct for each edge $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Then,

$$l_{1,\text{NR}} \geq \lceil \log \mu_{\mathcal{G}} \rceil + \max_{G \in \mathcal{G}} \lceil \log \lambda_G \rceil = \lceil \log(\mu_{G_1} + \mu_{G_2} - 1) \rceil + \max\{\lceil \log \lambda_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil\} \quad (17)$$

which can be obtained by inspecting (16) and considering a single partition, $V = \{V_1\}$, where $V_1 = \mathcal{G}$. For the perfect reconciliation scheme, we construct the reconciliation graph R as given in Fig. 1, 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,

$$l_{1,\text{R}} = \lceil \log 3 \rceil + \max_{G_i \in \mathcal{G}} (\lceil \log \lambda_{G_i} \rceil + \lceil \log \mu_{G_i} \rceil) \quad (18)$$

which also follows from (16). Lastly, consider a partial reconciliation scheme achieved by the partition $V = \{V_1, V_2\}$ such that $V_1 = \{G_1, G_3\}$, and $V_2 = \{G_2\}$. Then, from (16),

$$l_{1,\text{PR}} = \log 2 + \max\{\lceil \log \lambda_{G_1} \rceil + \lceil \log \mu_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil + \lceil \log \mu_{G_2} \rceil\}. \quad (19)$$

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

$$\log 2 + \max\{\lceil \log \lambda_{G_1} \rceil + \lceil \log \mu_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil + \lceil \log \mu_{G_2} \rceil\} < \lceil \log(\mu_{G_1} + \mu_{G_2} - 1) \rceil + \max\{\lceil \log \lambda_{G_1} \rceil, \lceil \log \lambda_{G_2} \rceil\}, \quad (20)$$

partial reconciliation outperforms both perfect and no reconciliation. A possible assignment is $\lambda_{G_1} = \mu_{G_2} = 8, \lambda_{G_2} = \mu_{G_1} = 4$, for which $l_{1,\text{PR}} = 6$ but $l_{1,\text{R}} = 7$ and $l_{1,\text{NR}} \geq 7$.

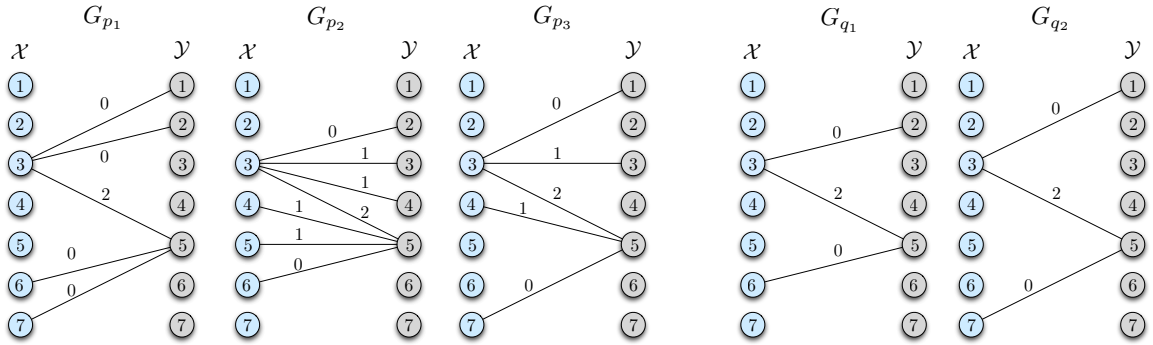

Figure 2: Bipartite graphs G_{p_1} for $p_1(x,y)$, G_{p_2} for $p_2(x,y)$, G_{p_3} for $p_3(x,y)$, G_{q_1} for $q_1(x,y)$, and G_{q_2} for $q_2(x,y)$. Edge labels represent the function values from (24). As an example, for G_{p_1}, maximum vertex degree is 3 for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$, but $\lambda_{G_{p_1}} = \mu_{G_{p_1}} = 2$.

Example 2. Consider two users where user 1 observes $x \in \mathcal{X} = \{1, \ldots, 7\}$ and user 2 observes $y \in \mathcal{Y} = \{1, \ldots, 7\}$ from a distribution $p(x,y)$ in $\mathcal{P} = \{p_1, p_2, p_3\}$ such that

$$p_1(x,y) = \begin{cases} 1/5 & \text{if } (x,y) \in \{(3,1),(3,2),(3,5),(6,5),(7,5)\} \\ 0 & \text{o.w.} \end{cases} \tag{21}$$

whereas

$$p_2(x,y) = \begin{cases} 1/7 & \text{if } (x,y) \in \{(3,2),(3,3),(3,4),(3,5),(4,5),(5,5),(6,5)\} \\ 0 & \text{o.w.} \end{cases} \tag{22}$$

and

$$p_3(x,y) = \begin{cases} 1/5 & \text{if } (x,y) \in \{(3,1),(3,3),(3,5),(4,5),(7,5)\} \\ 0 & \text{o.w.} \end{cases}. \tag{23}$$

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

$$f(x,y) = \begin{cases} 0 & \text{if } x - y > 0 \\ 1 & \text{if } -1 \leq x - y \leq 0 \\ 2 & \text{o.w.} \end{cases}. \tag{24}$$

User 2 assumes a distribution $q(x,y)$ from a set $\mathcal{Q} = \{q_1, q_2\}$, where

$$q_1(x,y) = \begin{cases} 1/3 & \text{if } (x,y) \in \{(3,2),(3,5),(6,5)\} \\ 0 & \text{o.w.} \end{cases} \tag{25}$$

and

$$q_2(x,y) = \begin{cases} 1/3 & \text{if } (x,y) \in \{(3,1),(3,5),(7,5)\} \\ 0 & \text{o.w.} \end{cases}. \tag{26}$$

Note that all of the G and H graphs are z-graphs. Using the no reconciliation scheme, user 1 sends an index "0" if $x \in \{6,7\}$, a "1" if $y \in \{4,5\}$, and a "2" otherwise, by using $\lceil \log 3 \rceil = 2$ bits. After receiving the index, user 2 can recover $f(x,y)$ perfectly, whether 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 $f(x,y)$. This protocol takes 4 bits. On the other hand, from (9), we find the lower bound

$\log 3 + \log 3 = 3.17$ on the message length. Hence, at least 4 bits need to be exchanged, therefore no reconciliation is optimal.

6 Conclusion

In this paper, we considered interactive function computation when the knowledge of the source distribution is mismatched at the two sides. We investigated the impact of reconciling mismatched information on the worst-case zero-error codeword length. We identified a reconciliation-communication tradeoff and studied the conditions under which perfectly or partially reconciling the information mismatch is better or worse than no reconciliation. Future work includes the interactive compression of logical descriptions, analysis of non-zero error protocols, interactive reconciliation and average message length.

References

[1] A. El Gamal and A. Orlitsky, "Interactive data compression," in *IEEE Symp. on Foundations of Computer Science (FOCS'84)*, 1984, pp. 100–108.

[2] A. Orlitsky, "Worst-case interactive communication i: Two messages are almost optimal," *IEEE Trans. on Inf. Theory*, vol. 36, no. 5, pp. 1111–1126, 1990.

[3] E.-H. Yang and D.-K. He, "Interactive encoding and decoding for one way learning: Near lossless recovery with side information at the decoder," *IEEE Trans. on Inf. Theory*, vol. 56, no. 4, pp. 1808–1824, 2010.

[4] A. C. Yao, "Some complexity questions related to distributed computing," in *ACM Symp. on Theory of Computing (STOC'79)*, 1979, pp. 209–213.

[5] T. Feder, E. Kushilevitz, M. Naor, and N. Nisan, "Amortized communication complexity," *SIAM Journal on Computing*, vol. 24, no. 4, pp. 736–750, 1995.

[6] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. on Inf. Theory*, vol. 47, no. 3, pp. 903–917, 2001.

[7] N. Ma and P. Ishwar, "Some results on distributed source coding for interactive function computation," *IEEE Trans. on Inf. Theory*, vol. 57, no. 9, pp. 6180–6195, 2011.

[8] J. Chen, D.-K. He, and A. Jagmohan, "On the duality between Slepian–Wolf coding and channel coding under mismatched decoding," *IEEE Trans. on Inf. Theory*, vol. 55, no. 9, pp. 4006–4018, 2009.

[9] B. Juba, A. T. Kalai, S. Khanna, and M. Sudan, "Compression without a common prior: an information-theoretic justification for ambiguity in language," *Innovations in Computer Science (ICS)*, 2011.

[10] E. Haramaty and M. Sudan, "Deterministic compression with uncertain priors," in *5th Conf. on Innovations in Theoretical Computer Science (ITCS'14)*, 2014, pp. 377–386.

[11] H. S. Witsenhausen, "The zero-error side information problem and chromatic numbers," *IEEE Trans. on Inf. Theory*, vol. 22, no. 5, pp. 592–593, 1976.

[12] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Trans. on Inf. Theory*, vol. 49, no. 9, pp. 2213–2218, 2003.

[13] J. Nayak and K. Rose, "Graph capacities and zero-error transmission over compound channels," *IEEE Trans. on Inf. Theory*, vol. 51, no. 12, pp. 4374–4378, 2005.

[14] V. Doshi, D. Shah, M. Médard, and M. Effros, "Functional compression through graph coloring," *IEEE Trans. on Inf. Theory*, vol. 56, no. 8, pp. 3901–3917, 2010.

[15] B. Guler and A. Yener, "Compressing semantic information with varying priorities," in *IEEE Data Compression Conf. (DCC'14)*, 2014, pp. 213–222.

[16] B. Guler, A. Yener, and P. Basu, "A study of semantic data compression," in *IEEE Global Conf. on Signal and Information Processing (GlobalSIP'13)*, 2013, pp. 887–890.

[17] E. Kushilevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 2006.