On Query-To-Communication Lifting for Adversary Bounds

Anurag Anshu
EECS & Challenge Institute for Quantum Computation,
University of California, Berkeley, CA, USA
Simons Institute for the Theory of Computing, Berkeley, CA, USA

Shalev Ben-David
University of Waterloo, Canada

Srijita Kundu
Centre for Quantum Technologies, National University of Singapore, Singapore

Abstract

We investigate query-to-communication lifting theorems for models related to the quantum adversary bounds. Our results are as follows:

1. We show that the classical adversary bound lifts to a lower bound on randomized communication complexity with a constant-sized gadget. We also show that the classical adversary bound is a strictly stronger lower bound technique than the previously-lifted measure known as critical block sensitivity, making our lifting theorem one of the strongest lifting theorems for randomized communication complexity using a constant-sized gadget.
2. Turning to quantum models, we show a connection between lifting theorems for quantum adversary bounds and secure 2-party quantum computation in a certain "honest-but-curious" model. Under the assumption that such secure 2-party computation is impossible, we show that a simplified version of the positive-weight adversary bound lifts to a quantum communication lower bound using a constant-sized gadget. We also give an unconditional lifting theorem which lower bounds bounded-round quantum communication protocols.
3. Finally, we give some new results in query complexity. We show that the classical adversary and the positive-weight quantum adversary are quadratically related. We also show that the positive-weight quantum adversary is never larger than the square of the approximate degree. Both relations hold even for partial functions.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum computing, query complexity, communication complexity, lifting theorems, adversary method

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.30

Related Version *Full Version*: https://arxiv.org/abs/2012.03415

Funding *Anurag Anshu*: Supported by the NSF QLCI program through grant number OMA-2016245. Part of this work was done when A.A. was affiliated to the Department of Combinatorics & Optimization and Institute for Quantum Computing, University of Waterloo, as well as to the Perimeter Institute for Theoretical Physics.
Shalev Ben-David: Supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), DGECR-2019-00027 and RGPIN-2019-04804[1].
Srijita Kundu: Supported by the National Research Foundation, including under NRF RF Award No. NRF-NRFF2013-13, the Prime Minister's Office, Singapore; the Ministry of Education, Singapore, under the Research Centres of Excellence program and by Grant No. MOE2012-T3-1-009; and in part by the NRF2017-NRF-ANR004 VanQuTe Grant. Part of this work was done when S.K. was visiting the Institute of Quantum Computing, University of Waterloo.

[1] Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), DGECR-2019-00027 et RGPIN-2019-04804.

Acknowledgements We thank Rahul Jain and Dave Touchette for helpful discussions related to the $\mathrm{QICZ}(G) > 0$ conjecture. We thank Robin Kothari for helpful discussions related to the adversary bounds. We thank Anne Broadbent for helpful discussions related to quantum secure 2-party computation. We thank Mika Göös for helpful discussions regarding critical block sensitivity and its lifting theorem. We thank Jevgēnijs Vihrovs and the other authors of [6] for helpful discussions regarding the classical adversary method, and particularly Krišjānis Prūsis for the proof of Lemma 27.

1 Introduction

Communication complexity is an important model of computation with deep connections to many parts of theoretical computer science [29]. In communication complexity, two parties, called Alice and Bob, receive inputs x and y from sets $\mathcal{X}$ and $\mathcal{Y}$ respectively, and wish to compute some joint function $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ on their inputs. Alice and Bob cooperate together, and their goal is to minimize the number of bits they must exchange before determining $F(x,y)$.

Recently, a lot of attention has been devoted to connections between communication complexity and query complexity. In particular, query-to-communication "lifting" theorems are powerful tools which convert lower bounds in query complexity into lower bounds in communication complexity in a black-box manner. Since query lower bounds are typically much easier to prove than communication lower bounds, these tools are highly useful for the study of communication complexity, and often come together with new communication complexity results (such as separations between different communication complexity models). For example, see [21, 22, 23, 25, 16].

Lifting theorems are known for many models of computation, including deterministic [23] and randomized [25] algorithms. Notably, however, a lifting theorem for quantum query complexity is not known; the closest thing available is a lifting theorem for approximate degree (also known as the polynomial method), which lifts to approximate logrank [37]. This allows quantum query lower bounds proved via the polynomial method to be turned into quantum communication lower bounds, but a similar statement is not known even for the positive-weight quantum adversary method [5, 39].

In this work, we investigate lifting theorems for the adversary method and related models. We prove a lifting theorem for a measure called the classical adversary bound. For the quantum adversary method, we show that there is a surprising connection with the cryptographic notion of secure 2-party computation. Specifically, we show that a lifting theorem for a simplified version of the positive-weight adversary method follows from a plausible conjecture regarding the impossibility of secure 2-party computation in a certain "honest but curious" quantum model. We also prove an unconditional lifting theorem which lower bounds bounded-round quantum algorithms.

Finally, we prove some query complexity results that may be of independent interest: first, a quadratic relationship between the positive-weight adversary bound and the classical adversary bound; and second, we show that the positive-weight adversary bound can never be larger than the square of the approximate degree. This means that the (positive) adversary method can never beat the polynomial method by more than a quadratic factor. These results hold even for partial functions.

1.1 Lifting theorems

The statement of a lifting theorem typically has the following form:

$$\mathrm{M}^{cc}(f \circ G) = \Omega(\mathrm{M}(f)).$$

Here $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is a (fixed) communication complexity function, called a "gadget", which typically has low communication cost; $f\colon \{0,1\}^n \to \{0,1\}$ is an arbitrary Boolean function; $\mathrm{M}(\cdot)$ is a measure in query complexity, representing the cost of computing the function f in query complexity; and $\mathrm{M}^{cc}(\cdot)$ is a measure in communication complexity. The notation $f \circ G$ denotes the block-composition of f with G. This is a communication complexity function defined as follows: Alice gets input $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$, Bob gets input $(y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n$, and they must output $f(G(x_1, y_1), G(x_2, y_2), \ldots, G(x_n, y_n))$. Hence $f \circ G$ is a function with signature $\mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}$.

There are two primary types of lifting theorems: those that work with a constant-sized gadget G (independent of f), and those that work with a gadget G whose size logarithmic in the input size n of f. [2] The latter type tend to be much more prevalent; recent lifting theorems for deterministic and randomized communication complexities all use log-sized or larger gadgets [23, 42, 17, 25, 16]. We remark, however, that even with log-sized or larger gadgets, lifting theorems are highly nontrivial to prove: a lifting theorem for BPP, which lifts randomized query lower bounds to randomized communication lower bounds, was only established in the last few years, while an analogous result for BQP remains an open problem.

Lifting theorems which work with a constant-sized gadget are even harder to prove, but often turn out to be much more useful. The reason is that common function families, like disjointness (which we denote $\textsc{Disj}_n$) [3] or inner product (which we denote IP_n) [4], are *universal*. This means for every communication function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, its communication matrix (that is, its truth table) is a submatrix of the communication matrix of a sufficiently large instance of the $\textsc{Disj}$ function. In other words, every communication function is a sub-function of $\textsc{Disj}_k$ and IP_k for sufficiently large k. If the size of G is constant, then it is necessarily contained in a $\textsc{Disj}$ function of *constant* size (and similarly for IP). Hence lifting with any constant-sized gadget G is enough to guarantee a lifting theorem with a constant-sized disjointness gadget and constant-sized inner product gadget (and similarly for every other universal function family). In short, lifting with a constant-sized gadget implies lifting with almost any gadget of your choice.

In particular, a lifting theorem with a constant-sized gadget immediately implies a lower bound for $\textsc{Disj}_n$ and IP_n themselves. To see this, suppose we had a lifting theorem

$$\mathrm{M}^{cc}(f \circ G) = \Omega(\mathrm{M}(f))$$

for all Boolean functions f and a fixed (constant-sized) communication gadget G. Then G is a sub-function of $\textsc{Disj}_k$ and of IP_k for some constant k. Note that $\textsc{Disj}_n = \mathrm{OR}_{n/k} \circ \textsc{Disj}_k$ and that $\mathrm{IP}_n = \textsc{Parity}_{n/k} \circ \mathrm{IP}_k$. Hence we get $\mathrm{M}^{cc}(\textsc{Disj}_n) = \Omega(\mathrm{M}(\mathrm{OR}_{n/k}))$ and $\mathrm{M}^{cc}(\mathrm{IP}_n) = \Omega(\mathrm{M}(\textsc{Parity}_{n/k}))$. Since k is constant, this can potentially give lower bounds on $\mathrm{M}^{cc}(\textsc{Disj}_n)$ and on $\mathrm{M}^{cc}(\mathrm{IP}_n)$ that are tight up to constant factors, depending on the measures $\mathrm{M}^{cc}(\cdot)$ and $\mathrm{M}(\cdot)$.

[2] Sometimes, lifting theorems use a gadget G which is large – polynomial in n – but which can still be computed using $O(\log n)$ communication.

[3] In the disjointness function, Alice and Bob receive n-bit strings x and y and must output 1 if and only if there exists an index $i \in [n]$ such that $x_i = y_i = 1$.

[4] In the inner product function, Alice and Bob receive n-bit strings x and y, and must compute inner product of those strings over $\mathbb{F}_2$.

There have only been a handful of lifting theorems which work with constant-sized gadgets. One such result follows from Sherstov's work for approximate degree and related measures [37]. The part of that work which is most relevant to us is the lifting of approximate degree to lower bounds on approximate logrank, and hence on the quantum communication complexity of the lifted function. Sherstov's work means that if one can prove a quantum lower bound for a query function f using the *polynomial method* [9], then this lower bound will also apply to the quantum communication complexity of $f \circ G$, where G is a constant sized gadget. Such a lifting theorem is not known to hold for the adversary methods [5, 39], however (not even with a log-sized gadget).

Another lifting theorem with a constant-sized gadget appears in [27, 24]. There, a query measure called *critical block sensitivity* [27] is lifted to a lower bound on randomized communication complexity.

1.2 Adversary methods

The quantum adversary bounds are extremely useful methods for lower bounding quantum query complexity. The original adversary method was introduced by Ambainis [5]. It was later generalized in several ways, which were shown to all be equivalent [39], and are known as the positive-weight adversary bound, denoted $\mathrm{Adv}(f)$. This bound has many convenient properties: it has many equivalent formulations (among them a semidefinite program), it is reasonably easy to use in practice, and it behaves nicely under many operations, such as composition. The positive-weight adversary bound is one of the most commonly used techniques for lower bounding quantum query complexity.

A related measure is called the negative-weight adversary bound, introduced in [26], which we denote by $\mathrm{Adv}^{\pm}(f)$. This is a strengthening of the positive adversary bound, and satisfies $\mathrm{Adv}^{\pm}(f) \geq \mathrm{Adv}(f)$ for all (possibly partial) Boolean functions f. Surprisingly, in [35, 32], it was shown that the negative-weight adversary is actually *equal* to quantum query complexity up to constant factors.

The quantum adversary methods have no known communication complexity analogues. However, that by itself does not rule out a lifting theorem: one might still hope to lift $\mathrm{Adv}(f)$ or $\mathrm{Adv}^{\pm}(f)$ to lower bounds on quantum communication complexity, similar to how critical block sensitivity $\mathrm{cbs}(f)$ was lifted to a lower bound on randomized communication complexity [27, 24]. Unfortunately, no such lifting theorems are currently known, not even for the positive-weight adversary method, and not even with a large gadget size.

Interestingly, it is possible to define a lower bound technique for *randomized* algorithms which is motivated by the (positive) quantum adversary method. This measure was first introduced in [1, 30], and different variants of it have been subsequently studied [6]. Here, we use the largest of these variants, which we denote by $\mathrm{CAdv}(f)$ (in [6], it was denoted by $\mathrm{CMM}(f)$). In [6], it was shown that for total functions f, $\mathrm{CAdv}(f)$ is (up to constant factors) equal to a measure called fractional block sensitivity, which we denote $\mathrm{fbs}(f)$. However, for partial functions, there can be a large separation between the two measures. For more on fractional block sensitivity, see [2, 28].

1.3 Our contributions

Lifting the classical adversary

Our first contribution is a lifting theorem for the classical adversary bound $\mathrm{CAdv}(f)$. We lift it to a lower bound on randomized communication complexity using a constant-sized gadget.

▶ **Theorem 1.** *There is an explicitly given function $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{RCC}(f\circ G)=\Omega(\mathrm{CAdv}(f)).$$

Here $\mathrm{RCC}(f\circ G)$ denotes the randomized communication complexity of $f\circ G$ with shared randomness. We note that [27, 24] provided a lifting theorem that has a similar form, only with the measure $\mathrm{cbs}(f)$ in place of $\mathrm{CAdv}(f)$. To compare the two theorems, we should compare the two query measures. We have the following theorem.

▶ **Lemma 2.** *For all (possibly partial) Boolean functions or relations f, $\mathrm{CAdv}(f)=\Omega(\mathrm{cbs}(f))$. Moreover, there is a family of total functions f for which $\mathrm{CAdv}(f)=\Omega(\mathrm{cbs}(f)^{3/2})$.*

Lemma 2 says that $\mathrm{CAdv}(f)$ is a strictly stronger lower bound technique than $\mathrm{cbs}(f)$, and hence Theorem 1 is stronger than the lifting theorem of [24]. A proof of Lemma 2 follows from our proof that $\mathrm{CAdv}(f)\ge \mathrm{cfbs}(f)/2$ in Lemma 26, together with the known power 3/2 separation between $\mathrm{fbs}(f)$ and $\mathrm{bs}(f)$ for total functions [20].[5] This makes Theorem 1 one of the strongest known lifting theorems for randomized communication complexity which works with a constant-sized gadget.[6]

We note that the lifting theorem of [24] for the measure $\mathrm{cbs}(f)$ also works when f is a *relation*, which is a more general setting than partial functions; indeed, most of their applications for the lifting theorem were for relations f rather than functions. We extend Theorem 1 to relations as well, and also show that $\mathrm{CAdv}(f)=\Omega(\mathrm{cbs}(f))$ for all relations. In fact, it turns out that for partial functions, $\mathrm{CAdv}(f)$ is equal to a fractional version of $\mathrm{cbs}(f)$, which we denote $\mathrm{cfbs}(f)$; however, for relations, $\mathrm{CAdv}(f)$ is a stronger lower bound technique than $\mathrm{cfbs}(f)$ (which in turn is stronger than $\mathrm{cbs}(f)$). We also note that our techniques for lifting $\mathrm{CAdv}(f)$ are substantially different from those of [27, 24].

Lifting quantum measures

Our first quantum result says that $\mathrm{CAdv}(f)$ lifts to a lower bound on bounded-round quantum communication protocols. This may seem surprising, as $\mathrm{CAdv}(f)$ does not lower bound quantum algorithms in query complexity; however, one can show that $\mathrm{CAdv}(f)$ does lower bound *non-adaptive* quantum query complexity, or even quantum query algorithms with limited adaptivity. This motivates the following result.

▶ **Theorem 3.** *There is an explicitly given function $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{QCC}^r(f\circ G)=\Omega(\mathrm{CAdv}(f)/r^2).$$

Here $\mathrm{QCC}^r(\cdot)$ denotes the quantum communication complexity for an r-round quantum protocol with shared entanglement.

We note that since any r-round protocol has communication cost at least r, we actually get a lower bound of $\mathrm{CAdv}(f)/r^2+r$. Minimizing over r yields a lower bound of $\mathrm{CAdv}(f)^{1/3}$ even on unbounded-round protocols. This may not seem very useful, since $\mathrm{CAdv}(f)^{1/3}$ is

[5] For total functions, we have $\mathrm{fbs}(f)=\mathrm{cfbs}(f)$ and $\mathrm{cbs}(f)=\mathrm{bs}(f)$.
[6] Sherstov's lifting theorem for approximate degree [37] also works with a constant-sized gadget, and is incomparable to our result as a lower bound technique for randomized communication complexity.

smaller than $\widetilde{\deg}(f)$, a measure we know how to lift [37]. However, we can generalize this result to relations. For relations, we do not know how to compare $\mathrm{CAdv}(f)^{1/3}$ to $\widetilde{\deg}(f)$, and therefore our lifting theorem gives something new, even in the unbounded-round setting.

▶ **Corollary 4.** *There is an explicitly given function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{QCC}(f \circ G) = \Omega(\mathrm{CAdv}(f)^{1/3}),$$

where QCC *denotes the quantum communication complexity with shared entanglement.*

We next turn our attention to lower bounding unbounded-round quantum communication protocols by lifting a quantum adversary method. Instead of aiming for the positive-weight adversary bound, we work with a simplified version, studied in [3], which we denote $\mathrm{Adv}_1(f)$. This measure is a restriction of Adv to a pairs of inputs with a single bit of difference.

We have $\mathrm{Adv}_1(f) \leq \mathrm{Adv}(f)$, and [3] showed that $\mathrm{Adv}_1(f) = O(\widetilde{\deg}(f))$. However, their proof of the latter is tricky, and we do not use it here; we give a direct lifting of $\mathrm{Adv}_1(f)$ (under a certain assumption), and we argue that the techniques we use are likely to generalize to lifting $\mathrm{Adv}(f)$ in the future.

We prove the following theorem, which lifts $\mathrm{Adv}_1(f)$ but has a dependence on a new complexity measure $\mathrm{QICZ}(G)$ that we introduce.

▶ **Theorem 5.** *For any (possibly partial) Boolean function or relation f and any communication function G which contains both AND_2 and OR_2 as subfunctions, we have*

$$\mathrm{QCC}(f \circ G) = \Omega(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

At first glance, this theorem might look very strong: not only does it lift the simplified adversary bound for a single gadget G, it even does so for *all* G and gives an explicit dependence on G. Unfortunately, there is a catch: the measure $\mathrm{QICZ}(G)$ may be 0 for some communication functions G. In fact, we cannot rule out the possibility that $\mathrm{QICZ}(G) = 0$ for *all* communication functions G, in which case Theorem 5 does not say anything. On the other hand, note that if $\mathrm{QICZ}(G) > 0$ for even a single function G, then Theorem 5 gives a lifting theorem for $\mathrm{Adv}_1(f)$ with a constant-sized gadget, which works even for relations.

We give an interpretation of the measure $\mathrm{QICZ}(G)$ in terms of a cryptographic primitive called secure 2-party computation. In such a primitive, Alice and Bob want to compute a function G on their inputs x and y, but they do not want to reveal their inputs to the other party. Indeed, Alice wants to hide everything about x from Bob and Bob wants to hide everything about y from Alice, with the exception of the final function value $G(x, y)$ (which they are both expected to know at the end of the protocol). We also seek information-theoretic security: there are no limits on the computational power of Alice and Bob. Since we are interested in a quantum version, we will allow Alice and Bob to exchange quantum communication rather than classical communication, potentially with shared entanglement.

Secure 2-party computation is known to be impossible in general, even quantumly [33, 18, 14, 19, 36]. However, in our case, we care about an "honest but curious" version of the primitive, in which Alice and Bob trust each other to execute the protocol faithfully, but they still do not trust each other not to try to learn the others' input. In the quantum setting, it is a bit difficult to define such an honest-but-curious model: after all, if Alice and Bob are honest, they might be forbidden by the protocol from ever executing intermediate measurements, and the protocol might even tell them to "uncompute" everything except for the final answer, to ensure all other information gets deleted. Hence it would seem that honest parties can trivially do secure 2-party computation.

The way we will define quantum secure two-party computation in the honest-but-curious setting will be analogous to the information-based classical definition (see, for example, [13]). Classically, the information leak that Alice and Bob must suffer in an honest execution of the best possible protocol is captured by $\mathrm{IC}(G)$, the information cost of the function G. The measure $\mathrm{IC}(G)$ is the amount Alice learns about Bob's input plus the amount Bob learns about Alice's input, given the best possible protocol and the worst possible distribution over the inputs; we note that this measure includes the value of $G(x, y)$ as part of what Alice and Bob learn about each others' inputs, whereas secure two-party computation does not count learning $G(x, y)$ as part of the cost, but this is only a difference of at most 2 bits of information (one on Alice's side and one on Bob's side); hence, up to an additive factor of 2, $\mathrm{IC}(G)$ captures the information leak necessary in a two-party protocol computing G.

For a quantum version of this, we will use QIC, a measure which is a quantum analogue of IC and which was introduced in [41]. However, we note that if Alice and Bob send the same bit $G(x, y)$ back and forth n times, this will add $\Theta(n)$ to the value of QIC for that protocol, due to subtleties in the definition of QIC (this does not occur classically with IC). Hence, in the quantum setting, QIC does not capture the two-party information leak as cleanly as IC did classically.

Instead, we modify the definition of QIC to a measure we denote $\mathrm{QICZ}(G)$. For this measure, Alice and Bob want a protocol Π such that for any distribution μ that has support only on 0-inputs or only on 1-inputs, $\mathrm{QIC}(\Pi, \mu)$ is small. In other words, if we use $\mathrm{QIC}\,0(\Pi)$ to denote the quantum information cost of Π against 0-distributions and $\mathrm{QIC}\,1(\Pi)$ to denote the quantum information cost of Π against 1-distributions, then we define $\mathrm{QICZ}(\Pi) = \max\{\mathrm{QIC}\,0(\Pi), \mathrm{QIC}\,1(\Pi)\}$.

When $\mathrm{QICZ}(\Pi)$ is near zero, it means that Alice and Bob learn nothing about each others' inputs when conditioned on the output of the function. The two-party secure computation question then becomes: does such a secure protocol Π exists for computing any fixed communication function G?

Intuitively, we believe that the answer should be no, at least for some communication functions G. This would align with the known impossibility of various types of secure 2-party quantum computation, though none of those impossibility results seem to apply to our setting. Interestingly, we have the following lemma, which follows directly form the way we define $\mathrm{QICZ}(G)$.

▶ **Lemma 6.** *Suppose that our version of secure 2-party quantum computation is impossible for a communication function G which contains both AND and OR as sub-functions. Then $\mathrm{QICZ}(G) > 0$, and hence $\mathrm{Adv}_1(\cdot)$ lifts to a quantum communication lower bound with the gadget G.*

We hope that future work can extend this lemma to a lifting theorem for the positive-weight quantum adversary $\mathrm{Adv}(\cdot)$; if so, the problem of lifting the positive quantum adversary bound will reduce to the problem of ruling out secure 2-party quantum computation in the model we outlined above.

New query relations

Finally, our study of the classical adversary bound led to some new relations in query complexity that are likely to be of independent interest.

▶ **Theorem 7.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{Adv}(f) = O(\widetilde{\deg}(f)^2).$$

Here $\widetilde{\deg}(f)$ is the approximate degree of f to bounded error.[7] This relationship is interesting, as it says that the positive-weight adversary method can never beat the polynomial method by more than a quadratic factor. Conceivably, this can even be used as a lower bound technique for the approximate degree of Boolean functions (which is a measure that is often of interest even apart from quantum lower bounds). In fact, we prove a strengthening of Theorem 7.

▶ **Theorem 8.** *For all (possibly partial) Boolean functions f,*

$$\widetilde{\deg}_\epsilon(f) \geq \frac{\sqrt{(1-2\epsilon)\,\mathrm{CAdv}(f)}}{\pi}.$$

This version of the theorem is stronger, since $\mathrm{Adv}(f) \leq \mathrm{CAdv}(f)$. Finally, we prove a quadratic relationship between the classical and quantum (positive-weight) adversary bounds.

▶ **Theorem 9.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{Adv}(f) \leq \mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2.$$

We note that all of these new relations hold even for partial functions. This is unusual in query complexity, where most relations hold only for total functions, and where most pairs of measures can be exponentially separated in the partial function setting.

1.4 Our Techniques

We introduce several new techniques that we believe will be useful in future work on adversary methods in communication complexity.

A lifting framework for adversary methods

One clear insight we contribute in this work is that lifting theorems for adversary method can be fruitfully attacked in a "primal" way, and using information cost. To clarify, our approach is to take a protocol Π for the lifted function $f \circ G$, and to convert it into a solution to the primal (i.e. minimization) program for the target adversary bound of f.

The primal program for an adversary method generally demands a non-negative weight $q(z,i)$ for each input string $z \in \{0,1\}^n$ and each index $i \in [n]$, such that a certain feasibility constraint is satisfied for each pair (z,w) with $f(z) \neq f(w)$, and such that $\sum_{i\in[n]} q(z,i)$ is small for each input z. Our approach is to use an information cost measure to define $q(z,i)$, where the information is measured against a distribution μ_z over n-tuples of inputs to G that evaluate to z, and where we only measure the information transmitted by the protocol about the i-th input to G, conditioned on the previous bits.

We show that this way of getting a solution to the (minimization version of) the adversary bound for f using a communication protocol for $f \circ G$ suffices for lifting CAdv to a randomized communication lower bound (with a constant-sized gadget), and that it also suffices for getting some quantum lifting theorems. Our information cost approach is similar to the approach taken in [7] to lower bound the information complexity of the AND function against the uniform distribution over 0-inputs.

[7] This is the minimum degree of an n-variate real polynomial p such that $|p(x)| \in [0,1]$ for all $x \in \{0,1\}^n$ and such that $|p(x) - f(x)| \leq 1/3$ for all x in the domain of f.

Product-to-sum reduction

One of the main tools we use in the proof of the lifting theorem for Adv_1 is what we call a product-to-sum reduction for quantum information cost. We show that if there is a protocol Π which computes some communication function F such that the geometric mean $\sqrt{\mathrm{QIC}(\Pi,\mu_0)\cdot\mathrm{QIC}(\Pi,\mu_1)}$ is small (where μ_0 and μ_1 are distributions over 0- and 1-inputs to F), then there is also a protocol Π' which also computes F and for which the arithmetic mean $\frac{1}{2}(\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1))$ is small. In particular, a lower bound for the latter measure implies a lower bound for the former. This is useful because the sum (or maximum) of the two quantum information costs is a natural operation on quantum information measures to which lower bound tools may apply, while the product is not; yet the product of these information measures arises naturally in the study of adversary methods for a lifted query function.

To prove our product-to-sum reduction, we employ a chain of reductions. First, we show that if one of $\mathrm{QIC}(\Pi,\mu_0)$ or $\mathrm{QIC}(\Pi,\mu_1)$ is much smaller than the other, then we can use Π to get a low-information protocol for $\mathrm{OR}\circ F$, the composition of the OR function with F. Next, we use an argument motivated by [11]: we use Belov's algorithm for the combinatorial group testing problem [10] to use the low-information cost protocol for $\mathrm{OR}\circ F$ to get a low-information cost protocol for the task of computing n copies of F. Finally, we use an argument from [41] to get a low-information cost protocol for F itself.

Connection to secure two-party computation

Another insight important for this work is that lifting theorems for quantum adversary methods are related to quantum secure two-party computation, a cryptographic primitive. This connection comes through the measure $\mathrm{QICZ}(G)$: for communication gadgets G for which $\mathrm{QICZ}(G)>0$, we know that secure two-party computation of G is impossible (in an "honest-but-curious" setting, where we require information-theoretic security); yet for such G, we can then lift $\mathrm{Adv}_1(f)$ to a lower bound on $\mathrm{QCC}(f\circ G)$. We believe this result is likely to extend to lifting theorems for other adversary methods in the future, though the dependence on $\mathrm{QICZ}(G)>0$ may still remain.

We provide a minimax theorem for $\mathrm{QICZ}(G)$, giving an alternate characterization of the measure. This minimax theorem is used in our lifting theorem, and may also be useful for a future lower bound on $\mathrm{QICZ}(G)$ for some communication function G, which we view as an interesting open problem.

Insights into query complexity

Our results for query complexity follow from the following insights. First, we show that for partial functions, $\mathrm{CAdv}(f)$ is equivalent to the measure $\mathrm{cfbs}(f)$ (a fractional version of critical block sensitivity [27]) by converting the primal versions of the two programs to each other; this is not difficult to do, and the main contribution comes from (1) using the correct definition of $\mathrm{CAdv}(f)$ (out of the several definitions in [6], which are not equivalent to each other for partial functions), and (2) using the correct definition of $\mathrm{cfbs}(f)$ (which is a new definition introduced in this work). We attribute one direction of this conversion to Krišjānis Prūsis (personal communication).

Second, we show that the positive-weight adversary method $\mathrm{Adv}(f)$ is smaller than, but quadratically related to, $\mathrm{CAdv}(f)$. Once again, this result is not difficult, but relies on using the correct definition of $\mathrm{CAdv}(f)$ and on using the primal versions (i.e. minimization versions) of both programs. (Indeed, we use only the primal form of all the adversary methods throughout this paper; one of our insights is that this primal form is more convenient for proving structural properties of the adversary methods, including lifting theorems.)

Finally, we show that $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{cfbs}(f)})$, and hence $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{Adv}(f)})$, and this holds even for partial functions. We do this by essentially reducing it to the task of showing $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{fbs}(f)})$. The latter is already known [28]; however, it was only known for total functions, whereas we need it to hold for partial functions as well. The problem is that the previous proof relied on recursively composing f with itself, an operation which turns the fractional block sensitivity $\mathrm{fbs}(f)$ into the block sensitivity $\mathrm{bs}(f)$; unfortunately, this trick works only for total Boolean functions. Instead, we use a different trick for turning $\mathrm{fbs}(f)$ into $\mathrm{bs}(f)$: we compose f with the promise-OR function, and show that the block sensitivity of $f \circ \textsc{PrOR}$ is proportional to the fractional block sensitivity of f. We then convert an arbitrary polynomial approximating f into a polynomial approximating $f \circ \textsc{PrOR}$ by composing it with a Chebyshev-like polynomial computing $\textsc{PrOR}$; finally, we appeal to the known result that the square root of block sensitivity lower bounds approximate degree to finish the proof.

2 Preliminaries

2.1 Distance & information measures

We define all the distance and information measures for quantum states. The classical versions can be obtained by making the corresponding registers classical.

The ℓ_1 distance between two quantum states ρ and σ is defined as

$$\|\rho - \sigma\|_1 = \mathrm{Tr}\sqrt{(\rho - \sigma)^\dagger(\rho - \sigma)}.$$

The entropy of a quantum state ρ_A on register A is defined as

$$H(A)_\rho = -\mathrm{Tr}(\rho \log \rho).$$

For a state ρ_{AB} on registers AB, the conditional entropy of A given B is

$$H(A|B)_\rho = H(AB)_\rho - H(B)_\rho.$$

Conditional entropy satisfies the following continuity bound [4]: if ρ and σ on registers AB satisfy $\|\rho - \sigma\|_1 \le \epsilon$, then

$$|H(A|B)_\rho - H(A|B)_\sigma| \le 4\epsilon \log|A| + 2h(\epsilon)$$

where $h(.)$ is the binary entropy function. For ρ_{ABC}, we define the mutual information and conditional mutual information as

$$I(A:B)_\rho = H(A)_\rho - H(A|B)_\rho \qquad I(A:B|C) = H(A|C)_\rho - H(A|BC)_\rho.$$

Mutual information satisfies

$$0 \le I(A:B|C)_\rho \le \min\{\log|A|, \log|B|\}$$

and the chain rule

$$I(A:BC)_\rho = I(A:B)_\rho + I(A:C|B)_\rho.$$

2.2 Query complexity

In query complexity, the primary object of study are Boolean functions, which are functions $f\colon \{0,1\}^n \to \{0,1\}$ where n is a positive integer. Often, we will actually study partial Boolean functions, which are defined on only a subset of $\{0,1\}^n$. We will use $\mathrm{Dom}(f)$ to denote the domain of f; this is a subset of $\{0,1\}^n$.

For a (possibly partial) Boolean function f, we use $\mathrm{D}(f)$, $\mathrm{R}(f)$, and $\mathrm{Q}(f)$ to denote its deterministic query complexity, randomized query complexity (to bounded error), and quantum query complexity (to bounded error), respectively. For the definition of these measures, see [15], though we won't use these definitions in this work.

2.2.1 Block sensitivity and its variants

We will use the following definitions.

Block notation. For a Boolean string $x \in \{0,1\}^n$ and a set $B \subseteq [n]$, we let x^B denote the string with the bits in B flipped; that is, $x^B_i = x_i$ for all $i \notin B$ and $x^B_i = 1 - x_i$ for all $i \in B$. The set B is called a *block*.

Sensitive block. For a (possibly partial) Boolean function f on n bits and an input $x \in \mathrm{Dom}(f)$, we say that a set $B \subseteq [n]$ is a *sensitive block* for x (with respect to f) if $x^B \in \mathrm{Dom}(f)$ and $f(x^B) \neq f(x)$.

Block sensitivity. The *block sensitivity* of a string $x \in \{0,1\}^n$ with respect to a (possibly partial) Boolean function f satisfying $x \in \mathrm{Dom}(f)$ is the maximum integer k such that there are k blocks $B_1, B_2, \ldots, B_k \subseteq [n]$ which are all sensitive for x and which are all disjoint. This is denoted $\mathrm{bs}(x,f)$.

Block sensitivity of a function. The *block sensitivity* of a (possibly partial) Boolean function f is the maximum value of $\mathrm{bs}(x,f)$ over $x \in \mathrm{Dom}(f)$. This is denoted $\mathrm{bs}(f)$. Block sensitivity was originally introduced by Nisan [34], and is discussed in the survey by Buhrman and de Wolf [15].

Fractional block sensitivity. The *fractional block sensitivity* of a string $x \in \{0,1\}^n$ with respect to a (possibly partial) Boolean function f satisfying $x \in \mathrm{Dom}(f)$ is the maximum possible sum of weights $\sum_B w_B$, where the weights $w_B \geq 0$ are assigned to each sensitive block of x and must satisfy $\sum_{B:i\in B} w_B \leq 1$ for all $i \in [n]$. This is denoted by $\mathrm{fbs}(x,f)$. The fractional block sensitivity of a function f, denoted $\mathrm{fbs}(f)$, is the maximum value of $\mathrm{fbs}(x,f)$ over $x \in \mathrm{Dom}(f)$. Fractional block sensitivity was defined by [2], but see also [28].

Critical block sensitivity. For a (possibly partial) Boolean function f, we say that a total Boolean function f' is a *completion* of f if $f'(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$. The *critical block sensitivity* of f, denoted $\mathrm{cbs}(f)$, is defined as

$$\min_{f'} \max_{x \in \mathrm{Dom}(f)} \mathrm{bs}(x, f'),$$

where the minimum is taken over completions f' of f. This measure was defined by [27]. It equals $\mathrm{bs}(f)$ for total functions, but may be larger for partial functions.

Critical fractional block sensitivity. For a (possibly partial) Boolean function f, we define its *critical fractional block sensitivity*, denoted $\mathrm{cfbs}(f)$, as

$$\min_{f'} \max_{x\in \mathrm{Dom}(f)} \mathrm{cfbs}(x, f'),$$

where the minimum is taken over completions f' of f. This measure has not previously appeared in the literature.

2.2.2 Adversary bounds

Positive adversary bound. For a (possibly partial) Boolean function f, we define the positive-weight adversary bound, denoted $\mathrm{Adv}(f)$, as the minimum of the following program. We will have one non-negative weight $q(x, i)$ for each $x \in \mathrm{Dom}(f)$ and each $i \in [n]$. We call such a weight scheme feasible if, for all $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we have

$$\sum_{i:x_i\neq y_i} \sqrt{q(x,i)q(y,i)} \geq 1.$$

Then $\mathrm{Adv}(f)$ is defined as the minimum of $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x, i)$ over feasible weight schemes $q(\cdot, \cdot)$. A different version of the positive-weight adversary bound was defined in [5], though the version we've currently defined appears in [30] and [39] (in the latter, our definition is equivalent to $\mathrm{MM}(f)$).

Classical adversary bound. For a (possibly partial) Boolean function f, we define the classical adversary bound, denoted $\mathrm{CAdv}(f)$, as the minimum of the following program. We will have one non-negative weight $q(x, i)$ for each $x \in \mathrm{Dom}(f)$ and each $i \in [n]$, as before. We call such a weight scheme feasible if, for all $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we have

$$\sum_{i:x_i\neq y_i} \min\{q(x,i), q(y,i)\} \geq 1.$$

Then $\mathrm{CAdv}(f)$ is defined as the minimum of $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x, i)$ over feasible weight schemes $q(\cdot, \cdot)$. Observe that this definition is the same as that of $\mathrm{Adv}(f)$, except that the feasibility constraint sums up the minimum of $q(x, i)$ and $q(y, i)$ instead of the geometric mean. This feasibility constraint is harder to satisfy, and hence we have $\mathrm{CAdv}(f) \geq \mathrm{Adv}(f)$. A different version of the classical adversary was defined in [1], though the version we've currently defined appears in [30] and [6] (in the latter, our definition is equivalent to $\mathrm{CMM}(f)$).

Singleton adversary bound. [3] introduced a simplified version of the quantum adversary bound, which we denote $\mathrm{Adv}_1(f)$. As in the other adversaries, this will be the minimum over a program that has one non-negative weight $q(x, i)$ for each pair of input $x \in \mathrm{Dom}(f)$ and index $i \in [n]$. The objective value will once again be $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x, i)$. The only difference is the constraints: instead of placing a constraint for each $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we only place this constraint for such x, y that have Hamming distance exactly 1. Observe that this is a relaxation of the constraint in the definition of $\mathrm{Adv}(f)$, and hence $\mathrm{Adv}_1(f) \leq \mathrm{Adv}(f)$ for all (possibly partial) Boolean functions f.

2.3 A generalization to relations

So far, we've defined our query measures for partial Boolean functions. However, in many cases we will be interested in studying *relations*, which are a generalization of partial Boolean functions.

In query complexity, a *relation* is a subset of $\{0,1\}^n \times \Sigma$, where Σ is some finite output alphabet. We will equate a relation $f \subseteq \{0,1\}^n \times \Sigma$ with a function that maps $\{0,1\}^n$ to subsets of Σ, so that for a string $x \in \{0,1\}^n$, the notation $f(x)$ denotes $\{\sigma \in \Sigma : (x,\sigma) \in f\}$. An algorithm which computes a relation f to error ϵ must have the guarantee that for inputs $x \in \{0,1\}^n$, the algorithm outputs a symbol in $f(x)$ with probability at least $1-\epsilon$.

Relations are generalizations of partial functions. This is because we can represent a partial function f with domain $\mathrm{Dom}(f) \subseteq \{0,1\}^n$ by a relation f' such that $f'(x) = \{f(x)\}$ for $x \in \mathrm{Dom}(f)$ and $f'(x) = \{0,1\}$ for $x \notin \mathrm{Dom}(f)$. In other words, the relational version f' of the partial function f will accept all input strings (it will be a total function), but it will consider every output symbol to be valid when given an input not in $\mathrm{Dom}(f)$. This essentially makes the inputs not in $\mathrm{Dom}(f)$ become trivial, and hence makes the relation f' intuitively equivalent to the partial function f.

We will generalize several of our query measures to relations.

Critical (fractional) block sensitivity. The original definition of $\mathrm{cbs}(f)$ from [27] actually defined it for relations. We say that a total function $f' \colon \{0,1\}^n \to \Sigma$ is a *completion* of a relation $f \subseteq \{0,1\}^n \times \Sigma$ if $(x, f'(x)) \in f$ for all $x \in \{0,1\}^n$. In other words, f' is a completion if it gives a fixed, valid output choice for each input to f. Next, we say an input $x \in \{0,1\}^n$ is *critical* if it has a unique valid output symbol in f; that is, if $|f(x)| = 1$. We let $\mathrm{crit}(f)$ denote the set of all critical inputs to f. (Note that if f is the relational version of a partial function, then $\mathrm{crit}(f)$ is equal to the domain of the partial function.) We then define

$$\mathrm{cbs}(f) \coloneqq \min_{f'} \max_{x \in \mathrm{crit}(f)} \mathrm{bs}(x, f')$$

$$\mathrm{cfbs}(f) \coloneqq \min_{f'} \max_{x \in \mathrm{crit}(f)} \mathrm{fbs}(x, f'),$$

where the minimizations are over completions f' of f. Observe that if f is the relational version of a partial function, these definitions match the previous ones.

Adversary bounds. The adversary bounds easily generalize to relations: both the positive adversary bound and the classical adversary bound will still be minimizations over weight schemes $q(x,i)$, with a non-negative weight assigned to each pair of input in $\{0,1\}^n$ and $i \in [n]$. The objective value to be minimized is the same as before: $\max_{x \in \{0,1\}^n} \sum_{i \in [n]} q(x,i)$. As for the constraints, we previously had one constraint for each pair of inputs x, y with $f(x) \neq f(y)$. For relations, we will replace this condition with the condition $f(x) \cap f(y) = \varnothing$ (that is, x and y have disjoint allowed-output-symbol sets). Hence the new constraint for $\mathrm{Adv}(f)$ becomes that for all pairs $x, y \in \{0,1\}^n$ with $f(x) \cap f(y) = \varnothing$, we have

$$\sum_{i : x_i \neq y_i} \sqrt{q(x,i) q(y,i)} \geq 1.$$

Similarly, the constraint for $\mathrm{CAdv}(f)$ is that for all pairs $x, y \in \{0,1\}^n$ with $f(x) \cap f(y) = \varnothing$, we have

$$\sum_{i : x_i \neq y_i} \min\{q(x,i), q(y,i)\} \geq 1,$$

and the constraint for $\mathrm{Adv}_1(f)$ is similar.

2.3.1 Degree measures

Degree of a function. For a (possibly partial) Boolean function f, we define its *degree* to be the minimum degree of a real polynomial p which satisfies $p(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$ as well as $p(x) \in [0, 1]$ for all $x \in \{0, 1\}^n$. We denote this by $\deg(f)$.

Approximate degree. For a (possibly partial) Boolean function f, we define its *approximate degree* to error ϵ to be the minimum degree of a real polynomial p which satisfies $|p(x) - f(x)| \le \epsilon$ for all $x \in \mathrm{Dom}(f)$ as well as $p(x) \in [0, 1]$ for all $x \in \{0, 1\}^n$. We denote this by $\widetilde{\deg}_\epsilon(f)$. When $\epsilon = 1/3$, we omit it and write $\widetilde{\deg}(f)$.

These measures are both defined and discussed in the survey by Buhrman and de Wolf [15]. We note that for partial functions, some authors do not include the requirement that the polynomial approximating the function is bounded outside of the promise set. Without this requirement, one gets a smaller measure. In this work we will only use degree and approximate degree to refer to the bounded versions of these measures.

We also note that approximate degree can be *amplified*: if a polynomial p approximates a function f to error ϵ, then we can modify p to get a polynomial q which approximates f to error $\epsilon' < \epsilon$ and which has degree that is at most a constant factor larger than the degree of p (this constant factor will depend on ϵ and ϵ').

2.3.2 Known relationships between measures

See Figure 1 for a summary of relationships between these measures (for partial functions).

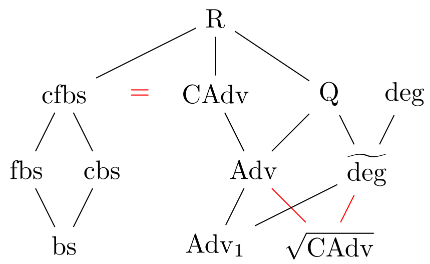

Figure 1 Relations between query complexity measures used in this work, applicable to partial functions. An upwards line from $\mathrm{M}_1(f)$ to $\mathrm{M}_2(f)$ means that $\mathrm{M}_1(f) = O(\mathrm{M}_2(f))$ for all (possibly partial) Boolean functions f. Red indicates new relationships proved in this work. We warn that some of these relationships are false for relations; in particular, CAdv may be strictly larger than cfbs and its square root may be incomparable to $\widetilde{\deg}$ for relations.

It is not hard to see that $\mathrm{bs}(f)$ is the smallest of the block sensitivity measures, and $\mathrm{cfbs}(f)$ is the largest. We know [2, 27] that $\mathrm{fbs}(f)$ and $\mathrm{cbs}(f)$ both lower bound $\mathrm{R}(f)$ for all (possibly partial) Boolean functions f; in Section 6, we show that $\mathrm{cfbs}(f)$ is also a lower bound.

We know [30, 39, 6] that $\mathrm{Q}(f) = \Omega(\mathrm{Adv}(f))$ and $\mathrm{R}(f) = \Omega(\mathrm{CAdv}(f))$. Although this it not ordinarily stated for relations, both lower bounds hold when f is a relation as well. In Section 6, we show that $\mathrm{CAdv}(f) = \Theta(\mathrm{cfbs}(f))$ for all partial functions f, and we also show that $\mathrm{CAdv}(f) = O(\mathrm{Adv}(f)^2)$ which holds for both partial functions and relations.

Approximate degree lower bounds quantum query complexity: $\mathrm{Q}(f) = \Omega(\widetilde{\deg}(f))$. It is known [9] that approximate degree is lower-bounded by $\sqrt{\mathrm{bs}(f)}$. Tal [40] showed that for total functions, $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{fbs}(f)})$. In Section 6, we extend this result to partial functions, and also prove that $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{cfbs}(f)})$.

In conclusion, $\mathrm{CAdv}(f)$ turns out to be the same as $\mathrm{cfbs}(f)$ for partial functions, and its square root lower bounds both $\mathrm{Adv}(f)$ and $\widetilde{\deg}(f)$, both of which are lower bounds on $\mathrm{Q}(f)$. Without taking square roots, $\mathrm{CAdv}(f)$ is a lower bound on $\mathrm{R}(f)$ but not on $\mathrm{Q}(f)$.

When we move from partial functions to relations, the measure $\mathrm{CAdv}(f)$ appears to get stronger in comparison to the other measures: it is strictly larger than $\mathrm{cfbs}(f)$, and appears to be incomparable to $\widetilde{\deg}(f)$ (though defining the latter for relations is a bit tricky, and we don't do so in this work).

2.4 Communication complexity

In the communication model, two parties, Alice and Bob, are given inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively, and in the most general case the task is to jointly compute a relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ by communicating with each other. In other words, on input (x, y), Alice and Bob must output a symbol $z \in \mathcal{Z}$ such that $(x, y, z) \in f$. Without loss of generality, we can assume Alice sends the first message, and Bob produces the output of the protocol.

In the classical randomized model, Alice and Bob are allowed to use shared randomness R (and also possibly private randomness R_A and R_B) in order to achieve this. The cost of a communication protocol Π, denoted by $\mathrm{CC}(\Pi)$ is the number of bits exchanged between Alice and Bob. The randomized communication complexity of a relation f with error ϵ, denoted by $\mathrm{R}_\epsilon^{\mathrm{CC}}(f)$, is defined as the minimum $\mathrm{CC}(\Pi)$ of a randomized protocol Π that computes f with error at most ϵ on every input.

Classical information complexity. The information complexity of a protocol with inputs X, Y according to μ, shared randomness R and transcript Π is given by

$$\mathrm{IC}(\Pi, \mu) = I(X : \Pi | YR)_\mu + I(Y : \Pi | XR)_\mu.$$

For any μ we have, $\mathrm{IC}(\Pi, \mu) \leq \mathrm{CC}(\Pi)$.

Quantum communication complexity. In a quantum protocol Π, Alice and Bob initially share an entangled state on registers $A_0 B_0$, and they get inputs x and y from a distribution μ. The global state at the beginning of the protocol is

$$|\Psi^0\rangle = \sum_{x,y} \sqrt{\mu(x,y)}\, |xxyy\rangle_{X\widetilde{X}Y\widetilde{Y}} \otimes |\Theta^0\rangle_{A_0 B_0}$$

where the registers $\widetilde{X}$ and $\widetilde{Y}$ purify X and Y and are inaccessible to either party. In the t-th round of the protocol, if t is odd, Alice applies a unitary $U_t : A'_{t-1} C_{t-1} \to A'_t C_t$, on her input, her memory register A'_{t-1} and the message C_{t-1} from Bob in the previous round (where $A'_0 = X A_0$ and C_0 is empty), to generate the new message C_t, which she sends to Bob, and new memory register A'_t. Similarly, if t is even, then Bob applies the unitary $U_t : B'_{t-1} C_{t-1} \to B'_t C_t$ and sends C_t to Alice. It is easy to see that $B'_t = B'_{t-1}$ for odd t, and $A'_t = A'_{t-1}$ for even t. We can assume that the protocol is safe, i.e., for all t, $A'_t = X A_t$ and $B'_t = Y B_t$, and U_t uses X or Y only as control registers. The global state at the t-th round is then

$$|\Psi^t\rangle = \sum_{x,y} \sqrt{\mu(x,y)}\, |xxyy\rangle_{X\widetilde{X}Y\widetilde{Y}} \otimes |\Theta^t\rangle_{A_t B_t C_t | xy}.$$

[31] (Proposition 9) showed that making a protocol safe does not decrease its QIC (defined below), so we shall often work with protocols of this form.

The quantum communication cost of a protocol Π, denoted by $\mathrm{QCC}(\Pi)$, is the total number of qubits exchanged between Alice and Bob in the protocol, i.e., $\sum_t \log |C_t|$. The quantum communication complexity of f with error ϵ, denoted by $\mathrm{Q}^{\mathrm{CC}}(f)$, is defined as the minimum $\mathrm{QCC}(\Pi)$ of a quantum protocol Π that computes f with error at most ϵ on every input.

Quantum information complexity. Given a quantum protocol Π as described above with classical inputs distributed as μ, its quantum information complexity is defined as

$$\mathrm{QIC}(\Pi, \mu) = \sum_{t \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_t | Y B'_t)_{\Psi^t} + \sum_{t \text{ even}} I(\widetilde{X}\widetilde{Y} : C_t | X A'_t)_{\Psi^t}.$$

The Holevo quantum information complexity is defined as

$$\begin{aligned}\mathrm{HQIC}(\Pi, \mu) &= \sum_{t \text{ odd}} I(X : B'_t C_t | Y)_{\Psi^t} + \sum_{t \text{ even}} I(Y : A'_t C_t | X)_{\Psi^t} \\ &= \sum_{t \text{ odd}} I(X : B_t C_t | Y)_{\Psi^t} + \sum_{t \text{ even}} I(Y : A_t C_t | X)_{\Psi^t} \quad \text{(for safe protocols).}\end{aligned}$$

For brevity, we shall often only use the classical input distribution μ as the subscript, or drop the subscript entirely, for these information quantities.

It was proved in [31], that for an r-round protocol Π^r, HQIC and QIC satisfy the following relation:

$$\mathrm{QIC}(\Pi^r, \mu) \geq \frac{1}{r} \mathrm{HQIC}(\Pi^r, \mu) \geq \frac{1}{2r} \mathrm{QIC}(\Pi^r, \mu).$$

Moreover, for any μ, $\mathrm{QIC}(\Pi, \mu) \leq \mathrm{QCC}(\Pi)$.

3 Lifting the classical adversary

3.1 The gadget and its properties

The gadget we use is the same one used in [24], called VER in that work. This is the function $\mathrm{VER}\colon \{0,1\}^2 \times \{0,1\}^2 \to \{0,1\}$ defined by $G(x, y) = 1$ if and only if $x + y$ is equivalent to 2 or 3 modulo 4, where $x, y \in \{0,1\}^2$ are interpreted as binary representations of integers in $\{0, 1, 2, 3\}$. This gadget has the property of being *versatile*, which means that it satisfies the following three properties:

1. Flippability: given any input (x, y), Alice and Bob can perform a local operation on their respective inputs (without communicating) to get (x', y') such that $G(x', y') = 1 - G(x, y)$.
2. Random self-reducibility: given any input (x, y), Alice and Bob can use shared randomness (without communicating) to generate (x', y') which is uniformly distributed over $G^{-1}(G(x, y))$. That is, Alice and Bob can convert any 0-input into a random 0-input and any 1-input into a random 1-input, without any communication. More formally, if the domain of G is $\mathcal{X} \times \mathcal{Y}$, we require a probability distribution ν_G over pairs of permutations in $S_{\mathcal{X}} \times S_{\mathcal{Y}}$ such that for each $(x, y) \in \mathcal{X} \times \mathcal{Y}$, sampling (σ_A, σ_B) from ν_G and constructing the pair $(\sigma_A(x), \sigma_B(y))$ gives the uniform distribution over $G^{-1}(G(x, y))$.
3. Non-triviality: the function G contains AND_2 as a sub-function (and by flippability, it also contains OR_2 as a sub-function).

These three properties were established in [24] for the function VER; a gadget which satisfies them is called *versatile*. Our lifting proof will work for any versatile gadget G. We will need the following simple lemma, which allows us to generate n-tuples of inputs to G that evaluate to either a string $s \in \{0,1\}^n$ or its complement $\hat{s} \in \{0,1\}^n$. We use the notation $G^{-1}(s)$ to denote the set of all n-tuples of inputs to G that together evaluate to $s \in \{0,1\}^n$; this abuses notation slightly (we would technically need to write $(G^{\oplus n})^{-1}(s)$, where $G^{\oplus n}$ is the function we get by evaluating n independent inputs to G).

▶ **Lemma 10.** *Let $s \in \{0,1\}^m$ be a given string and G be a versatile gadget. Then there is a protocol with no communication using shared randomness between Alice and Bob, who receive inputs (a,b) in the domain of G such that*

- *If $G(a,b) = 0$, Alice and Bob produce output strings (x,y) that are uniformly distributed in $G^{-1}(s)$*
- *If $G(a,b) = 1$, Alice and Bob produce output strings (x,y) that are uniformly distributed in $G^{-1}(\hat{s}) = G^{-1}(s \oplus 1^m)$.*

Proof. Alice and Bob share independent instances of the permutations ν_G, σ_A and σ_B as randomness. Applying independent instances of ν_G, Alice and Bob can produce (x', y') that are uniformly distributed in $G^{-1}((G(a,b))^m)$: this is done by applying m independent instances of σ_A and σ_B from ν_G to a and b respectively. Now Alice and Bob know where s differs from 0^m. By applying independent instances of the local flipping operation on x' and y' at these locations, they can negate the output of G. It is clear the resultant string (x,y) is uniformly distributed in $G^{-1}(s)$ if $G(a,b) = 0$ and in $G^{-1}(\hat{s})$ if $G(a,b) = 1$. ◀

We additionally have the following lemma, which uses the non-triviality property of a versatile gadget.

▶ **Lemma 11.** *If G is a constant-sized non-trivial gadget (containing AND_2 and OR_2 as subfunctions), and μ_0 and μ_1 are uniform distributions over its 0- and 1-inputs, then any classical protocol Π for computing G with bounded error has $\mathrm{IC}(\Pi, \mu_0), \mathrm{IC}(\Pi, \mu_1) = \Omega(1)$.*

Proof. G contains the AND_2 function, and μ_0 puts uniform $\Omega(1)$ weight on the 0-inputs of the AND_2 subfunction. [8] showed that any protocol computing the AND_2 function must have $\Omega(1)$ information cost with respect to the distribution that puts $1/3$ weight on all 0-inputs of the AND_2 function. Hence any protocol for G must also have $\mathrm{IC}(\Pi, \mu_0) = \Omega(1)$. Similarly, by considering the fact that G contains the OR_2 function, we can show that $\mathrm{IC}(\Pi, \mu_1) = \Omega(1)$. ◀

Although we only need a single versatile gadget, such as VER, we will briefly remark that there is actually an infinite family of versatile gadgets, and that this family is universal (i.e. every communication function is a sub-function of some gadget in the family).

▶ **Lemma 12.** *There is a universal family of versatile gadgets.*

Proof. For ease of notation, let G denote VER. For each $n \in \mathbb{N}^+$, we define G_n to be $\textsc{Parity}_n \circ G$. Note that G_n has the signature $\{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}$. We observe that G_n is versatile for each $n \in \mathbb{N}^+$. This is because, given a single input $((x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n))$ to G_n with $x_i, y_i \in \{0,1,2,3\}$ for each $i \in [n]$, Alice and Bob can locally generate the uniform distribution over all inputs with the same G_n-value. They can do this by first negating a random subset of the positions i of even size (using the flippability property of VER), and then converting each of the n resulting inputs to G into a random input to G with the same G-value.

Suppose z is the n-bit string with $z_i = G(x_i, y_i)$. Then flipping a random even subset of the bits of z is equivalent to generating a random string w that has the same parity as z. It follows that the above procedure generates a random input to G_n that has the same G_n

value as the original input, meaning that G_n is random self-reducible. By flipping any single gadget G within G_n, we can negate G_n, so it is also flippable. Finally, since G_n contains G as a sub-function, it also contains AND as a sub-function, so G_n is versatile for each $n \in \mathbb{N}^+$.

It remains to show that $\{G_n\}_n$ is universal. We note that since G contains AND as a sub-function, and since $G_n = \textsc{Parity}_n \circ G$, the function G_n contains $\textsc{Parity}_n \circ \text{AND}$ as a sub-function. The latter is the inner product function IP_n on n bits, which is well-known to be universal. Hence G_n is also universal. ◀

3.2 The lifting theorem

▶ **Theorem 13.** *Let G be a constant-sized versatile gadget such as VER, and let $f\colon \{0,1\}^n \to \Sigma$ be a relation. Then* $\text{RCC}(f \circ G) = \Omega(\text{CAdv}(f))$.

Proof. Let Π be a randomized protocol for $f \circ G$ which uses T rounds of communication (with one bit sent each round), and successfully computes $f \circ G$ with probability at least $1 - \epsilon/2$ for each input. Consider inputs XY distributed according to $\mu_z = \mu_{z_1} \otimes \ldots \otimes \mu_{z_n}$, where each μ_{z_i} is the uniform distribution over (x_i, y_i) in $G^{-1}(z_i)$. Suppose Π uses public randomness R which is independent of the inputs XY. We introduce the dependency-breaking random variables D and U [8] in the following way: D is independent of X, Y, R and is uniformly distributed on $\{0,1\}^n$. For each $i \in [n]$, if $D_i = 0$, then $U_i = X_i$, and if $D_i = 1$, then $U_i = Y_i$. Defined this way, given $D_i U_i$, X_i and Y_i are independent under μ_z. We shall use this algorithm to give a weight scheme $q'(z,i)$:

$$q'(z,i) = I(X_i : \Pi | X_{<i} Y D U R)_{\mu_z} + I(X_i : \Pi | Y_{<i} X D U R)_{\mu_z}$$

where $X_{<i}$ denotes $X_1 \ldots X_{i-1}$, and similarly for $Y_{<i}$. Clearly q is non-negative, and we shall show that

$$\sum_{i: z_i \neq w_i} \min\{q'(z,i), q'(w,i)\} = \Omega(1)$$

for all z, w such that $f(z) \cap f(w) = \varnothing$, where the constant in the $\Omega(1)$ is universal. Using this constant to normalize $q'(z,i)$, we get $q(z,i)$ which is a valid weight scheme. Since for any fixed value of $DU = du$, $I(X : \Pi | YR)_{\mu_{zdu}}$ is an information cost,

$$\sum_{i \in [n]} q'(z,i) = I(X : \Pi | Y D U R)_{\mu_z} + I(Y : \Pi | X D U R)_{\mu_z} \leq \text{CC}(\Pi),$$

we have for any protocol Π,

$$\text{CC}(\Pi) \geq \Omega\left(\sum_{i \in [n]} q(z,i)\right) \geq \Omega\left(\min_{\{q(z,i)\}} \sum_{i \in [n]} q(z,i)\right)$$

where the minimization is over all valid weight schemes. This proves the result.

Let z and w be two inputs to f such that $f(z) \cap f(w) = \varnothing$. Suppose z and w differ on indices in the block $\mathcal{B}$. Let $\mathcal{B}^1$ be the subset of indices in $\mathcal{B}$ where $\min\{q'(z,i), q'(w,i)\}$ is achieved by $q'(z,i)$, and $\mathcal{B}^2$ be the subset where the minimum is achieved by $q'(w,i)$. For an index $i \in \mathcal{B}^1$, let $\mathcal{B}^1_i$ denote $\mathcal{B}^1 \cap [i-1]$, and $\mathcal{B}^2_i$ denote $\mathcal{B}^2 \cap [i-1]$. We also use $\mathcal{B}^{1,c}$ to denote $[n] \setminus \mathcal{B}^1$, and $\mathcal{B}^{1,c}_i$ to denote $[i-1] \setminus \mathcal{B}^1_i$. Then,

$$
\begin{aligned}
&\sum_{i:z_i\neq w_i} \min\{q'(z,i), q'(w,i)\} \\
&= \sum_{i\in\mathcal{B}^1} (I(X_i:\Pi|X_{<i}YDUR)_{\mu_z} + I(Y_i:\Pi|Y_{<i}XDUR)_{\mu_z}) \\
&\quad + \sum_{i\in\mathcal{B}^2} (I(X_i:\Pi|X_{<i}YDUR)_{\mu_w} + I(Y_i:\Pi|Y_{<i}XDUR)_{\mu_w}) \\
&\overset{(1)}{=} \frac{1}{2}\sum_{i\in\mathcal{B}^1} (I(X_i:\Pi|X_{<i}YD_{-i}U_{-i}R)_{\mu_z} + I(Y_i:\Pi|Y_{<i}XD_{-i}U_{-i}R)_{\mu_z}) \\
&\quad + \frac{1}{2}\sum_{i\in\mathcal{B}^2} (I(X_i:\Pi|X_{<i}YD_{-i}U_{-i}R)_{\mu_w} + I(Y_i:\Pi|Y_{<i}XD_{-i}U_{-i}R)_{\mu_w}) \\
&\overset{(2)}{\geq} \frac{1}{2}\sum_{i\in\mathcal{B}^1} (I(X_i:\Pi|X_{\mathcal{B}^1_i}Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_i:\Pi|Y_{\mathcal{B}^1_i}X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z}) \\
&\quad + \frac{1}{2}\sum_{i\in\mathcal{B}^2} (I(X_i:\Pi|X_{\mathcal{B}^2_i}Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_i:\Pi|Y_{\mathcal{B}^2_i}X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w}) \\
&= \frac{1}{2}(I(X_{\mathcal{B}^1}:\Pi|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_{\mathcal{B}^1}:\Pi|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_z}) \\
&\quad + \frac{1}{2}(I(X_{\mathcal{B}^2}:\Pi|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_{\mathcal{B}^2}:\Pi|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w}). \qquad (1)
\end{aligned}
$$

Above, equality (1) follows by using the definition of D_iU_i. The inequality (2) follows from the fact that given Y_i, X_i is independent of all other X_j, Y_j D_j and U_j under both the z and w distributions, hence $I(Y_i:\Pi|Y_{<i}X(DU)_{-i}R)_{\mu_z} \geq I(Y_i:\Pi|Y_{\mathcal{B}^1_i}X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}}R)_{\mu_z}$, and equivalent inequalities hold for the other terms.

Consider $v \in \{0,1\}^n$ which agrees with w on the bits in $\mathcal{B}^1$, with z on the bits in $\mathcal{B}^2$, and with both of them outside $\mathcal{B}$. Since $f(z)$ and $f(w)$ are disjoint, at least one of the following must be true:

1. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(z)] \leq \frac{1}{2}$
2. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(w)] \leq \frac{1}{2}$.

In case 1, we shall give a protocol Π' that computes G correctly with probability at least $1-\epsilon$ in the worst case, such that

$$\mathrm{IC}(\Pi', \mu_0) = O(I(X_{\mathcal{B}^1}:\Pi|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_{\mathcal{B}^1}:\Pi|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z}).$$

Similarly, in case 2, we can use Π to give a protocol Π'' for G, such that

$$\mathrm{IC}(\Pi'', \mu_1) = O(I(X_{\mathcal{B}^2}:\Pi|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_{\mathcal{B}^2}:\Pi|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w}).$$

Due to equation (1) and Lemma 11, this proves the theorem.

In fact we only show how to construct the protocol Π' in case 1; the construction of Π'' is identical. Since z is in the domain of f and Π has worst case correctness for $f \circ G$, we must have $\Pr_{(x,y)\sim\mu_z}[\Pi(x,y) \in f(z)] \geq 1-\epsilon/2$. Therefore, in case 1, Π can distinguish between samples from μ_z and μ_v on average: on getting a sample from μ_z or μ_v, we can run Π to see if it gives an output in $f(z)$ or not, and output z or v accordingly. This average distinguishing probability can be boosted by running Π multiple times.

In Π', Alice and Bob will share $RD_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R_AR_B$ as randomness, where we use R_A and R_B to denote Alice and Bob's part of the shared randomness from Lemma 10, required to generate $z_{\mathcal{B}^1}$ if $G(a,b)=0$ and $v_{\mathcal{B}^1}$ if $G(a,b)=1$. On input (a,b) to G, Alice and Bob do the following k times for $k = \frac{2}{\epsilon^2}\ln(1/\epsilon)$:

- Alice sets $x_{\mathcal{B}^1} = R_A(a)$ and Bob sets $y_{\mathcal{B}^1} = R_B(b)$.
- Alice samples $x_{\mathcal{B}^{1,c}}$ and Bob $y_{\mathcal{B}^{1,c}}$ from private randomness, so that $G^{|\mathcal{B}^{1,c}|}(x_{\mathcal{B}^{1,c}}, y_{\mathcal{B}^{1,c}}) = z_{\mathcal{B}^{1,c}}$. They can do this since given $(DU)_{\mathcal{B}^{1,c}}$, $X_{\mathcal{B}^{1,c}}$ and $Y_{\mathcal{B}^{1,c}}$ are independent under μ_z and μ_v.
- They run Π on this (x, y) and generate the corresponding output.

(There are k independent instances of the D, U, R_A, R_B variables for each run above, but we denote all of them the same way for brevity.) The final output of Π' is 1 if the number of runs which have given an output in $f(z)$ is at least $(1-\epsilon)k$, and 0 otherwise.

Clearly if $G(a, b) = 0$, then (x, y) generated this way is uniformly distributed in the support of μ_z, and if $G(a, b) = 1$, then (x, y) is uniform in the support of μ_v. Calling the protocol in the i-th round of Π', Π_i, notice that the transcript of each Π_i is independent of AR_A, where A is the random variable for Alice's input, given the generated $X_{\mathcal{B}^1}$ (and this holds true even conditioned on $BR_B D_{\mathcal{B}^{1,c}} U_{\mathcal{B}^{1,c}} R$). Moreover, both $X_{\mathcal{B}^1}$ and Π_i are independent of BR_B given $Y_{\mathcal{B}^1}$ and of $Y_{\mathcal{B}^1}$ given BR_B (even conditioned on $D_{\mathcal{B}^1} U_{\mathcal{B}^1} R$). Let $\mu_{0,z}$ denote the distribution of $ABR_A R_B (DU)_{\mathcal{B}^{1,c}}$ when $G(a, b) = 0$, which induces the distribution $\mu_{z_{\mathcal{B}^1}}$ on $X_{\mathcal{B}^1} Y_{\mathcal{B}^1}$. Hence,

$$\begin{aligned}
I(A : \Pi_i | BR_A R_B (DU)_{\mathcal{B}^1} R)_{\mu_{0,z}} &\leq I(AR_A : \Pi_i | BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(1)}{\leq} I(X_{\mathcal{B}^1} : \Pi_i | BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(2)}{\leq} I(X_{\mathcal{B}^1} : \Pi_i | Y_{\mathcal{B}^1} BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(3)}{=} I(X_{\mathcal{B}^1} : \Pi_i | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(4)}{=} I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z}.
\end{aligned}$$

where inequality (1) follows from the fact that $I(U' : V|W) \leq I(U : V|W)$ if U' is independent of V given UW, (2) follows from the fact that $I(U : V|W) \leq I(U : V|WW')$ if V is independent of W' given W, and (3) follows from $I(U : V|WW') = I(U : V|W')$ if U and V are both independent of W given W'. Finally, (4) follows from the definition of Π_i and the the fact that the variables AB don't appear in the expression, so we can switch from $\mu_{0,z}$ to μ_z. Similarly,

$$I(B : \Pi_i | AR_A R_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \leq I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z},$$

which lets us conclude that

$$\mathrm{IC}(\Pi_i, \mu_0) \leq I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z}.$$

Thus $\mathrm{IC}(\Pi', \mu_0)$ is at most $k(I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z})$.

Now let us analyze the worst case error made by Π'. Since the output of Π_i on (a, b) is expected output of Π on (x, y) uniformly sampled from either μ_z or μ_v, Π_i produces an output in $f(z)$ on (a, b) such that $G(a, b) = 0$ with probability at least $1 - \epsilon/2$, and on (a, b) such that $G(a, b) = 1$ with probability at most $\frac{1}{2}$. Hence by the Hoeffding bound, the probability of $(1-\epsilon)k$ many 0 outputs in the first case is at least

$$1 - e^{-\epsilon^2 k/2} \geq 1 - \epsilon$$

and in the second case is at most

$$e^{-2(1/2-\epsilon)^2 k} \leq \epsilon.$$

Hence the probability of error on either input is at most ϵ. ◀

4 Quantum bounded-round lifting

The following result, analogous to Lemma 11 except with round dependence, holds in the quantum case.

▶ **Lemma 14.** *Let G be a constant-sized gadget which contains AND_2 and OR_2 as sub-functions, and μ_0 and μ_1 be uniform distributions over its 0- and 1-inputs. Then any r-round quantum protocol Π for computing G with bounded error has $\mathrm{QIC}(\Pi, \mu_0), \mathrm{QIC}(\Pi, \mu_1) = \tilde{\Omega}(1/r)$.*

The lemma has a similar proof to the classical case, and invokes the near-optimal lower bound for the quantum information cost of the AND_2 and OR_2 functions due to [12].

▶ **Theorem 15.** *If G is a constant-sized versatile gadget, then $\mathrm{QCC}^r(f\circ G) = \tilde{\Omega}(\mathrm{CAdv}(f)/r^2)$.*

Proof. For an r-round quantum protocol Π that computes $f \circ G$ to error at most $\epsilon/2$, we define

$$q'(z,i) = \sum_{t \text{ odd}} I(X_i : B_t C_t | X_{<i} Y D U)_{\mu_z} + \sum_{t \text{ even}} I(Y_i : A_t C_t | Y_{<i} X D U)_{\mu_z}$$

where the the distribution μ_z and correlation-breaking variables DU are as in the classical case. Clearly,

$$\begin{aligned}\frac{1}{r}\sum_{i=1}^{n} q'(z,i) &= \frac{1}{r}\sum_{t \text{ odd}} I(X : B_t C_t | Y D U)_{\mu_z} + \sum_{t \text{ even}} I(Y : A_t C_t | X D U)_{\mu_z} \\ &= \frac{1}{r}\,\mathrm{HQIC}(\Pi, \mu_z) \le \mathrm{QIC}(\Pi, \mu_z) \le \mathrm{QCC}(\Pi).\end{aligned}$$

Clearly $q'(z,i)$ is non-negative, and for all z, w such that $f(z) \cap f(w) = \varnothing$, we shall show that

$$\sum_{i: z_i \neq w_i} \min\{q'(z,i), q'(w,i)\} = \widetilde{\Omega}(1/r). \tag{2}$$

Thus, defining $q(z,i)$ as our weight scheme by normalizing $q'(z,i)$ with the r factor, we get the required result.

Showing (2) proceeds very similar to the classical case. For two inputs z, w to f such that $f(z) \cap f(w) = \varnothing$, which differ on the bits in block $\mathcal{B}$, let $\mathcal{B}^1 \subseteq \mathcal{B}$ be the indices where $\min\{q'(z,i), q'(w,i)\}$ is achieved by $q'(z,i)$, and $\mathcal{B}^2 \subseteq \mathcal{B}$ be the indices where it is achieved by $q'(w,i)$. By the same chain of inequalities as in the classical case, we have

$$\begin{aligned}&\sum_{i: z_i \neq w_i} \min\{q'(z,i), q'(w,i)\} \\ &\ge \frac{1}{2}\left(\sum_{t \text{ odd}} I(X_{\mathcal{B}^1} : B_t C_t | Y_{\mathcal{B}^1} D_{\mathcal{B}^{1,c}} U_{\mathcal{B}^{1,c}})_{\mu_z} + \sum_{t \text{ even}} I(Y_{\mathcal{B}^1} : A_t C_t | X_{\mathcal{B}^1} D_{\mathcal{B}^{1,c}} U_{\mathcal{B}^{2,c}})_{\mu_z}\right) \\ &\quad + \frac{1}{2}\left(\sum_{t \text{ odd}} I(X_{\mathcal{B}^2} : B_t C_t | Y_{\mathcal{B}^2} D_{\mathcal{B}^{2,c}} U_{\mathcal{B}^{2,c}})_{\mu_w} + \sum_{t \text{ even}} I(Y_{\mathcal{B}^2} : A_t C_t | X_{\mathcal{B}^2} D_{\mathcal{B}^{2,c}} U_{\mathcal{B}^{2,c}})_{\mu_w}\right).\end{aligned}$$

Note that if we had used a QIC-based definition, instead of an HQIC-based definition, for $q'(z,i)$, where we conditioned on the B_t, A_t registers, the above chain of inequalities would not have been valid, since X_i is not independent of $X_j Y_j D_j U_j$ at $j \neq i$ conditioned on B_t, and the same holds for Y_i.

Define the hybrid input v which agrees with w on the bits in $\mathcal{B}^1$, with z on the bits in $\mathcal{B}^2$ and with both outside $\mathcal{B}$. At least one of the following is true of v:

1. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y)\in f(z)]\leq\frac{1}{2}$
2. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y)\in f(w)]\leq\frac{1}{2}$.

In case 1, we shall give a protocol Π' that computes G correctly with probability at least $1-\epsilon$ in the worst case, such that

$$\begin{aligned}&\mathrm{HQIC}(\Pi',\mu_0)\\&=O\left(\sum_{t\text{ odd}}I(X_{\mathcal{B}^1}:B_tC_t|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}})_{\mu_z}+\sum_{t\text{ even}}I(Y_{\mathcal{B}^1}:A_tC_t|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}})_{\mu_z}\right).\end{aligned}$$

Similarly, in case 2, we can use Π to give a protocol Π'' for G, such that

$$\begin{aligned}&\mathrm{HQIC}(\Pi'',\mu_1)\\&=O\left(\sum_{t\text{ odd}}I(X_{\mathcal{B}^2}:B_tC_t|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w}+\sum_{t\text{ even}}I(Y_{\mathcal{B}^2}:A_tC_t|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w}\right).\end{aligned}$$

The number of rounds in Π' and Π'' will be kr, for $k=\frac{2}{\epsilon^2}\ln(1/\epsilon)$. This proves the theorem due to Lemma 14, and the fact that $\mathrm{HQIC}(\Pi',\mu)=\Omega(\mathrm{QIC}(\Pi',\mu))$ for any μ.

We only describe the protocol Π'. In Π', Alice and Bob will share the initial entangled state of Π, as well as $D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R_AR_B$ as randomness, where R_A and R_B are Alice and Bob's parts of the shared randomness from Lemma 10. Note that sharing randomness is equivalent to sharing an entangled state whose Schmidt coefficients are equal to the square roots of the corresponding probabilities, and locally measuring this state to get classical variables to use. We denote the inputs of Π' by (x',y') here to avoid confusion with the memory registers. On input (x',y'), Alice and Bob do the following k times in Π':

- Alice sets $x_{\mathcal{B}^1}=R_A(x')$ and Bob sets $y_{\mathcal{B}^1}=R_B(y')$.
- Alice samples $x_{\mathcal{B}^{1,c}}$ and Bob samples $y_{\mathcal{B}^{1,c}}$ from private randomness (this can be done unitarily), so that $G^{|\mathcal{B}^{1,c}|}(x_{\mathcal{B}^{1,c}},y_{\mathcal{B}^{1,c}})=z_{\mathcal{B}^{1,c}}$. They can do this since given $(DU)_{\mathcal{B}^{1,c}}$, $X_{\mathcal{B}^{1,c}}$ and $Y_{\mathcal{B}^{1,c}}$ are independent under μ_z and μ_v.
- They run Π on this (x,y) and generate the corresponding output.

The final output of Π' is 1 if the number of runs which have given an output in $f(z)$ is at least $(1-\epsilon)k$, and 0 otherwise.

Let $\mu_{0,z}$ denote the distribution of $X'Y'R_AR_B(DU)_{\mathcal{B}^{1,c}}$ when $G(x',y')=0$, which induces $\mu_{z_{\mathcal{B}^1}}$ on $X_{\mathcal{B}^1}Y_{\mathcal{B}^1}$. Let $C_{t,i}$ denote the message and $A_{t,i},B_{t,i}$ the memory registers of the i-th run of Π in Π', which we denote by Π_i. (There are also independent D,U,R_A,R_B variables for each run, but we drop the i dependence here.) For every i, and an odd round t, we have similar to the classical case,

$$I(X':B_{t,i}C_{t,i}|Y'R_AR_B(DU)_{\mathcal{B}^{1,c}})_{\mu_{0,z}}\leq I(X_{\mathcal{B}^1}:B_tC_t|Y_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}$$

Similarly, for even t,

$$I(Y':A_{t,i}C_{t,i}|X'R_AR_B(DU)_{\mathcal{B}^{1,c}})_{\mu_{0,z}}\leq I(Y_{\mathcal{B}^1}:A_tC_t|X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}$$

which gives us

$$\mathrm{HQIC}(\Pi_i,\mu_0)\leq\sum_{t\text{ odd}}I(X_{\mathcal{B}^1}:B_tC_t|Y_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}+\sum_{t\text{ even}}I(Y_{\mathcal{B}^1}:A_tC_t|X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}.$$

Finally, $\mathrm{HQIC}(\Pi',\mu_0)=k\,\mathrm{HQIC}(\Pi_i,\mu_0)$.

Since z is in the domain of f and Π is correct for $f\circ G$ with probability at least $1-\epsilon/2$, we have $\Pr_{(x,y)\sim\mu_z}[\Pi(x,y)\in f(z)]\geq 1-\epsilon/2$, and the probability when (x,y) is sampled according to μ_v instead is at most $\frac{1}{2}$. Therefore, by the definition of Π' and the Hoeffding bound, Π' is correct for G with probability at least $1-\epsilon$. This completes the proof. ◀

5 Towards a full quantum adversary lifting theorem

In this section, we will prove a conditional lifting theorem for a somewhat weak quantum adversary method, Adv_1. The goal of this section is primarily to introduce some tools that we believe will be helpful in eventually proving a lifting theorem for the positive-weight quantum adversary method (hopefully with a constant-sized gadget such as the VER). Specifically, we prove a product-to-sum reduction for quantum information cost in Section 5.2, which should be helpful for handling the $\sqrt{q(z,i)q(w,i)}$ terms that occur in the positive-weight adversary method; indeed, we use this product-to-sum reduction for our Adv_1 lifting theorem. We also show how lifting theorems for quantum adversary methods are related to 2-party secure communication.

We now introduce the definition of $\mathrm{QICZ}(G)$, our measure of the information leak that must happen in any purported 2-party secure computation of G.

▶ **Definition 16.** *Let $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ be a communication function. Let P be the set of all communication protocols which solve G to worst-case error $1/3$. Let Δ_0 be the set of all probability distributions over $G^{-1}(0)$, and let Δ_1 be the set of all probability distributions over $G^{-1}(1)$. We define*

$$\mathrm{QICZ}(G) \coloneqq \inf_{\Pi\in P}\ \sup_{\mu\in\Delta_0\cup\Delta_1} \mathrm{QIC}(\Pi,\mu).$$

We note that since $\mathrm{QIC}(\Pi,\cdot)$ is a continuous function of distributions [12], the inner supremum is actually attained as a maximum. We can now state our lifting theorem, as follows.

▶ **Theorem 17.** *Let $f\colon\{0,1\}^n\to\Sigma$ be a relation (where $n\in\mathbb{N}^+$ and Σ is a finite alphabet) and let $G\colon\mathcal{X}\times\mathcal{Y}\to\{0,1\}$ be a communication function which contains both AND_2 and OR_2 as subfunctions. Then*

$$\mathrm{QCC}(f\circ G)=\tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

5.1 A minimax for QICZ

Before attacking the proof of Theorem 17, we first prove a minimax theorem for the measure $\mathrm{QICZ}(G)$, giving an alternate characterization of it. To do so, we invoke Sion's minimax theorem [38].

▶ **Fact 18** (Sion's minimax). *Let V_1 and V_2 be real topological vector spaces, and let $X\subseteq V_1$ and $Y\subseteq V_2$ be convex. Let $\alpha\colon X\times Y\to\mathbb{R}$ be semicontinuous and saddle. If either X or Y is compact, then*

$$\inf_{x\in X}\sup_{y\in Y}\alpha(x,y)=\sup_{y\in Y}\inf_{x\in X}\alpha(x,y).$$

To understand the statement of this theorem, we need a few definitions:

1. A real-valued function ϕ is *convex* if $\phi(\lambda x_1+(1-\lambda)x_2)\le\lambda\phi(x_1)+(1-\lambda)\phi(x_2)$ for all $x_1,x_2\in\mathrm{Dom}(\phi)$ and all $\lambda\in(0,1)$.
2. A real-valued function ϕ is *concave* if $\phi(\lambda x_1+(1-\lambda)x_2)\ge\lambda\phi(x_1)+(1-\lambda)\phi(x_2)$ for all $x_1,x_2\in\mathrm{Dom}(\phi)$ and all $\lambda\in(0,1)$.
3. A function $\alpha\colon X\times Y\to\mathbb{R}$ is *saddle* if $\alpha(\cdot,y)$ is convex as a function of x for each fixed $y\in Y$, and if $\alpha(x,\cdot)$ is concave as a function of y for each fixed $x\in X$.

4. A real-valued function ϕ is *upper semicontinuous* at a point x if for any $\epsilon > 0$, there exists a neighborhood U of x such that for all $x' \in U$, we have $\phi(x') < \phi(x) + \epsilon$.
5. A real-valued function ϕ is *lower semicontinuous* at a point x if for any $\epsilon > 0$, there exists a neighborhood U of x such that for all $x' \in U$, we have $\phi(x') > \phi(x) - \epsilon$.
6. A function $\alpha\colon X \times Y \to \mathbb{R}$ is *semicontinuous* if $\alpha(\cdot, y)$ is lower semicontinuous over all of X for each $y \in Y$ and if $\alpha(x, \cdot)$ is upper semicontinuous over all of Y for each $x \in X$.

We now use Sion's minimax theorem to prove a minimax theorem for QICZ.

▶ **Theorem 19.** *Fix a communication function G. Let P be the set of all protocols which solve G to worst-case error $1/3$, let Δ_0 be the set of probability distributions over 0-inputs to G, and let Δ_1 be the set of probability distributions over 1-inputs to G. Then*

$$\frac{1}{2} \max_{\substack{\mu_0 \in \Delta_0 \\ \mu_1 \in \Delta_1}} \inf_{\Pi \in P} \mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1) \le \mathrm{QICZ}(G) \le \max_{\substack{\mu_0 \in \Delta_0 \\ \mu_1 \in \Delta_1}} \inf_{\Pi \in P} \mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1).$$

Moreover, the maximum is attained.

Proof. We will aim to use Sion's minimax theorem [38]. To this end, we start with a bit of formalism. The set P of protocols is, of course, an infinite set, and has somewhat complicated structure. In order to apply a minimax theorem, however, we want to switch over to a convex subset of a real topological vector space. To do so, we first consider the free real vector space over P, which we denote by $V(P)$. This is the real vector space consisting of all formal (finite) linear combinations of elements in P; the set P is a basis of this vector space. We further consider the 1-norm on this space, where we define the 1-norm of a formal (finite) linear combination as the sum of absolute values of coefficients in the linear combination. This norm induces a topology over $V(P)$, making it a real topological vector space.

Our set of algorithms will be the subset of $V(P)$ consisting of vectors with norm 1 that have non-negative coefficients in the linear combination; we denote this subset by R. It is not hard to see that the elements of R are simply all the finite-support probability distributions over protocols in P. We observe that R is a convex set. This will be the set over which we take the infimum in Sion's minimax theorem.

Observe that since the input set $\mathrm{Dom}(G)$ of G is finite, the sets Δ_0 and Δ_1 are both convex, compact subsets of the real vector space $\mathbb{R}^{|\mathrm{Dom}(G)|}$, which has a standard topology. It follows that the set $\Delta_0 \times \Delta_1$ is also convex and compact (as a subset of the real topological vector space $\mathbb{R}^{2|\mathrm{Dom}(G)|}$). This will be the set over which we take the supremum in Sion's minimax.

Let $A \in R$. This is a finite-support probability distribution over protocols in P; however, it is always possible to use shared randomness to construct a single protocol $\Pi_A \in P$ whose behavior exactly matches that of A (that is, in Π_A, Alice and Bob will sample a protocol from A using shared randomness, and then run that protocol). Finally, we define $\alpha\colon R \times (\Delta_0 \times \Delta_1) \to [0, \infty)$ by setting

$$\alpha(A, (\mu_0, \mu_1)) \coloneqq \mathrm{QIC}(\Pi_A, \mu_0) + \mathrm{QIC}(\Pi_A, \mu_1).$$

This will be the function on which we apply Sion's minimax.

It remains to show that α is semicontinuous and quasisaddle. It is not hard to see that the sum of two semicontinuous functions (on the same domain) is semicontinuous, and that the sum of two saddle functions is saddle. It will therefore be sufficient to show that QIC is semicontinuous and saddle.

In [41] (Lemma 5), it was shown that $\mathrm{QIC}(\cdot,\mu)$ is linear (and hence convex) for each μ. In [41] (Lemma 6), it was shown that $\mathrm{QIC}(\Pi,\cdot)$ is concave. Hence QIC is saddle, and therefore so is α. In [12] (Lemma 4.8), it was shown that $\mathrm{QIC}(\Pi,\cdot)$ is continuous.

It remains to show the lower semicontinuity of $\mathrm{QIC}(\cdot,\mu)$. More explicitly, for each fixed distribution μ, each $A\in R$ and each $\epsilon>0$, there exists $\delta>0$ such that for all $A'\in R$ with $\|A-A'\|_1<\delta$, we have $\mathrm{QIC}(\Pi_{A'},\mu)>\mathrm{QIC}(\Pi_A,\mu)-\epsilon$.

We can write $A=(1-p)B+pC$ and $A'=(1-p)B+pC'$ where $B,C,C'\in R$, and (C,C') is a pair of distributions with disjoint support. In other words, B is the probability distribution consisting of the (normalized) overlap between A and A', while C and C' are the probability distributions we get from subtracting out the overlap from A and from A' respectively. If $\|A-A'\|_1<\delta$, we must have $p<\delta/2$. Now, by the linearity of $\mathrm{QIC}(\cdot,\mu)$, we have $\mathrm{QIC}(\Pi_A,\mu)=(1-p)\,\mathrm{QIC}(\Pi_B,\mu)+p\,\mathrm{QIC}(\Pi_C,\mu)$ and $\mathrm{QIC}(\Pi_{A'},\mu)=(1-p)\,\mathrm{QIC}(\Pi_B,\mu)+p\,\mathrm{QIC}(\Pi_{C'},\mu)$. We want to choose δ so that $\mathrm{QIC}(\Pi_{A'},\mu)>\mathrm{QIC}(\Pi_A,\mu)-\epsilon$, or equivalently, so that $\mathrm{QIC}(\Pi_{C'},\mu)>\mathrm{QIC}(\Pi_C,\mu)-\epsilon/p$. This rearranges to wanting $\epsilon/p>\mathrm{QIC}(\Pi_C,\mu)-\mathrm{QIC}(\Pi_{C'},\mu)$; hence it is sufficient to choose δ so that $2\epsilon/\delta>\mathrm{QIC}(\Pi_C,\mu)-\mathrm{QIC}(\Pi_{C'},\mu)$. It is clear that such δ can always be chosen, as $\mathrm{QIC}(\Pi_C,\mu)$ must be finite.

We conclude that $\mathrm{QIC}(\cdot,\mu)$ is lower semicontinuous. Sion's minimax theorem (Fact 18) then gives

$$\begin{aligned}&\inf_{A\in R}\sup_{(\mu_0,\mu_1)\in\Delta_0\times\Delta_1}\mathrm{QIC}(\Pi_A,\mu_0)+\mathrm{QIC}(\Pi_A,\mu_1)\\&=\sup_{(\mu_0,\mu_1)\in\Delta_0\times\Delta_1}\inf_{A\in R}\mathrm{QIC}(\Pi_A,\mu_0)+\mathrm{QIC}(\Pi_A,\mu_1).\end{aligned}$$

Since R contains P as a subset, and since every protocol in R can be converted into an equivalent protocol in P, taking an infimum over $A\in R$ is the same as taking an infimum over $\Pi\in P$. It is then clear that the left hand side is at least $\mathrm{QIC}(G)$ (since the latter has only one $\mathrm{QIC}(\Pi,\mu_0)$ or $\mathrm{QIC}(\Pi,\mu_1)$ term instead of both), but no more than twice $\mathrm{QIC}(G)$ (since the maximum of $\mathrm{QIC}(\Pi,\mu_0)$ and $\mathrm{QIC}(\Pi,\mu_1)$ is at least the average of the two). Hence the desired result follows. The attainment of the maximum comes from the fact that an upper semicontinuous function on a nonempty compact set attains is maximum, combined with the fact that a pointwise infimum of upper semicontinuous functions is upper semicontinuous. ◀

5.2 Product-to-sum reduction for quantum information

In order to prove Theorem 17, we will need a way to bound the product of quantum information cost on the "0-input" side and the quantum information cost on the "1-input" side. We start with the following definition.

▶ **Definition 20.** *Let G be a communication function. We say a distribution μ is* nontrivial *for G if for any protocol Π computing G (to bounded error against worst-case inputs), it holds that $\mathrm{QIC}(\Pi,\mu)>1/\operatorname{poly}(r)$, where r is the number of rounds of Π. (In particular, it should not be possible to achieve $\mathrm{QIC}(\Pi,\mu)=0$ if μ is nontrivial.)*

Using this definition, we state the following theorem, which is the main result of this subsection.

▶ **Theorem 21.** *Let G be a gadget, let μ_0 and μ_1 be nontrivial 0- and 1-distributions for G, and let Π be a protocol solving G (to bounded error against worst-case inputs). Then there is a protocol Π' which also solves G (to bounded error against worst-case inputs) which satisfies*

$$\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)=O\left(\sqrt{\mathrm{QIC}(\Pi,\mu_0)\,\mathrm{QIC}(\Pi,\mu_1)}\cdot\operatorname{polylog} r\right),$$

where r is the number of rounds of Π and where the constant in the big-O is universal. Moreover, the number of rounds of Π' is polynomial in that of Π.

Before we prove this, we will need a few lemmas. In the following, we will use $G^{\oplus n}$ to denote the direct sum of n copies of G; that is, if $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$, then $G^{\oplus}\colon (\mathcal{X}^n)\times(\mathcal{Y}^n)\to\{0,1\}^n$ is the function that takes in n separate copies to G and outputs n separate outputs from G.

▶ **Lemma 22.** *Let (G,μ_0,μ_1) be any gadget, 0-distribution, and 1-distribution, let $n\in\mathbb{N}^+$, and let Π be a protocol which solves $G^{\oplus n}$ (to bounded error against worst-case inputs). Then there is a protocol Π' which solves G (to bounded error against worst-case inputs) which satisfies*

$$\frac{\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)}{2}\le\frac{1}{n}\cdot\max_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right).$$

Proof. Let for $z\in\{0,1\}^n$, let $\Pi_{i,z}$ be the protocol which: takes an input to G; artificially generates $n-1$ inputs from μ_{z_j} for $j\neq i$ for all the gadgets G except at position i; places the true input at position i; runs Π on the resulting input to $G^{\oplus n}$; traces out all the outputs except for position i; and returns the result. Note that $\Pi_{i,z}$ does not depend on the value of z_i, but depends on the rest of z. If we use z^i to denote the string x with i flipped, we have $\Pi_{i,z}=\Pi_{i,z^i}$ for all x and i.

[41] (Lemma 4) showed that for all $x\in\{0,1\}^n$,

$$\sum_{i=1}^{n}\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})=\mathrm{QIC}\left(\Pi,\mu_z\right).$$

Let $\Pi':=\frac{1}{n}\frac{1}{2^n}\sum_{i=1}^{n}\sum_{x\in\{0,1\}^n}\Pi_{i,z}$. Again by [41] (Lemma 5),

$$\begin{aligned}\frac{\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)}{2}&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_0)+\mathrm{QIC}(\Pi_{i,z},\mu_1)}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})+\mathrm{QIC}(\Pi_{i,z},\mu_{z^i_i})}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})+\mathrm{QIC}(\Pi_{i,z^i},\mu_{z^i_i})}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})\\&=\frac{1}{n2^n}\sum_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right)\\&\le\frac{1}{n}\cdot\max_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right).\end{aligned}$$ ◀

▶ **Lemma 23.** *Let $G_1,G_2,\ldots,G_n$ be any sequence of communication tasks, and for each $i\in[n]$ let Π_i be a protocol which solves G_i (to bounded error against worst-case inputs). Let F be a (possibly partial) query function on n bits, and let Q be a T-query quantum query algorithm computing F (to bounded error against worst-case inputs). Then there is a protocol Π' computing $F\circ\{G_i\}_i$ (to bounded error against worst-case inputs) such that for any $z\in\mathrm{Dom}(F)$ and any distribution μ_z supported on $(G_1\oplus G_2\oplus\cdots\oplus G_n)^{-1}(z)$, we have*

$$\mathrm{QIC}(\Pi',\mu_z)=\tilde{O}\left(T\log\log n\cdot\max_{i\in[n]}\mathrm{QIC}(\Pi_i,\mu_z^i)\right),$$

where μ_z^i is the marginal of μ_z on gadget number i.

Proof. Let $\hat{\Pi}_i$ be the amplified and purified version of Π_i, reducing its worst-case error on G to $\delta/T^{10}\log n$ and using the uncomputing trick to clean up garbage (δ will be chosen later). Then the information cost of $\hat{\Pi}_i$ against any fixed distribution increases by a factor of at most $O(\log T + \log\log n + \log 1/\delta)$ compared to Π_i. Next, Π' be the protocol where Alice runs the query algorithm for F, and whenever she needs to make a query i, she sends i to Bob and they compute gadget number i using $\hat{\Pi}_i$. Since F succeeds with bounded error on worst-case inputs and since $\hat{\Pi}_i$ has such a low probability of error, the protocol Π' correctly computes $F \circ G$ on worst-case inputs.

Fix $z \in \mathrm{Dom}(F)$ and μ_z supported on $(G^{\oplus n})^{-1}(z)$. We will expand out $\mathrm{QIC}(\Pi', \mu_z)$. In round $t \leq T$ of the query algorithm, there are two types of messages between Alice and Bob: one message from Alice to Bob containing a copy E_t of the query register D_t for step $t \leq T$, which Alice knows from her simulation of the algorithm Q for F; and all the messages between Alice and Bob implementing $\hat{\Pi}_i$. Denote those messages by $C_{t,j}$. Note that E_t also gets passed back from Bob to Alice at the end of each round for cleanup purposes.

We name the rest of the registers. Let the input registers be X and Y, and let Alice hold register D_t specifying the position to query at round t, a work register $\tilde{A}_t$ related to the implementation of the algorithm Q for f (which stays untouched for all j), and register $A_{t,j}$ related to the implementation of the $\hat{\Pi}_i$ protocols for round t. Bob holds query register E_t (passed from Alice, untouched for all j) as well as work register $B_{t,j}$ for the implementation of the $\hat{\Pi}_i$ protocols. Let R be the purification register. Then using r to denote the index of the last round of the $\hat{\Pi}_i$, we have

$$
\begin{aligned}
\mathrm{QIC}(\Pi', \mu_z) = &\sum_{t=1}^{T} I(\widetilde{X}\widetilde{Y} : E_t | Y B_{t,0})_{\Psi_z^t} + \sum_{t=1}^{T} I(\widetilde{X}\widetilde{Y} : E_t | X A_{t,r} \tilde{A}_t D_t)_{\Psi_z^t} \\
&+ \sum_{t=1}^{T} \sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j} E_t)_{\Psi_z^{t,j}} \\
&+ \sum_{t=1}^{T} \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} D_t \tilde{A}_t)_{\Psi_z^{t,j}}.
\end{aligned}
$$

For the terms $I(\widetilde{X}\widetilde{Y} : E_t | Y B_{t,0})_{\Psi_z^t}$ and $I(\widetilde{X}\widetilde{Y} : E_t | X A_{t,r} \tilde{A}_t D_t)_{\Psi_z^t}$, we note that $B_{t,0}$ and $A_{t,r}$ are the start state on Bob's side and the end state on Alice's side for $\hat{\Pi}$, and can be assumed to be independent of all other registers. Hence we shall ignore the registers $B_{t,0}$ and $A_{t,r}$ in the conditioning systems. Let $|\Phi^t\rangle$ that is obtained by replacing the $\tilde{A}_t D_t E_t$ registers of $|\Psi_z^t\rangle$ with the state of the query algorithm for f after t queries (with the query register D_t duplicated). $|\Phi^t\rangle_{\tilde{A}_t D_t E_t | zxy}$ depends on x and y only through z, which is fixed. Hence $I(\widetilde{X}\widetilde{Y} : E_t | Y)_{\Phi_z^t}$ and $I(\widetilde{X}\widetilde{Y} : E_t | X \tilde{A}_t)_{\Phi_z^t}$ are both 0. Clearly, Φ_z^t is the state the protocol would have been in if $\hat{\Pi}_i$ were run with 0 error. Since the protocol runs of $\hat{\Pi}_i$ make very small error instead, we have instead $\left\| |\Psi^t\rangle_z - |\Phi^t\rangle_z \right\|_1 \leq \epsilon$, where $\epsilon = O(\delta/\operatorname{poly}(T)\log n)$. This implies

$$
\begin{aligned}
I(\widetilde{X}\widetilde{Y} : E_t | Y)_{\Psi_z^t} &= H(E_t|Y)_{\Psi_z^t} - H(E_t|\widetilde{X}\widetilde{Y}Y)_{\Psi_z^t} \\
&\leq H(E_t|Y)_{\Phi_z^t} - H(E_t|\widetilde{X}\widetilde{Y}Y)_{\Phi_z^t} + 8\epsilon \log|E_t| + 4h(\epsilon) \\
&= 8\epsilon + 4h(\epsilon).
\end{aligned}
$$

The total sum of $I(\widetilde{X}\widetilde{Y} : E_t | Y B_{t,0})_{\Psi_z^t}$ over all t is therefore at most $\delta/2$, and the same applies to $I(\widetilde{X}\widetilde{Y} : E_t | X \tilde{A}_t)_{\Psi_z^t}$.

We then have

$$\begin{aligned}
\mathrm{QIC}(\Pi', \mu_z) &\le \delta + \sum_{t=1}^{T} \left(\sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j} E_t)_{\Psi_z^t} \right. \\
&\quad \left. + \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} D_t \tilde{A}_t)_{\Psi_z^t} \right) \\
&= \delta + \sum_{t=1}^{T} \sum_{i=1}^{n} \Pr[D_t = i] \left(\sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j})_{\Psi^t_{z, D_t = i}} \right. \\
&\quad \left. + \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} \tilde{A}_t)_{\Psi^t_{z, D_t = i}} \right) \\
&= \delta + \sum_{t=1}^{T} \sum_{i=1}^{n} \Pr[D_t = i] \, \mathrm{QIC}(\hat{\Pi}_i, \mu_z^i) \\
&\le \delta + T \max_i \mathrm{QIC}(\hat{\Pi}_i, \mu_z^i) \\
&\le \delta + O(T(\log T + \log\log n + \log 1/\delta) \max_i \mathrm{QIC}(\Pi_i, \mu_z^i)).
\end{aligned}$$

Setting $\delta = T \max_i \mathrm{QIC}(\Pi_i, \mu_z^i)$, but ensuring $\epsilon \le 1/3$ (since we can't amplify a negative amount), we get

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(T \max_i \mathrm{QIC}(\Pi_i, \mu_z^i) \log\left(2 + \frac{T^{10} \log n}{\max_i \mathrm{QIC}(\Pi_i, \mu_z^i)}\right)\right). \quad ◀$$

▶ **Lemma 24.** *Let G be a gadget, let μ_0 and μ_1 be a 0-distribution and a 1-distribution for G, let $n \in \mathbb{N}^+$, and let Π be a protocol computing $OR_n \circ G$ (to bounded error against worst-case inputs). Then there is a protocol Π' computing $G^{\oplus n}$ (to bounded error against worst-case inputs) such that*

$$\max_{z \in \{0,1\}^n} \mathrm{QIC}(\Pi', \mu_z) = \tilde{O}\left(\sqrt{n} \cdot \max_{z \in \{0,1\}^n} \mathrm{QIC}(\Pi, \mu_z) \right).$$

Proof. Consider the following task: the goal is to output a hidden string $z \in \{0,1\}^n$, and the allowed queries are subset-OR queries, meaning that for each subset $S \subseteq [n]$ there is a query which returns $\mathrm{OR}(z_S)$ (which equals 1 if $z_i = 1$ for some $i \in S$, and returns 0 otherwise). We can model this task as a query function F on a promise set $P \subseteq \{0,1\}^{2^n}$. Each string in P is a long encoding $u(z) \in \{0,1\}^{2^n}$ of some string $z \in \{0,1\}^n$, with the long encoding $u(z)$ being a string with $(u(z))_S = \mathrm{OR}(z_S)$ for all S. In other words, u is a function $u \colon \{0,1\}^n \to \{0,1\}^{2^n}$. The function F is defined by $F(u(z)) = z$ for all $z \in \{0,1\}^n$, where $\mathrm{Dom}(F) = \{u(z) : z \in \{0,1\}^n\}$. It is not hard to verify that this function is well-defined.

The function f is sometimes called the combinatorial group testing problem. We have $\mathrm{D}(F) \le n$, since we can query $u(z)_{\{i\}}$ for all $i \in [n]$ to get the bits z_i one by one and then output all of z. (Note that the input size to F is of length $N = 2^n$, so n does not represent the input size here.) Belovs [10] showed that $\mathrm{Q}(F) = O(\sqrt{n})$. This result will play a key role in our analysis here, which is motivated by [11] (where this algorithm of Belovs was similarly used to reduce direct-sum computations to OR computations).

Now, observe that $F \circ u$ is the identity function on n bit strings. The protocol Π' for $G^{\oplus}$ will be a protocol for $F \circ u \circ G$. We use Lemma 23 on the query function F and the communication tasks $u(G^{\oplus n})_1, u(G^{\oplus n})_2, \ldots, u(G^{\oplus n})_{2^n}$. The query algorithm for F makes

$T = O(\sqrt{n})$ queries. Each of the communication tasks is of the following form: take as input n copies to G, and output the OR of a fixed subset S of the copies of G. To solve this task, which we denote F_S, we describe a protocol Π_S. In this protocol, Alice and Bob will use their shared randomness to replace the inputs in positions $i \notin S$ by independent samples from μ_0. They will then run Π to compute the OR of the n copies of G.

The correctness of Π' is clear, so we analyze its information cost. Fix $z \in \{0,1\}^n$, and denote by z_S the string satisfying $(z_S)_i = z_i$ if $i \in S$ and $(z_S)_i = 0$ if $i \in S$. In order to upper bound $\mathrm{QIC}(\Pi', \mu_z)$ using Lemma 23, we let μ'_z be the distribution on strings of length $\{0,1\}^{n2^n}$ that we get by sampling a string from μ_z and making 2^n copies of it. We observe that that the behavior of Π' when acting on μ_z is exactly the composed behavior of the query algorithm for F composed with the protocols Π_S acting on the distribution μ'_z; Lemma 23 therefore gives us

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(\sqrt{n} \cdot \max_S \mathrm{QIC}(\Pi_S, \mu_z) \log\left(2 + \frac{n^5 \log N}{\max_S \mathrm{QIC}(\Pi_S, \mu_z)}\right)\right)$$

(where we used the more precise bound given in the proof of Lemma 23). Recall that Π_S replaces the samples of μ_z that correspond to bits $i \notin S$ with freshly-generated samples from μ_0, and then runs Π; hence $\mathrm{QIC}(\Pi_S, \mu_z) = \mathrm{QIC}(\Pi_S, \mu_{z_S}) \leq \mathrm{QIC}(\Pi, \mu_{z_S})$. The maximum over sets S of $\mathrm{QIC}(\Pi, \mu_{z_S})$ is clearly at most the maximum over $w \in \{0,1\}^n$ of $\mathrm{QIC}(\Pi, \mu_w)$. Using $\log N = n$, we can therefore write

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(\sqrt{n} \cdot \max_w \mathrm{QIC}(\Pi, \mu_w) \log\left(2 + \frac{n^6}{\max_w \mathrm{QIC}(\Pi, \mu_w)}\right)\right). \qquad \blacktriangleleft$$

▶ **Lemma 25.** *Let G be a gadget, let μ_0 and μ_1 be a 0-distribution and a 1-distribution for G, and let Π be a protocol computing G (to bounded error against worst-case inputs). Then for any $n \in \mathbb{N}^+$, there is a protocol Π' computing $OR_n \circ G$ (to bounded error against worst-case inputs) such that*

$$\max_{z \in \{0,1\}^n} \mathrm{QIC}\left(\Pi', \mu_z\right) = O(n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)).$$

Proof. In order to compute $\mathrm{OR}_n \circ G$, the protocol Π' will simply compute each copy of G one at a time, stopping as soon as a 1 has been found. The idea is that this will ensure the number of computations of 0-inputs to G is at most $O(n)$ while the number of computations of 1-inputs to G is $\tilde{O}(1)$.

To be more formal, we consider a cleaned up version $\hat{\Pi}$ of Π, which will have error $O(1/n)$ and which cleans up all the garbage and resets Alice and Bob's states to their initial states after the computation is complete. The protocol Π' will run $\hat{\Pi}$ on each input to G, in sequence, stopping when an output 1 has been found. To implement this, we will name the registers: suppose the protocol $\hat{\Pi}$ uses registers A and O_A on Alice's side and registers B and O_B on Bob's side, where O_A and O_B store the final output of $\hat{\Pi}$. At the beginning of $\hat{\Pi}$, the registers are expected to be $|0\rangle_A |0\rangle_B |0\rangle_{O_A} |0\rangle_{O_B}$. The guarantee of $\hat{\Pi}$ is that at the end of the algorithm, the registers will be in the state $|0\rangle_A |0\rangle_B |b\rangle_{O_A} |b\rangle_{O_B}$, where b is close to the output of G on that input. We now implement Π' by adding an additional register on each side, denoted S_A and S_B, which stores the strings of outputs of all the runs of $\hat{\Pi}$. These registers are each initialized to 0^n. At the end of run i of $\hat{\Pi}$ (which computes gadget i), Alice and Bob will each swap the register O_A with the i-th bit of S_A; this resets the registers used by $\hat{\Pi}$ to be all zero, and it stores the output of the i-th run of $\hat{\Pi}$ so that Π' has access to it. It also preserves the property that $S_A = S_B$ throughout the algorithm.

The final detail is that in Π', Alice and Bob only run $\hat{\Pi}$ on gadget i if they see that all the previous runs resulted in output 0; that is, they control the implementation of $\hat{\Pi}$ on the registers S_A and S_B being equal to 0^n. This will ensure that once a 1 is found, no further information will be exchanged between Alice and Bob. The final output of Π' will be 0 if S_A and S_B are 0^n, and it will be 1 otherwise.

The correctness of Π' (to worst-case bounded error) is clear, so we analyze its information cost against μ_z for a fixed string $z \in \{0,1\}^n$. The information cost $\mathrm{QIC}(\Pi', \mu_z)$ is a sum of information exchanged over all rounds; let $\mathrm{QIC}_i(\Pi', \mu_z)$ denote the sum of information exchanged only in the rounds corresponding to the computation of the i-th copy of G, so that $\mathrm{QIC}(\Pi', \mu_z) = \sum_{i=1}^n \mathrm{QIC}_i(\Pi', \mu_z)$.

Let T be the number of rounds used by $\hat{\Pi}$. Let $S_{A,i}$ and $S_{B,i}$ be the registers S_A and S_B during the computation of the i-th copy of G. Use X and Y to denote Alice and Bob's inputs respectively, with X_i and Y_i being the inputs to copy i of G and with $\tilde{X}$ and $\tilde{Y}$ denoting their purifications, and let C be the register passed back and forth between Alice and Bob in $\hat{\Pi}$. Then

$$\mathrm{QIC}_i(\Pi', \mu_z) = \sum_{t \le T \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_t | Y B_t S_{B,i}) + \sum_{t \le T \text{ even}} I(\widetilde{X}\widetilde{Y} : C_t | X A_t S_{A,i}).$$

We note that the register $S_{B,i}$ in the odd terms is classical, as is the register $S_{A,i}$. Hence the conditional mutual information conditioned on $S_{B,i}$ is the expectation of the conditional mutual information conditioned on the events $S_{B,i} = w$ for each string $w \in \{0,1\}^n$ (see, for example, [12], end of Section 3.1). In other words,

$$\begin{aligned} &I(\widetilde{X}\widetilde{Y} : C_t | Y B_t S_{B,i}) \\ &= \Pr[S_{B,i} = 0^n] I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=0^n} + \Pr[S_{B,i} \ne 0^n] I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i} \ne 0^n}. \end{aligned}$$

Note that by the construction of Π', in the second term we have $I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=\ne 0^n} = 0$, since the registers C_t are all 0 as Alice and Bob do not run $\hat{\Pi}$ at all when $S_{B,i} \ne 0^n$. The term $I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=0^n}$ is just $I(\widetilde{X}_i\widetilde{Y}_i : C_t | Y_i B_t)$, since the run of $\hat{\Pi}$ ignores everything outside of the input to the i-th copy of G. Hence we have

$$\begin{aligned} &\mathrm{QIC}_i(\Pi', \mu_z) \\ &= \Pr[S_{B,i} = 0^n] \sum_{t \le T \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | Y_i B_t) + \Pr[S_{A,i} = 0^n] \sum_{t \le T \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | X_i A_t) \\ &= \Pr[S_{A,i} = 0] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}). \end{aligned}$$

From this, it follows that

$$\mathrm{QIC}(\Pi', \mu_z) = \sum_{i=1}^n \Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}).$$

To upper bound this, we note that the total sum of all the terms $\Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i})$ for i such that $z_i = 0$ is at most $n \, \mathrm{QIC}(\hat{\Pi}, \mu_0)$, where we've upper bounded $\Pr[S_{A,i} = 0^n] \le 1$. For i such that $z_i = 1$, we split into two cases: in the case where i is the first index such that $z_i = 1$, we upper bound $\Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}) \le \mathrm{QIC}(\hat{\Pi}, \mu_1)$. In contrast, for all i such that $z_i = 1$ and for which there was a previous index $j < i$ with $z_j = 1$, we note that the $1/n$ error guarantee of $\hat{\Pi}$ ensures that $\Pr[S_{A,i} = 0^n] \le 1/n$; hence these terms are individually at most $(1/n) \, \mathrm{QIC}(\hat{\Pi}, \mu_1)$, and the sum of all of them is at most $\mathrm{QIC}(\hat{\Pi}, \mu_1)$. We conclude that

$$\mathrm{QIC}(\Pi', \mu_z) \le n \, \mathrm{QIC}(\hat{\Pi}, \mu_0) + 2 \, \mathrm{QIC}(\hat{\Pi}, \mu_1).$$

Finally, we note that $\hat{\Pi}$ simply repeats Π $O(\log n)$ times and takes a majority votes in order to amplify (and then runs this in reverse to clean up garbage). Hence we have

$$\begin{aligned}
&\mathrm{QIC}(\hat{\Pi}, \mu_0) = O(\log n \cdot \mathrm{QIC}(\Pi, \mu_0)),\\
&\mathrm{QIC}(\hat{\Pi}, \mu_1) = O(\log n \cdot \mathrm{QIC}(\Pi, \mu_1)).
\end{aligned}$$

This gives the upper bound on $\mathrm{QIC}(\Pi', \mu_z)$ of $O(n \log n \cdot \mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1))$.

Finally, we sketch how to shave the log factor from the $\mathrm{QIC}(\Pi, \mu_0)$ term. To do so, we avoid amplifying $\hat{\Pi}$. Instead, we simply run $\hat{\Pi}$ on each input. If the output is 1, we run $\hat{\Pi}$ again on the same copy of G. We do so until the number of 0 outputs outnumbers the number of 1 outputs. If $O(\log n)$ repetitions happened and the number of 1-outputs is still larger than the number of 0 inputs, we finally "believe" that this gadget evaluates to 1 and halt. Otherwise, if the 0s outnumber the 1s before that point, then we assume the gadget evaluated to 0 and move on to the next one.

By analyzing this as the "monkey on a cliff" problem, it is not hard to see that a 1 gadget is correctly labelled as such with constant probability. The total number of runs of $\hat{\Pi}$ on 0-inputs will, on expectation, be at most $O(n)$, while the total number of runs of $\hat{\Pi}$ on 1-inputs will be at most $O(\log n)$ on expectation; since we avoided the $O(\log n)$ loss from amplification, this protocol is more efficient, and we shave a log factor from the $\mathrm{QIC}(\Pi, \mu_0)$ dependence.[8] ◀

We are now ready to prove Theorem 21.

Proof. (of Theorem 21.) Using Lemma 25, we get a protocol Π_2 computing $\mathrm{OR}_n \circ G$ such that for any $z \in \{0,1\}^n$, $\mathrm{QIC}(\Pi_2, \mu_z) = O(n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1))$. Using Lemma 24, we get a protocol Π_3 computing $G^{\oplus n}$ such that for any $z \in \{0,1\}^n$,

$$\begin{aligned}
&\mathrm{QIC}(\Pi_3, \mu_z)\\
&= O\Big((n^{3/2} \cdot \mathrm{QIC}(\Pi, \mu_0) + \sqrt{n}\log n \cdot \mathrm{QIC}(\Pi, \mu_1))\cdot\\
&\quad \log\left(2 + \frac{n^6}{n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)}\right)\Big).
\end{aligned}$$

Finally, using Lemma 22, we get a protocol Π_4 computing G such that

$$\begin{aligned}
&\mathrm{QIC}(\Pi_4, \mu_0) + \mathrm{QIC}(\Pi_4, \mu_1)\\
&= O\left(\left(\sqrt{n} \cdot \mathrm{QIC}(\Pi, \mu_0) + \frac{\log n}{\sqrt{n}} \cdot \mathrm{QIC}(\Pi, \mu_1)\right)\cdot\right.\\
&\quad \left.\log\left(2 + \frac{n^6}{n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)}\right)\right).
\end{aligned}$$

Moreover, by negating the output of G, such a protocol Π_4 also exists with the μ_0 and μ_1 reversed.

Now, assume without loss of generality that $\mathrm{QIC}(\Pi, \mu_0) \le \mathrm{QIC}(\Pi, \mu_1)$. Let ℓ be the ratio $\mathrm{QIC}(\Pi, \mu_1)/\mathrm{QIC}(\Pi, \mu_0) \ge 1$ (here we use the assumption that $\mathrm{QIC}(\Pi, \mu_0) > 0$ and that $\mathrm{QIC}(\Pi, \mu_1) > 0$). Let $n \in \mathbb{N}^+$ be $\lceil 2\ell \log 2\ell \rceil$. Note that n is at most $3\ell \log 2\ell$, so $n = \Theta(\ell \log 2\ell)$ and $\log n = \Theta(\log 2\ell)$. Using this value of n, we get Π' such that

[8] We thank Thomas Watson and Mika Göös for pointing out this "monkey on a cliff" strategy for computing OR on a noisy oracle.

$$\mathrm{QIC}(\Pi', \mu_0) + \mathrm{QIC}(\Pi', \mu_1)$$
$$= O\left(\sqrt{\mathrm{QIC}(\Pi, \mu_0)\,\mathrm{QIC}(\Pi, \mu_1)}\,\log^{1/2} \frac{\mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1)}{\sqrt{\mathrm{QIC}(\Pi, \mu_0), \mathrm{QIC}(\Pi, \mu_1)}} \cdot \log(2+\alpha)\right),$$

where

$$\alpha = \frac{(\mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1))^{11}}{\mathrm{QIC}(\Pi, \mu_0)^6\,\mathrm{QIC}(\Pi, \mu_1)^6}.$$

If μ_0 and μ_1 are nontrivial, so that we have (say) $\mathrm{QIC}(\Pi, \mu_0) > 1/r^{10}$ and $\mathrm{QIC}(\Pi, \mu_1) > 1/r^{10}$, this can be simplified to

$$\mathrm{QIC}(\Pi', \mu_0) + \mathrm{QIC}(\Pi, \mu_1) = O(\sqrt{\mathrm{QIC}(\Pi, \mu_0)\,\mathrm{QIC}(\Pi, \mu_1)}\,\log^{3/2} r).$$

Finally, since n is at most polynomial in r, it is not hard to check that each of these reductions increases the number of rounds by only a polynomial factor in r, so the final protocol Π' has number of rounds $\mathrm{poly}(r)$. ◀

5.3 Proving the lifting theorem

▶ **Theorem 17.** *Let $f\colon \{0,1\}^n \to \Sigma$ be a relation (where $n \in \mathbb{N}^+$ and Σ is a finite alphabet) and let $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a communication function which contains both AND_2 and OR_2 as subfunctions. Then*

$$\mathrm{QCC}(f \circ G) = \tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

Proof. Let μ_0' and μ_1' be the distributions for G provided by Theorem 19. Let μ_0 be the equal mixture of μ_0' and the uniform distribution over 0-inputs to the AND_2 gadget inside of G, and let μ_1 be the equal mixture of μ_1' and the uniform distribution over the 1-inputs to the OR_2 gadget inside of G. We note that for any protocol Π, $\mathrm{QIC}(\Pi, \mu_0) \geq \mathrm{QIC}(\Pi, \mu_0')/2$ and $\mathrm{QIC}(\Pi, \mu_1) \geq \mathrm{QIC}(\Pi, \mu_1')/2$. By [12], if Π has r rounds, $\mathrm{QIC}(\Pi, \mu_0), \mathrm{QIC}(\Pi, \mu_1) = \Omega(1/r)$. So μ_0 and μ_1 are nontrivial for G.

Let Π be a protocol computing $f \circ G$ to error ϵ, and let r be the number of rounds used by Π, and let T be the communication cost of Π. For $z \in \{0,1\}^n$, we define

$$q'(z,i) := \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B_t')_{\mu_z} + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A_t')_{\mu_z}$$

where C_t is the message in the t-th round of Π and A_t', B_t' are Alice and Bob's memory registers (which don't necessarily have safe copies of their inputs). By the chain rule of mutual information, we have

$$\sum_{i=1}^{n} q'(z,i) = \mathrm{QIC}(\Pi, \mu_z) \leq T$$

for all $z \in \{0,1\}^n$. A feasible weight scheme $q(z,i)$ for $\mathrm{Adv}_1(f)$ will be defined by normalizing $q'(z,i)$.

Let $z, w \in \{0,1\}^n$ be such that $f(z)$ and $f(w)$ are disjoint, and such that their Hamming distance is 1. Let $i \in [n]$ be the bit on which they disagree, so that $z^i = w$ (where z^i denotes the string z with bit i flipped). Suppose without loss of generality that $z_i = 1$ and $w_i = 0$. In order to lower bound $q'(z,i) \cdot q'(w,i)$, we will use the protocol Π for $f \circ G$ to construct a protocol Π' for G.

The protocol Π' is given by [41] (Lemma 4). Alice and Bob start with the shared entangled state of Π, as well the $\widetilde{X}_{-i}X_{-i}\widetilde{Y}_{-i}Y_{-i}$ registers of their inputs and purification according to $\mu_{z_{-i}}$ $(=\mu_{w_{-i}})$ in Π, with Alice holding $A_0\widetilde{X}_{<i}\widetilde{Y}_{<i}X_{-i}$ and Bob holding $B_0\widetilde{X}_{>i}\widetilde{Y}_{>i}Y_{-i}$ (here X_{-i} denotes $X_1 \dots X_{i-1}X_{i+1} \dots X_n$ and the same is true for other variables). They will embed their inputs for Π', which we call X', Y' (with purifications $\widetilde{X}'\widetilde{Y}'$), into the i-th input register for Π (with $\widetilde{X}', \widetilde{Y}'$ being embedded as $\widetilde{X}_i, \widetilde{Y}_i$), and use their shared entanglement for the rest of the input registers, to run Π. After running Π, they will output 1 if Π outputs a symbol in $f(z)$ (outputting 0 otherwise). Note that since Π outputs a symbol in $f(z)$ with probability at least $1-\epsilon$ when given an input from $(G^{\oplus n})^{-1}(z)$ and with probability at most ϵ when given an input from $(G^{\oplus n})^{-1}(w)$ (since $f(w) \cap f(z) = \varnothing$), it follows that Π' succeeds to error ϵ on worst-case inputs to G.

We now analyze the information cost of Π'. Against the distribution μ_0,

$$\begin{aligned}
\mathrm{QIC}(\Pi', \mu_0) &= \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B'_t)_{\mu_{w_{-i}} \otimes \mu_{w_i}} \\
&\quad + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A'_t)_{\mu_{w_{-i}} \otimes \mu_{w_i}} \\
&= \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B'_t)_{\mu_w} + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A'_t)_{\mu_w} \\
&= q'(w, i).
\end{aligned}$$

Similarly, $\mathrm{QIC}(\Pi', \mu_1) = q(z, i)$, so we have

$$\sqrt{q'(z,i)q'(w,i)} = \sqrt{\mathrm{QIC}(\Pi', \mu_0)\,\mathrm{QIC}(\Pi', \mu_1)}.$$

By Theorem 21, there is a protocol Π'' such that

$$\sqrt{q'(z,i)q'(w,i)} = \Omega\left(\frac{\mathrm{QIC}(\Pi'', \mu_0) + \mathrm{QIC}(\Pi'', \mu_1)}{\operatorname{polylog} r}\right).$$

By the choice of μ_0 and μ_1, we therefore have

$$\sqrt{q'(z,i)q'(w,i)} = \Omega(\mathrm{QICZ}(G)/\operatorname{polylog} r),$$

and hence by taking $q(z,i) = O(\operatorname{polylog} r/\mathrm{QICZ}(G)) \cdot q'(z,i)$, we get $q(z,i)q(w,i) \geq 1$. If we start with a protocol Π with number of rounds r at most $\mathrm{QCC}_\epsilon(f \circ G)$, we conclude

$$\mathrm{QCC}_\epsilon(f \circ G) = \tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)),$$

as desired. ◀

6 New query relations

In this section, we prove our new relationships in query complexity. We start by showing that $\mathrm{cfbs}(f)$ is equivalent to $\mathrm{CAdv}(f)$ for partial functions. To do so, we will first need the well-known dual form for the fractional block sensitivity at a specific input, $\mathrm{fbs}(x, f)$. This dual form can be derived by writing the weight scheme defining $\mathrm{fbs}(x, f)$ as a linear program, and taking the dual; this gives a minimization program in which $\mathrm{fbs}(x, f)$ is the minimum, over weight schemes $q(i) \geq 0$ assigned to each $i \in [n]$ that satisfy $\sum_{i \in B} q(i) \geq 1$ for each sensitive block $B \subseteq [n]$ of x (with respect to f), of the sum $\sum_{i \in [n]} q(i)$. See any of [2, 40, 28] for a formal proof.

▶ **Lemma 26.** $\mathrm{cfbs}(f) \le 2\,\mathrm{CAdv}(f)$.

Proof. Let $q(x,i)$ be a feasible weight scheme for $\mathrm{CAdv}(f)$ with objective value equal to $\mathrm{CAdv}(f)$. We construct a completion f' of f as follows. For each $z \notin \mathrm{Dom}(f)$, let $z' \in \mathrm{Dom}(f)$ be the input in the domain of f which minimizes $\sum_{i:z'_i \neq z_i} q(z',i)$. Set $f'(z) = f(z')$. Now let x be any input in $\mathrm{Dom}(f)$; we wish to upper bound $\mathrm{fbs}(x,f')$.

To this end, we pick weights $q(i) = 2q(x,i)$, and claim that they are a feasible solution to the fractional block sensitivity for f' at x. Let B be any sensitive block for x with respect to f'. Then x^B is some input z which disagrees with x exactly on the bits in B, and which satisfies $f'(z) \neq f(x)$. Let z' be the input in $\mathrm{Dom}(f)$ which minimizes $\sum_{i:z'_i \neq z_i} q(z',i)$, so that $f'(z) = f(z')$. Then $f(z') \neq f(x)$, and in fact z' must be closer to z than to x; hence

$$\begin{aligned}
\sum_{i\in B} q(i) &= \sum_{i:x_i\neq z_i} 2q(x,i)\\
&\ge \sum_{i:x_i\neq z_i} q(x,i) + \sum_{i:z'_i\neq z_i} q(z',i)\\
&\ge \sum_{i:x_i\neq z_i} \min\{q(x,i),q(z',i)\} + \sum_{i:z'_i\neq z_i} \min\{q(x,i),q(z',i)\}\\
&\ge \sum_{i:x_i\neq z'_i} \min\{q(x,i),q(z',i)\} \ge 1.
\end{aligned}$$

We conclude that $q(i)$ is feasible. Its objective value is $\sum_{i\in[n]} q(i) = \sum_{i\in[n]} 2q(x,i) \le 2\,\mathrm{CAdv}(f)$, and hence $\mathrm{cfbs}(f) \le 2\,\mathrm{CAdv}(f)$, as desired. ◀

▶ **Lemma 27** (Krišjānis Prūsis, personal communication)**.** $\mathrm{CAdv}(f) \le \mathrm{cfbs}(f)$.

Proof. Let f' be a completion of f for which $\mathrm{fbs}(x,f') \le \mathrm{cfbs}(f)$ for all $x \in \mathrm{Dom}(f)$. For each $x \in \mathrm{Dom}(f)$, let $q_x(i)$ be a feasible weight scheme for the minimization problem of $\mathrm{fbs}(x,f')$ which satisfies $\sum_{i\in[n]} q_x(i) \le \mathrm{fbs}(x,f') \le \mathrm{cfbs}(f)$ and for each sensitive block B of f', $\sum_{i\in B} q_x(i) \ge 1$.

We construct a weight scheme for $\mathrm{CAdv}(f)$ by setting $q(x,i) = q_x(i)$ for all $x \in \mathrm{Dom}(f)$. We claim this weight scheme is feasible. To see this, let $x, y \in \mathrm{Dom}(f)$ be such that $f(x) \neq f(y)$. Define the input $z \in \{0,1\}^n$ such that $z_i = x_i$ if $x_i = y_i$, and otherwise, $z_i = x_i$ if $q(x,i) \ge q(y,i)$ and $z_i = y_i$ if $q(y,i) > q(x,i)$. Suppose that $f'(z) \neq f(x)$. Then

$$\begin{aligned}
\sum_{i:x_i\neq y_i} \min\{q(x,i),q(y,i)\} &= \sum_{i:x_i\neq z_i} \min\{q(x,i),q(y,i)\} + \sum_{i:y_i\neq z_i} \min\{q(x,i),q(y,i)\}\\
&\ge \sum_{i:x_i\neq z_i} q(x,i) + \sum_{i:y_i\neq z_i} q(y,i) \ge 1.
\end{aligned}$$ ◀

▶ **Lemma 28.** *For any (possibly partial) Boolean function f, we have*

$$\widetilde{\deg}_\epsilon(f) \ge \frac{\sqrt{2}}{\pi}\sqrt{(1-2\epsilon)\,\mathrm{fbs}(f)}.$$

Proof. Let $x \in \mathrm{Dom}(f)$ be such that $\mathrm{fbs}(x,f) = \mathrm{fbs}(f)$. By negating the input bits of f if necessary, we may assume that $x = 0^n$ (note that negating input bits does not affect $\mathrm{fbs}(f)$ or $\widetilde{\deg}(f)$). By negating the output of f if necessary, we can further assume that $f(0^n) = 0$. Let p be a polynomial of degree $\widetilde{\deg}_\epsilon(f)$ which approximates f to error ϵ.

Let $\textsc{PrOR}_k$ be the promise problem on k bits whose domain contains all the strings of Hamming weights 0 or 1, and which outputs 0 given 0^k and outputs 1 given an input of Hamming weight 1.

We give an exact polynomial representation of this function. To do so, let T_d be the Chebyshev polynomial of degree d; this is the single-variate real polynomial satisfying $T_d(\cos\theta) = \cos(d\theta)$. This polynomial is bounded in $[-1, 1]$ on the interval $[-1, 1]$. Moreover, it satisfies $T_d(1) = 1$ and $T_d(\cos(\pi/d)) = -1$. Hence the polynomial $r(t) = (1 - T_d(1 - (1 - \cos(\pi/d))t))/2$ evaluates to 0 at $t = 0$ and to 1 at $t = 1$. Moreover, since this T_d is bounded in $[-1, 1]$ on the interval $[-1, 1]$, we conclude that $r(t)$ is bounded in $[0, 1]$ on the interval $[0, 2/(1 - \cos(\pi/d))]$. Since $\cos(z) \geq 1 - z^2/2$, we have $2/(1 - \cos(\pi/d)) \geq 4d^2/\pi^2$. Hence $r(t)$ is bounded in $[0, 1]$ on the interval $[0, 4d^2/\pi^2]$. If we pick d such that $4d^2/\pi^2 \geq k$, that is, d at least $\lceil \pi\sqrt{k}/2 \rceil$, then we would know that $r(t)$ is bounded on $[0, k]$. In that case, the k-variate polynomial $q(x) = r(x_1 + x_2 + \cdots + x_k)$ would exactly compute $\textsc{PrOR}_k$, and it would have degree at most $\lceil \pi\sqrt{k}/2 \rceil \leq \pi\sqrt{k}/2 + 1$.

Next, consider the function $f \circ \textsc{PrOR}_k$. We can approximate this function to error ϵ simply by plugging in n independent copies of the polynomial q into the variables of the polynomial p. This means that the approximate degree of $f \circ \textsc{PrOR}_k$ to error ϵ is at most $\widetilde{\deg}_\epsilon(f) \cdot (\pi\sqrt{k}/2 + 1)$.

On the other hand, we now claim that for appropriate choice of k, we have $\mathrm{bs}(0^{kn}, f \circ \textsc{PrOR}_k) \geq k\,\mathrm{fbs}(0^n, f)$, and hence $\mathrm{bs}(f \circ \textsc{PrOR}_k) \geq k\,\mathrm{fbs}(f)$. To see this, let $\{w_B\}_B$ be an optimal weight scheme for the fractional block sensitivity of 0^n with respect to f, so that $\sum_{B:i\in B} w_B \leq 1$ and $\sum_B w_B = \mathrm{fbs}(f)$. Note that since fractional block sensitivity is a linear program, the optimal solution can be taken to be rational; let L be a common denominator of all the weights, so that Lw_B is an integer for each sensitive block B. Now take k to be an integer multiple of L. For each sensitive block B of 0^n with respect to f, we define kw_B different sensitive blocks of 0^{kn} with respect to $f \circ \textsc{PrOR}_k$, such that all of the new blocks are mutually disjoint. To do so, we simply use a different bit in each copy of $\textsc{PrOR}_k$ for each block. Since the sum of weights w_B for blocks that use bit i of the input to f is at most 1, the total number of new blocks we will generate that use copy i of $\textsc{PrOR}_k$ is at most k, and hence we can give each block a different bit of that copy of $\textsc{PrOR}_k$. The total number of disjoint blocks will then be $k\sum_B w_B = k\,\mathrm{fbs}(f)$.

We conclude that $\mathrm{bs}(f \circ \textsc{PrOR}_k) \geq k\,\mathrm{fbs}(f)$ as long as k is a multiple of a certain integer L. Now, by a standard result [9, 15], we know that the approximate degree to error ϵ of a (possibly partial) Boolean function is at least the square root of its block sensitivity; more explicitly, we have

$$\widetilde{\deg}_\epsilon(f \circ \textsc{PrOR}_k) \geq \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)}\,\mathrm{bs}(f \circ \textsc{PrOR}_k)} \geq \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)} k\,\mathrm{fbs}(f)}.$$

Combined with our upper bound on this degree, we have

$$\widetilde{\deg}_\epsilon(f) \cdot (\pi\sqrt{k}/2 + 1) \geq \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)} k\,\mathrm{fbs}(f)},$$

and since k can go to infinity, we must have

$$\widetilde{\deg}_\epsilon(f) \geq \frac{\sqrt{2}}{\pi}\sqrt{\frac{(1-2\epsilon)}{1-\epsilon}\,\mathrm{fbs}(f)},$$

from which the desired result follows. ◀

▶ **Theorem 29.** *For all (possibly partial) Boolean functions f, we have*

$$\widetilde{\deg}_\epsilon(f) \geq \frac{\sqrt{(1-2\epsilon)\,\mathrm{cfbs}(f)}}{\pi}.$$

Proof. Let p be a polynomial which approximates f to error ϵ. Then $p(x) \in [0,1]$ for all $x \in \{0,1\}^n$, so define $f'(x)$ by $f'(x) = 1$ if $p(x) \geq 1/2$ and $f'(x) = 0$ if $p(x) < 1/2$. It is clear that $f'(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$, so f' is a completion of f. Let $x \in \mathrm{Dom}(f)$ be an input so that $\mathrm{fbs}(x, f') \geq \mathrm{cfbs}(f)$. To complete the proof, it will suffice to lower bound the degree of p by $\Omega(\sqrt{\mathrm{fbs}(x, f')})$.

Suppose without loss of generality that $f(x) = 0$ (otherwise, negate f and f' and replace p with $1-p$). Then we know that $p(x) \in [0,\epsilon]$, and that for any $y \in \{0,1\}^n$ such that $f'(x) \neq f'(y)$, we have $p(y) \in [1/2, 1]$. This means that the polynomial $q(z) = (2p(z) + 1 - 2\epsilon)/(3 - 2\epsilon)$ has the same degree as p, is bounded in $[0,1]$ on $\{0,1\}^n$, and approximates f' to error $1/(3-2\epsilon)$ on the input x and on all inputs $y \in \{0,1\}^n$ such that $f'(x) \neq f'(y)$. In other words, consider the partial function f'_x which is the restriction of f' to the promise set $\{x\} \cup \{y \in \{0,1\}^n : f'(y) \neq f'(x)\}$. Then q approximates f'_x to error $1/(3-2\epsilon)$, and has the same degree as p. Now, it is not hard to see that $\mathrm{fbs}(f'_x) = \mathrm{fbs}(x, f')$. Hence it suffices to lower bound the degree of q by $\Omega(\sqrt{\mathrm{fbs}(f'_x)})$. Such a lower bound follows from Lemma 28; indeed, we conclude that the degree of p is at least

$$\frac{1}{\pi}\sqrt{\frac{1-2\epsilon}{1-\epsilon}\,\mathrm{cfbs}(f)}. \qquad \blacktriangleleft$$

▶ **Theorem 30.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2.$$

Proof. Let f be a (possibly partial) Boolean function, and $q(x,i)$ be a feasible weight scheme for $\mathrm{Adv}(f)$ that has $\sum_{i\in[n]} q(x,i) \leq \mathrm{Adv}(f)$ for all i. Fix any $x, y \in \mathrm{Dom}(f)$ such that $f(x) \neq f(y)$. Then

$$1 \leq \sum_{i:x_i\neq y_i} \sqrt{q(x,i)q(y,i)} = \sum_{i:x_i\neq y_i} \sqrt{\min\{q(x,i),q(y,i)\}\max\{q(x,i),q(y,i)\}}$$

$$\leq \sqrt{\sum_{i:x_i\neq y_i} \min\{q(x,i),q(y,i)\} \cdot \sum_{i:x_i\neq y_i} \max\{q(x,i),q(y,i)\}}.$$

Note that $\max\{q(x,i),q(y,i)\} \leq q(x,i) + q(y,i)$, and we know the sum over i of $q(x,i)$ and $q(y,i)$ are each at most $\mathrm{Adv}(f)$. Hence we get

$$\sum_{i:x_i\neq y_i} \max\{q(x,i),q(y,i)\} \leq 2\,\mathrm{Adv}(f),$$

and hence

$$\sum_{i:x_i\neq y_i} \min\{q(x,i),q(y,i)\} \geq \frac{1}{2\,\mathrm{Adv}(f)}.$$

This means that if we scale the weights $q(x,i)$ up by a uniform factor of $2\,\mathrm{Adv}(f)$, the resulting weight scheme $q'(x,i)$ will be feasible for $\mathrm{CAdv}(f)$. The objective value of this new weight scheme will then be the maximum over x of

$$\sum_{i\in[n]} q'(x,i) = 2\,\mathrm{Adv}(f) \sum_{i\in[n]} q(x,i) \leq 2\,\mathrm{Adv}(f)^2,$$

so $\mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2$, as desired. ◀

References

1 Scott Aaronson. Lower bounds for local search by quantum arguments. *SIAM Journal on Computing*, 35(4):804–824, 2006. Previous version in STOC 2004. doi:10.1137/s0097539704447237.

2 Scott Aaronson. Quantum certificate complexity. *Journal of Computer and System Sciences*, 74(3):313–322, 2008. Previous version in CCC 2003. doi:10.1016/j.jcss.2007.06.020.

3 Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem, 2020. Preprint,. arXiv:2010.12629.

4 R Alicki and M Fannes. Continuity of quantum conditional information. *Journal of Physics A: Mathematical and General*, 37(5):L55–L57, 2004. doi:10.1088/0305-4470/37/5/l01.

5 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Previous version in STOC 2000. doi:10.1006/jcss.2002.1826.

6 Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All classical adversary methods are equivalent for total functions. In *Proceedings in the 35th Symposium on Theoretical Aspects of Computer Science (STACS)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2018. doi:10.4230/LIPICS.STACS.2018.8.

7 Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 209–218, 2002. doi:10.1109/SFCS.2002.1181944.

8 Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. Previous version in FOCS 2002. doi:10.1016/j.jcss.2003.11.006.

9 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Previous version in FOCS 1998. doi:10.1145/502090.502097.

10 Aleksandrs Belovs. Quantum algorithms for learning symmetric juntas via the adversary bound. *Computational Complexity*, 24(2):255–293, 2015. Previous version in CCC 2014. doi:10.1007/s00037-015-0099-2.

11 Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018. doi:10.1109/focs.2018.00040.

12 Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-optimal bounds on the bounded-round quantum communication complexity of disjointness. *SIAM Journal on Computing*, 47(6):2277–2314, 2018. Previous version in FOCS 2015. doi:10.1137/16m1061400.

13 Mark Braverman and Omri Weinstein. An interactive information odometer and applications. In *Proceedings of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2015. doi:10.1145/2746539.2746548.

14 Harry Buhrman, Matthias Christandl, and Christian Schaffner. Complete insecurity of quantum protocols for classical two-party computation. *Physical Review Letters*, 109(16), 2012. doi:10.1103/physrevlett.109.160501.

15 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

16 Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting for BPP using inner product. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2019. doi:10.4230/LIPICS.ICALP.2019.35.

17 Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudo-random properties. *Computational Complexity*, 28(4):617–659, 2019. doi:10.1007/s00037-019-00190-7.

18 Roger Colbeck. Impossibility of secure two-party classical computation. *Physical Review A*, 76(6), 2007. doi:10.1103/physreva.76.062308.

19 Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In *Proceedings of the 10th Theory of Cryptography Conference (TCC)*, pages 281–296. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-36594-2_16.

20 Justin Gilmer, Michael Saks, and Sudarshan Srinivasan. Composition limits and separating examples for some boolean function complexity measures. *Combinatorica*, 2016. Previous version in CCC 2013. doi:10.1007/s00493-014-3189-x.

21 Mika Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076. IEEE, 2015. doi:10.1109/FOCS.2015.69.

22 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. Previous version in STOC 2015. doi:10.1137/15M103145X.

23 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM Journal on Computing*, 2018. Previous version in FOCS 2015. doi:10.1137/16M1059369.

24 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM Journal on Computing*, 47(5):1778–1806, 2018. Previous version in STOC 2014. doi:10.1137/16m1082007.

25 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM Journal on Computing*, 49(4):FOCS17–441–FOCS17–461, 2020. Previous version in FOCS 2017. doi:10.1137/17m115339x.

26 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

27 Trinh Huynh and Jakob Nordstrom. On the virtue of succinct proofs. In *Proceedings of the 44th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2012. doi:10.1145/2213977.2214000.

28 Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chicago Journal of Theoretical Computer Science*, 2016. doi:10.4086/cjtcs.2016.008.

29 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996. doi:10.1017/cbo9780511574948.

30 Sophie Laplante and Frédéric Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. *SIAM Journal on Computing*, 2008. Previous version in CCC 2004. doi:10.1137/050639090.

31 Mathieu Laurière and Dave Touchette. The flow of information in interactive quantum protocols: the cost of forgetting. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2017. doi:10.4230/LIPICS.ITCS.2017.47.

32 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

33 Hoi-Kwong Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154–1162, 1997. doi:10.1103/physreva.56.1154.

34 Noam Nisan. Crew prams and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. Previous version in STOC 1989. doi:10.1137/0220062.

35 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–569. SIAM, 2011. `doi:10.1137/1.9781611973082.44`.

36 Louis Salvail, Christian Schaffner, and Miroslava Sotáková. Quantifying the leakage of quantum protocols for classical two-party cryptography. *International Journal of Quantum Information*, 13(04):1450041, 2014. `doi:10.1142/s0219749914500415`.

37 Alexander A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011. Previous version in STOC 2008. `doi:10.1137/080733644`.

38 Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. `doi:10.2140/pjm.1958.8.171`.

39 Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2, 2006. Previous version in ICALP 2005. `doi:10.4086/toc.2006.v002a001`.

40 Avishay Tal. Properties and applications of boolean function composition. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 441–454, 2013. `doi:10.1145/2422436.2422485`.

41 Dave Touchette. Quantum information complexity. In *Proceedings of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2015. `doi:10.1145/2746539.2746613`.

42 Xiaodi Wu, Penghui Yao, and Henry Yuen. Raz-mckenzie simulation with the inner product gadget, 2017. Preprint. `arXiv:2017/010`.