

Guesswork With Quantum Side Information

Eric P. Hanson^{1b}, Vishal Katariya^{1b}, Nilanjana Datta, and Mark M. Wilde^{1b}, *Senior Member, IEEE*

Abstract—What is the minimum number of guesses needed on average to guess a realization of a random variable correctly? The answer to this question led to the introduction of a quantity called *guesswork* by Massey in 1994, which can be viewed as an alternate security criterion to entropy. In this paper, we consider the guesswork in the presence of quantum side information, and show that a general sequential guessing strategy is equivalent to performing a single quantum measurement and choosing a guessing strategy based on the outcome. We use this result to deduce entropic one-shot and asymptotic bounds on the guesswork in the presence of quantum side information, and to formulate a semi-definite program (SDP) to calculate the quantity. We evaluate the guesswork for a simple example involving the BB84 states, both numerically and analytically, and we prove a continuity result that certifies the security of slightly imperfect key states when the guesswork is used as the security criterion.

Index Terms—Guesswork, guesswork with quantum side information, quantum cryptography, quantum key distribution, semi-definite program.

I. INTRODUCTION

INFORMATION theory, among other things, concerns the security of messages against attacks by malicious agents. Conventionally, it is accepted that the more unpredictable a message is, and the higher the (Shannon) entropy of the distribution from which it is drawn, the more secure it is to brute force attacks. Therefore, when establishing a secret key or a cipher, the gold standard is to choose a key whose elements are picked uniformly at random from some alphabet.

Entropy, however, is not the only such criterion for security. Another relevant quantity, which is also maximized by messages drawn uniformly, is the guesswork. First put forth by Massey [1], the quantity is operationally described by the following guessing game. Consider the problem of guessing a realization of a random variable X , taking values in a finite alphabet $\mathcal{X}$, by asking questions of the form “Is $X = x$?”.

Manuscript received December 31, 2020; revised July 28, 2021; accepted September 25, 2021. Date of publication October 8, 2021; date of current version December 23, 2021. The work of Eric P. Hanson was supported by the Cantab Capital Institute for the Mathematics of Information (CCIMI). The work of Vishal Katariya was supported by the Louisiana State University Economic Development Assistantship. The work of Mark M. Wilde was supported by the U.S. National Science Foundation under Grant 1907615. (Corresponding author: Vishal Katariya.)

Eric P. Hanson and Nilanjana Datta are with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 0WA, U.K.

Vishal Katariya and Mark M. Wilde are with the Department of Physics and Astronomy, Hearne Institute for Theoretical Physics, and the Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: vkatariya8@gmail.com).

Communicated by S. Beigi, Associate Editor for Quantum Information Theory.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIT.2021.3118878>.

Digital Object Identifier 10.1109/TIT.2021.3118878

0018-9448 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

The *guesswork* $G(X)$ is defined as the minimum value of the average number of questions of this form that needs to be asked until the answer is “yes”. That is,

$$G(X) = \sum_{k=1}^{|\mathcal{X}|} k \cdot p_G(k) \quad (1)$$

where $p_G(k)$ is the probability of the k th guess being correct. In the real world, questions of this form arise from query access to a resource; for example, if a hacker is attempting to guess a user’s password on an online portal, he or she can only ask this kind of question (as opposed to, say, “Is $X \geq x$?”) and is allowed a limited number of guesses before being locked out. Therefore, for someone setting up a password, the number of guesses allowed by the portal provides the operational security criterion against which his or her password must compare.

In contrast, the entropy of a distribution is approximately the minimum value of the average number of guesses required to obtain the correct guess when one is allowed to ask questions of the form, “Is $X \in \tilde{\mathcal{X}}$?”, where each $\tilde{\mathcal{X}}$ is some subset of the alphabet $\mathcal{X}$ [2, Theorem 5.4.1]. Qualitatively speaking, entropy can be considered to be the query complexity of a binary search-type algorithm, whereas guesswork corresponds to the query complexity of a linear search-type algorithm [3]. Figure 1 illustrates this difference. It is well known that binary search has a smaller complexity than linear search, which leads to the simple claim that the entropy of a distribution is always less than the guesswork. Massey [1] proved the following stronger lower bound on the guesswork in terms of the Shannon entropy of the random variable X :

$$G(X) \geq \frac{1}{4} 2^{H(X)} + 1, \quad (2)$$

provided that $H(X) \geq 2$ bits.

Massey [1] also showed that the optimal algorithm to minimize the guesswork, and even the positive moments of the number of guesses, consists of the intuitive strategy of simply guessing elements in decreasing order of their probability of occurrence. Guesswork can also be considered in the presence of classical side information given in the form of a random variable Y that is correlated with X . In this case, the guesswork is the minimal number of questions of the form “Is $X = x$?” that is required on average to obtain the correct answer, given the value of Y . The average is taken over the number of guesses, with each guess number weighted by its probability of being correct. That is,

$$G(X|Y = y) = \sum_{k=1}^{|\mathcal{X}|} k \cdot p_G(k) \quad (3)$$

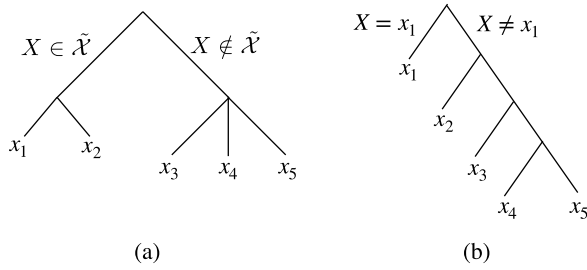


Fig. 1. Consider an unknown value of X in the set $\{x_1, x_2, x_3, x_4, x_5\}$. The search tree for finding the correct value of X when asking questions of the form (a) “Is $X \in \tilde{\mathcal{X}}$?” or of the form (b) “Is $X = x$?” The former search tree, similar to that in binary search, has fewer branches and the relevant operational quantity is the entropy of the prior distribution. The specific question we ask in this tree is as follows: “Is $X \in \tilde{\mathcal{X}} = \{x_1, x_2\}$?” The latter tree, similar to that of linear search, characterizes the scenario of guesswork. The tree encapsulates a strategy in which one sequentially guesses x_1, x_2 , and so on, until x_5 .

where in this case, $p_G(k)$ is the probability of the k th guess being correct, given that $Y = y$. The optimal guessing strategy is simply to guess in decreasing order of conditional probability $p_{X|Y}(x|y)$. Arikan [4] obtained upper and lower bounds on the guesswork and its positive moments, in this scenario as well as in the case without side information. Further work on guesswork in the classical setting has been done in [5]–[12].

In this paper, we consider a natural generalization of the above guessing problem to the case in which the classical side information is replaced by quantum side information. This generalization was first considered in [13]. In this case, the guesser (say, Bob) holds a quantum system B , instead of a classical random variable (or equivalently, a classical system) Y . Here, the joint state of X and B is given by a classical-quantum (c-q) state, which we denote as ρ_{XB} (see Section II for details). Suppose that Alice possesses the classical system X containing the value of the letter x , which is to be guessed by Bob. We define guesswork in the presence of quantum side information to be the minimum number of guesses needed, on average, for Bob to correctly guess Alice’s choice, by performing a general sequential protocol, as follows. Bob acts on his system B with a quantum instrument, yielding a classical outcome $\hat{x}$ which he guesses, as well as a post-measurement state on system B . If his guess is incorrect, he performs another instrument on his system B (possibly adapted based on his previous guess), and repeats the protocol until he either guesses correctly or runs out of guesses (which might be the case if he is allowed a limited number of guesses $K < |\mathcal{X}|$).

While the case of classical side information admits a very simple optimal strategy (which amounts to simply sorting the conditional probabilities $p_{X|Y}(\cdot|y)$ in non-increasing order and guessing accordingly), the quantum case requires measurement on the quantum system B , which potentially disturbs the state of B , a priori complicating the analysis of the sequence of guesses in the optimal strategy. We show in Section III-B, however, that a general sequential strategy is in fact equivalent to performing a single generalized measurement yielding a classical random variable Y of outcomes and then doing the optimal strategy using this Y as the classical side information.

The earlier work [13] instead defined guesswork with quantum side information as the latter quantity, i.e., a measured version of the guesswork in the presence of classical side information. We prove here that these definitions are equivalent, and thus there is ultimately no difference between them, but we consider the definition in terms of a sequential protocol to be a more natural one. Moreover, the above-mentioned equivalence is proved via an explicit construction, allowing such a guessing strategy to be implemented sequentially. The single-measurement protocol could in general involve making a measurement with exponentially (in $|\mathcal{X}|$) many outcomes. Hence it may be more efficient to implement it instead as a sequence of (linearly-many) measurements with linearly-many outcomes, as allowed by the above construction.

Moreover, we consider a slight generalization of the guesswork in which Bob may make only $K \leq |\mathcal{X}|$ guesses in total, and in which the “cost” of needing to make k guesses is given by a vector $\vec{c} = (c_1, c_2, \dots, c_K)$, which could be different from $(1, 2, \dots, K)$, the latter of which corresponds to the expected value. These generalizations can be better models for certain situations; in the password-guessing example, e.g., Bob may be locked out after K guesses and hence is limited to a small number of guesses, or perhaps one has to wait after each guess before making another, and the time that one waits increases with the number of incorrect guesses. We show that this generalized situation (including the guesswork as a special case) admits a semi-definite programming (SDP) representation in which the number of variables scales as $|\mathcal{X}|^K$, and hence smaller values of K yield smaller problems and better scaling with $|\mathcal{X}|$. See Section V for more on the computational aspects of the guesswork.

One can consider a related task, in which one wishes to maximize what is known as the “guessing probability” $p_{\text{guess}}(X|B)$ [14]. In this case, the guesser is given only one attempt to guess the value of X (and hence is free to perform any arbitrary measurement on his system B). The guessing probability is related to the so-called *conditional min-entropy* $H_{\min}(X|B)$ of the c-q state ρ_{XB} . In some sense, we can consider the guesswork to be an extension of the guessing probability. However, the nature of the optimization being done is different: instead of maximizing the probability of success in one attempt, we minimize the total number of guesses required. Therefore, the operations that a guesser performs to minimize the guesswork may be very different from those needed to maximize the guessing probability. Some of the connections between these two tasks have been investigated in [13].

Overview: In Section II, we formally describe the task of guesswork with quantum side information. In Section III, we define classical and quantum guessing strategies in a unified framework, and Theorem 1 states that three classes of quantum strategies are equivalent. In Section IV, we establish one-shot and asymptotic entropic bounds on the guesswork, using analogous bounds developed by Arikan for the case of classical side information [4]. In Section V, we revisit the idea that the guesswork may be formulated as a semi-definite optimization problem (SDP) (originally discussed in [13]), and we use such a representation to prove that the guesswork

in a concave function (see Section VII-A) and a Lipschitz continuous function (see Section VII-B). We discuss the dual formulation of the SDP in Section V-A, a resulting algorithm to efficiently compute upper bounds in Section V-B, and we present a mixed-integer SDP representation in Section V-C. Section VIII shows two simple examples of the guesswork involving the BB84 states and the Y states, and Section IX provides a robustness result for using guesswork as a security criterion.

II. STATEMENT OF THE PROBLEM

Alice chooses a letter $x \in \mathcal{X}$ with some probability $p_X(x)$, where $\mathcal{X}$ is a finite alphabet. This naturally defines a random variable $X \sim p_X(x)$. She then sends a quantum system B to Bob, prepared in the state ρ_B^x , which depends on her choice x . Bob knows the set of states $\{\rho_B^x : x \in \mathcal{X}\}$, and the probability distribution $\{p_X(x) : x \in \mathcal{X}\}$, but he does not know which particular state is sent to him by Alice. Bob's task is to guess x correctly with as few guesses as possible. From Bob's perspective, he therefore has access to the B -part of the c-q state

$$\rho_{XB} := \sum_x p_X(x) |x\rangle\langle x|_X \otimes \rho_B^x. \quad (4)$$

In the purely classical case, this task reduces to the following scenario: Alice holds the random variable $X \sim p_X(x)$, and Bob holds a correlated random variable Y and knows the joint distribution of (X, Y) . In this case ρ_{XB} reduces to the state

$$\rho_{XY} = \sum_x p_X(x) |x\rangle\langle x|_X \otimes \sum_y p_{Y|X}(y|x) |y\rangle\langle y|_Y. \quad (5)$$

In this case, if Bob's random variable Y has value y , then an optimal guessing strategy is to sort the conditional distribution $p_{X|Y}(\cdot|y)$ in non-increasing order so that

$$p_{X|Y}(x_1|y) \geq p_{X|Y}(x_2|y) \geq \dots \geq p_{X|Y}(x_{|\mathcal{X}|}|y) \quad (6)$$

and simply guess first x_1 , then x_2 , etc., until he gets it correct [4]. We note that in the absence of side information, Bob simply guesses in non-increasing order of $p_X(\cdot)$.

In the case in which Bob's system B is quantum, he is allowed to perform any local operations he wishes on B , and then make a first guess x_1 . He is told by Alice whether or not his guess is correct; then he can perform local operations on B , and make another guess, and so forth. We are interested in determining the minimal number of guesses needed on average for a given ensemble $\{p_X(x), \rho_B^x\}_{x \in \mathcal{X}}$ and the associated optimal strategy.

More generally, we allow Bob to make K guesses, with possibly $K < |\mathcal{X}|$. Formally, we assume that Bob always makes all K guesses; any guess after the correct guess simply does not factor into the calculation of the minimal number of guesses (see Section III for a more detailed definition of the minimal number of guesses). Thus, Bob makes a sequence of guesses, $g_1, \dots, g_K \in \mathcal{X}$ with some probability.

We could consider the scenario in which Bob makes a guess x_1 , then learns whether or not the guess was correct, and uses that information to make his second guess x_2 , and so forth.

However, if Bob learns that his j th guess x_j is correct, then it does not matter what he guesses subsequently (it has no bearing on the minimal number of guesses). If the guess is incorrect, then his subsequent guesses do matter, and he should make his next guess accordingly. Hence, in such a protocol, the feedback about whether or not the j th guess is correct does not help, and Bob might as well assume that each guess is incorrect.

III. GUESSING STRATEGIES

When Alice chooses $x^* \in \mathcal{X}$, a guessing strategy for Bob outputs a sequence of guesses $\vec{g} = (g_1, \dots, g_K) \in \mathcal{X}^K$ with some probability $p_{\vec{G}|X}(\vec{g}|x^*)$. Hence, formally, a *guessing strategy for X with K guesses* is a random variable $\vec{G}$ on $\mathcal{X}^K$ that is correlated with X , such that the joint random variable $(X, \vec{G})$ has marginal $X \sim p_X$. Note that the definition of a guessing strategy has no reference to the side information (if any) to which Bob has access; instead, the side information dictates the set of guessing strategies to which Bob has access. This allows various types of side information to be analyzed within a uniform framework; in particular, the set of strategies available when Bob has access to some classical side information Y is described in Section III-A, while the case of quantum side information is described in Section III-B.

We are interested in the minimal number of guesses required to guess x^* correctly with a fixed sequence of guesses $\vec{g}$. This is defined as follows:

$$N(\vec{g}, x^*) := \begin{cases} \min \{j : g_j = x^*\} & g_j = x^* \text{ for some } j = 1, \dots, K \\ \infty & \text{else,} \end{cases} \quad (7)$$

where the outcome ∞ occurs when none of the K guesses are correct. We can view N as a random variable taking values in $\{1, 2, \dots, K, \infty\}$. Given a guessing strategy $\vec{G}$, the quantity of interest is $N(\vec{G}, X)$, the corresponding random variable. We define

$$\mathcal{S}_K(X) := \{N(\vec{G}, X) : X\vec{G} \sim p_{X\vec{G}}\} \quad (8)$$

to be the set of all possible random variables N associated to all guessing strategies $\vec{G}$ with K guesses. We say two guessing strategies $\vec{G}$ and $\vec{G}'$ for X with K guesses are *equivalent* if $N(\vec{G}', X) = N(\vec{G}, X)$.

Note that if $\vec{G}$ and $\vec{G}'$ are two strategies with K guesses for X that differ only in guesses made after guessing the correct answer, then they are equivalent. This formalizes the notion introduced at the end of the previous section: since guesses made after the correct answer do not change the value of $N(\vec{g}, x^*)$, feedback of whether or not $g_j = x^*$ can only lead to equivalent strategies.

A. Classical Strategies

Consider a pair of random variables (X, Y) where X has a finite alphabet $\mathcal{X}$ and Y has a countable alphabet $\mathcal{Y}$. Alice chooses $x^* \in \mathcal{X}$ (with probability $p_X(x^*)$) and Bob is given $y \in \mathcal{Y}$ (with probability $p_{Y|X}(y|x^*)$). Bob's task is to guess x^* . Since Bob's sequence of guesses $(g_1, \dots, g_K)$ can only

depend on x^* via y , a classical guessing strategy $\vec{G}$ is any random variable $\vec{G}$ such that the ordered triple $(X, Y, \vec{G})$ of random variables forms a Markov chain, which we denote as $X - Y - \vec{G}$. Hence, given a joint probability distribution p_{XY} , we define the set of random variables N associated to classical guessing strategies as follows:

$$\mathcal{S}_K^{\text{Classical}}(p_{XY}) := \left\{ N(\vec{G}, X) : X - Y - \vec{G} \right\} \subseteq \mathcal{S}_K(X). \quad (9)$$

B. Equivalence of Quantum Strategies

Let us consider three classes of quantum strategies:

- 1) Measured strategy: Bob performs an arbitrary POVM $\{E_y\}_{y \in \mathcal{Y}}$ on the B system. Let Y be the random variable with outcomes in a finite alphabet $\mathcal{Y}$ corresponding to his measurement outcomes, i.e.

$$p_{Y|X}(y|x) = \text{tr}[E_y \rho_B^x], \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}. \quad (10)$$

Bob then employs a classical guessing strategy on (X, Y) . The set of random variables corresponding to the possible number of guesses under such a strategy is given by

$$\mathcal{S}_K^{\text{Measured}}(\rho_{XB}) := \left\{ N(\vec{G}, X) : X - Y - \vec{G}, \right. \\ \left. Y \text{ satisfies (10) for finite alphabet } \mathcal{Y} \right. \\ \left. \& \text{ POVM } \{E_y\}_{y \in \mathcal{Y}} \right\}. \quad (11)$$

We then observe that

$$\mathcal{S}_K^{\text{Measured}}(\rho_{XB}) \subseteq \mathcal{S}_K(X). \quad (12)$$

- 2) Ordered strategy: Bob performs a measurement with outcomes in $\mathcal{X}^K$, which are identified with guessing orders; i.e., if the outcome is $(x_1, \dots, x_K) \in \mathcal{X}^K$, Bob first guesses x_1 , then x_2 , and so forth. In this case, Bob performs a POVM $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}^K}$ and the guessing strategy $\vec{G}$ is distributed according to

$$p_{\vec{G}|X}(\vec{g}|x) = \text{tr}[E_{\vec{g}} \rho_B^x]. \quad (13)$$

As above, we define

$$\mathcal{S}_K^{\text{Ordered}}(\rho_{XB}) := \left\{ N(\vec{G}, X) : (\vec{G}, X) \text{ satisfy} \right. \\ \left. (13) \text{ for some POVM } \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}^K} \right\} \subseteq \mathcal{S}_K(X). \quad (14)$$

It is evident that

$$\mathcal{S}_K^{\text{Ordered}}(\rho_{XB}) \subseteq \mathcal{S}_K^{\text{Measured}}(\rho_{XB}) \quad (15)$$

because any such ordered strategy is a special type of measured strategy (with $Y = \vec{G}$). However, any measured strategy can in fact be simulated by an ordered strategy. Suppose we have a measured strategy with alphabet $\mathcal{Y}$, POVM $\{E_y\}_{y \in \mathcal{Y}}$, and $\vec{G}$ satisfying $X - Y - \vec{G}$. Then

$$p_{\vec{G}|X}(\vec{g}|x) = \sum_{y \in \mathcal{Y}} p_{\vec{G}|Y}(\vec{g}|y) p_{Y|X}(y|x) \quad (16)$$

$$= \sum_{y \in \mathcal{Y}} p_{\vec{G}|Y}(\vec{g}|y) \text{tr}[E_y \rho_B^x], \quad (17)$$

where we have used the Markov property for the first equality and (10) for the second equality.

Let $\tilde{E}_{\vec{g}} := \sum_{y \in \mathcal{Y}} p_{\vec{G}|Y}(\vec{g}|y) E_y$. Note $\{\tilde{E}_{\vec{g}}\}_{\vec{g} \in \mathcal{X}^K}$ is a POVM: each element is positive semi-definite since $\{E_y\}_{y \in \mathcal{Y}}$ is a POVM, and

$$\sum_{\vec{g} \in \mathcal{X}^K} \tilde{E}_{\vec{g}} = \sum_{\vec{g} \in \mathcal{X}^K} \sum_{y \in \mathcal{Y}} p_{\vec{G}|Y}(\vec{g}|y) E_y \quad (18)$$

$$= \sum_{y \in \mathcal{Y}} \sum_{\vec{g} \in \mathcal{X}^K} p_{\vec{G}|Y}(\vec{g}|y) E_y = \sum_{y \in \mathcal{Y}} E_y = \mathbf{1}_B, \quad (19)$$

using again that $\{E_y\}_{y \in \mathcal{Y}}$ is a POVM. Then substituting the definition of $\tilde{E}_{\vec{g}}$ into (16) yields

$$p_{\vec{G}|X}(\vec{g}|x) = \text{tr}[\tilde{E}_{\vec{g}} \rho_B^x] \quad (20)$$

and hence (13) is satisfied with $E = \tilde{E}$. Therefore,

$$\mathcal{S}^{\text{Ordered}}(\rho_{XB}) = \mathcal{S}^{\text{Measured}}(\rho_{XB}). \quad (21)$$

- 3) Sequential quantum strategy: Suppose that Alice chooses x (which occurs with probability $p_X(x)$), and hence Bob has the state ρ_B^x . To make his first guess, Bob chooses a set of generalized measurement operators $\{M_x^{(1)}\}_{x \in \mathcal{X}}$ and reports the measurement outcome as his guess. He gets outcome x_1 with probability

$$p_{G_1|X}(x_1|x) = \text{tr}[M_{x_1}^{(1)} \rho_B^x M_{x_1}^{(1)\dagger}] \quad (22)$$

and his post-measurement state is

$$\frac{1}{p_{G_1|X}(x_1|x)} M_{x_1}^{(1)} \rho_B^x M_{x_1}^{(1)\dagger}. \quad (23)$$

Note: in general, Bob could perform a unitary operation U_1 on his state before measuring it. However, this would simply correspond to measuring with $\{M_x^{(1)} U_1\}_{x \in \mathcal{X}}$ instead. Hence, it suffices to simply consider a generalized measurement $\{M_x^{(1)}\}_{x \in \mathcal{X}}$.

Then, after learning the outcome x_1 , Bob chooses a new set of generalized measurement operators $\{M_x^{(2|x_1)}\}_{x \in \mathcal{X}}$. Note that this set of measurement operators can depend on x_1 . Without loss of generality, we can keep the same outcome set $\mathcal{X}$, since Bob could set, e.g. $M_{x_1}^{(2|x_1)} = 0$ to avoid guessing the same number twice. Bob measures his state and gets the outcome x_2 with probability

$$p_{G_2|G_1 X}(x_2|x_1, x) \\ = \frac{1}{p_{G_1|X}(x_1|x)} \text{tr}[M_{x_2}^{(2|x_1)} M_{x_1}^{(1)} \rho_B^x M_{x_1}^{(1)\dagger} M_{x_2}^{(2|x_1)\dagger}]. \quad (24)$$

Multiplying by $p_{G_1|X}(x_1|x)$ we see the joint distribution is given by

$$p_{G_1 G_2|X}(x_1, x_2|x) \\ = \text{tr}[M_{x_2}^{(2|x_1)} M_{x_1}^{(1)} \rho_B^x M_{x_1}^{(1)\dagger} M_{x_2}^{(2|x_1)\dagger}]. \quad (25)$$

To make his j th guess, we allow Bob to choose a new set of generalized measurement operators

$\{M_x^{(j|x_1, \dots, x_{j-1})}\}_{x \in \mathcal{X}}$, which may depend on the previous $j-1$ outcomes. Repeating the previous logic, in the end we find that

$$p_{G_1 G_2 \dots G_K | X}(x_1, x_2, \dots, x_K | x) = \text{tr}[M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})} \dots M_{x_2}^{(2|x_1)} M_{x_1}^{(1)} \rho_B^x] \quad (26)$$

Under such a strategy, the possible random variables giving the number of guesses is given by

$$\mathcal{S}^{\text{Sequential}}(\rho_{XB}) := \left\{ N(\vec{G}, X) : (\vec{G}, X) \text{ satisfy (26) for some collections of measurement operators } \{M_{x_j}^{(j|x_1, x_2, \dots, x_{j-1})}\}_{x_j \in \mathcal{X}}, j = 1, \dots, K, x_1, x_2, \dots, x_K \in \mathcal{X} \right\}. \quad (27)$$

Theorem 1: Let ρ_{XB} be a c-q state as defined in (4) and K a natural number with $K \leq |\mathcal{X}|$. Then

$$\mathcal{S}_K^{\text{Sequential}}(\rho_{XB}) = \mathcal{S}_K^{\text{Ordered}}(\rho_{XB}) = \mathcal{S}_K^{\text{Measured}}(\rho_{XB}). \quad (28)$$

We see that all three sets of random variables of the number of guesses obtained from various classes of strategies all coincide. Hence, we call the single class that of *quantum strategies*, denoted as $\mathcal{S}_K^{\text{Quantum}}(\rho_{XB})$.

Proof: The second equality was already stated in (21) and proven before that, and so it remains to prove the first equality. Consider a sequential strategy, with the notation of point 3 above. Define

$$E_{x_1, \dots, x_K} := M_{x_1}^{(1)\dagger} M_{x_2}^{(2|x_1)\dagger} \dots M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})\dagger} M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})} \dots M_{x_2}^{(2|x_1)} M_{x_1}^{(1)}. \quad (29)$$

We see that $E_{x_1, \dots, x_K} = A^\dagger A$ for $A = M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})} \dots M_{x_2}^{(2|x_1)} M_{x_1}^{(1)}$, and hence it is positive semi-definite. Moreover,

$$\sum_{x_1, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} = I_B \quad (30)$$

as can be seen by first summing (29) over x_K , using

$$\sum_{x_K \in \mathcal{X}} M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})\dagger} M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})} = I_B \quad (31)$$

since $\{M_x^{(K|x_1, x_2, \dots, x_{K-1})}\}_{x \in \mathcal{X}}$ is a set of generalized measurement operators, and then similarly summing over x_{K-1} , x_{K-2} , ..., and finally x_1 . Let us write $E_{\vec{x}}$ where $\vec{x} = (x_1, \dots, x_K)$ for $E_{x_1, \dots, x_K}$. We have shown that $\{E_{\vec{x}}\}_{\vec{x} \in \mathcal{X}^K}$ is a POVM. Moreover,

$$p_{G_1 G_2 \dots G_K | X}(x_1, x_2, \dots, x_K | x) = \text{tr}[E_{x_1, \dots, x_K} \rho_B^x]. \quad (32)$$

Hence, Bob's strategy is equivalent to simply performing the single POVM $\{E_{\vec{x}}\}_{\vec{x} \in \mathcal{X}^K}$ once, obtaining an outcome $\vec{x} = (x_1, \dots, x_K)$, and then making x_1 his first guess, x_2 his second guess, and so forth. That is, any such strategy can be recast as an ordered strategy.

On the other hand, any such ordered strategy can be reformulated as an adaptive strategy, by the following recursive

approach. Suppose that we are given $\{E_{\vec{y}}\}_{\vec{y} \in \mathcal{X}^K}$. For each $x_1 \in \mathcal{X}$, define

$$M_{x_1}^{(1)} = \sqrt{\sum_{x_2, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K}} \quad (33)$$

where we have chosen the positive semi-definite square root. We have that

$$\sum_{x_1 \in \mathcal{X}} M_{x_1}^{(1)\dagger} M_{x_1}^{(1)} = \sum_{x_1 \in \mathcal{X}} (M_{x_1}^{(1)})^2 \quad (34)$$

$$= \sum_{x_1 \in \mathcal{X}} \sum_{x_2, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} = I_B, \quad (35)$$

and so $\{M_{x_1}^{(1)}\}_{x_1 \in \mathcal{X}}$ forms a set of generalized measurement operators with outcomes in $\mathcal{X}$. Next, for each $x_1 \in \mathcal{X}$, corresponding to obtaining outcome x_1 on the first measurement, we define measurement operators $\{M_{x_2}^{(2|x_1)}\}_{x_2 \in \mathcal{X}}$ by

$$M_{x_2}^{(2|x_1)} = \sqrt{(M_{x_1}^{(1)})^{-1} \sum_{x_3, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} (M_{x_1}^{(1)})^{-1}}. \quad (36)$$

Then

$$\sum_{x_2 \in \mathcal{X}} (M_{x_2}^{(2|x_1)})^2 \quad (37)$$

$$= (M_{x_1}^{(1)})^{-1} \sum_{x_2 \in \mathcal{X}} \sum_{x_3, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} (M_{x_1}^{(1)})^{-1} \quad (38)$$

$$= (M_{x_1}^{(1)})^{-1} (M_{x_1}^{(1)})^2 (M_{x_1}^{(1)})^{-1} \quad (39)$$

$$= I_B. \quad (40)$$

Likewise, we define

$$M_{x_3}^{(3|x_1, x_2)} = \left\{ (M_{x_2}^{(2|x_1)})^{-1} (M_{x_1}^{(1)})^{-1} \sum_{x_4, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} (M_{x_1}^{(1)})^{-1} (M_{x_2}^{(2|x_1)})^{-1} \right\}^{1/2}. \quad (41)$$

Then

$$\sum_{x_3 \in \mathcal{X}} (M_{x_3}^{(3|x_1, x_2)})^2 \quad (42)$$

$$= (M_{x_2}^{(2|x_1)})^{-1} (M_{x_1}^{(1)})^{-1} \left(\sum_{x_3 \in \mathcal{X}} \sum_{x_4, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} \right) (M_{x_1}^{(1)})^{-1} (M_{x_2}^{(2|x_1)})^{-1} \quad (43)$$

$$= (M_{x_2}^{(2|x_1)})^{-1} (M_{x_2}^{(2|x_1)})^2 (M_{x_2}^{(2|x_1)})^{-1} \quad (44)$$

$$= I_B. \quad (45)$$

Repeating this process, we define

$$M_{x_\ell}^{(\ell|x_1, x_2, \dots, x_{\ell-1})} = \left\{ (M_{x_{\ell-1}}^{(\ell-1|x_1, x_2, \dots, x_{\ell-2})})^{-1} \dots (M_{x_1}^{(1)})^{-1} \sum_{x_{\ell+1}, \dots, x_K \in \mathcal{X}} E_{x_1, \dots, x_K} (M_{x_1}^{(1)})^{-1} \dots M_{x_{\ell-1}}^{(\ell-1|x_1, x_2, \dots, x_{\ell-2})})^{-1} \right\}^{1/2} \quad (46)$$

to obtain a generalized measurement operator for step ℓ (to use when having obtained outcomes $x_1, \dots, x_{\ell-1}$ during the previous steps). At the last step, $\ell = K$, there is no sum, namely

$$M_{x_K}^{(K|x_1, x_2, \dots, x_{K-1})} = \left\{ (M_{x_{K-1}}^{(K-1|x_1, x_2, \dots, x_{K-2})})^{-1} \dots (M_{x_1}^{(1)})^{-1} E_{x_1, \dots, x_K} (M_{x_1}^{(1)})^{-1} \dots M_{x_{K-1}}^{(K-1|x_1, x_2, \dots, x_{K-2})})^{-1} \right\}^{1/2}. \quad (47)$$

Lastly, we check that by design, (29) holds. Thus, we can work backwards from that equation and see that our newly created adaptive strategy yields the same outcomes with the same probabilities as the initial ordered strategy. ■

C. Success Metrics

Given a random variable X and a maximal number K of allowed guesses, how do we measure the success of a guessing strategy $\vec{G}$? We will focus on expectations of $N(\vec{G}, X)$. In particular, we consider the expected number of guesses required to guess correctly:

$$\mathbb{E}[N(\vec{G}, X)] = \begin{cases} \sum_{k=1}^K k \cdot p_{N(\vec{G}, X)}(k) & \text{if } p_{N(\vec{G}, X)}(\infty) = 0 \\ \infty & \text{if } p_{N(\vec{G}, X)}(\infty) > 0. \end{cases} \quad (48)$$

Here, $p_{N(\vec{G}, X)}(k)$ is the probability that, for guessing strategy $\vec{G}$, the k th guess is correct. We also consider a general cost vector $\vec{c} = \{c_1, c_2, \dots, c_{|\mathcal{X}|}\} \in (\mathbb{R} \cup \{\infty\})^{|\mathcal{X}|}$ with

$$0 \leq c_1 \leq c_2 \leq \dots \leq c_{|\mathcal{X}|}. \quad (49)$$

Then we define the modified expectation

$$E_{\vec{c}}(N(\vec{G}, X)) := \sum_{k=1}^{|\mathcal{X}|} c_k \cdot p_{N(\vec{G}, X)}(k).$$

Imposing a maximal number $K < |\mathcal{X}|$ of allowed guesses is equivalent to choosing $c_{K+1} = \dots = c_{|\mathcal{X}|} = \infty$, using the convention $\infty \cdot 0 = 0$. Accordingly, we implicitly associate K with $\vec{c}$ in all the following via the rule that $K = |\mathcal{X}|$ if and only if $c_{|\mathcal{X}|} < \infty$, and otherwise $K = \min\{i : c_i = \infty\}$. The case $K = |\mathcal{X}|$ therefore corresponds to $|\mathcal{X}|$ guesses being allowed, each with finite cost, and the case $K < |\mathcal{X}|$ corresponds to a limited number of allowed guesses, with a corresponding infinite cost if the correct answer is not obtained in K guesses.

Given a c-q state ρ_{XB} and a cost vector $\vec{c}$ as in (49), we define the generalized guesswork with quantum side information as

$$G_{\vec{c}}(X|B)_{\rho} := \inf_{N \in \mathcal{S}_K^{\text{Quantum}}(\rho_{XB})} E_{\vec{c}}(N). \quad (50)$$

Likewise, given a joint distribution p_{XY} , let

$$G_{\vec{c}}(X|Y)_p := \inf_{N \in \mathcal{S}_K^{\text{Classical}}(p_{XY})} E_{\vec{c}}(N). \quad (51)$$

From the equality

$$\mathcal{S}_K^{\text{Quantum}}(\rho_{XB}, K) = \mathcal{S}_K^{\text{Measured}}(\rho_{XB}) \quad (52)$$

of Theorem 1 it follows that

$$G_{\vec{c}}(X|B)_{\rho} = \inf_{\{E_y\}_{y \in \mathcal{Y}}} G_{\vec{c}}(X|Y)_p \quad (53)$$

where the infimum is over all finite alphabets $\mathcal{Y}$ and POVMs $\{E_y\}_{y \in \mathcal{Y}}$ and $p_{XY}(x, y) = p_X(x) \text{tr}[E_y \rho_B^x]$.

In the standard case in which $\vec{c} = (1, 2, \dots, |\mathcal{X}|)$, we define the *guesswork with quantum side information* as

$$G(X|B) \equiv G(X|B)_{\rho} := G_{\vec{c}}(X|B)_{\rho} \quad (54)$$

and likewise define $G(X|Y)_p = G_{\vec{c}}(X|Y)_p$ in the case of classical side information Y .

Remark 2: In Ref. [13], guesswork with quantum side information was defined by the right-hand side of (53) (with $\vec{c} = (1, 2, \dots, |\mathcal{X}|)$). Moreover, Proposition 1 of that work shows that the infimum in (53) in that case may be restricted to POVMs whose elements are all rank one.

IV. ENTROPIC BOUNDS

In this section, we use the results of Section III to obtain one-shot and asymptotic entropic bounds on $G(X|B)$ in terms of measured versions of bounds known in the classical case.

A. One-Shot Bounds

In the case in which $K = |\mathcal{X}|$, Arikan [4] showed that

$$\frac{1}{1 + \ln |\mathcal{X}|} \exp(H_{\frac{1}{2}}^{\dagger}(X|Y)_p) \leq G(X|Y)_p \leq \exp(H_{\frac{1}{2}}^{\dagger}(X|Y)_p) \quad (55)$$

where $H_{\alpha}^{\dagger}(X|Y)_p$ for $\alpha \in (0, 1) \cup (1, \infty)$ denotes the following α -conditional entropy of a joint distribution p_{XY} given by

$$H_{\alpha}^{\dagger}(X|Y) = \frac{\alpha}{1 - \alpha} \ln \left(\sum_{y \in \mathcal{Y}} \left(\sum_{x \in \mathcal{X}} p_{XY}(x, y)^{\alpha} \right)^{1/\alpha} \right) \quad (56)$$

$$= \sup_{q_Y} [-D_{\alpha}(p_{XY} \| \mathbf{1}_X \otimes q_Y)] \quad (57)$$

where the supremum is over probability distributions q_Y on $\mathcal{Y}$, and D_{α} is the α -Rényi relative entropy,

$$D_{\alpha}(p_X \| q_X) = \frac{1}{\alpha - 1} \ln \left(\sum_x p_X(x)^{\alpha} q_X(x)^{1-\alpha} \right). \quad (58)$$

The second equality of (56) follows from [15, Theorem 4]. Arikan's bound (55) applies to each $G_{\vec{c}}(p_{XY}, K)$ in (53), and hence by minimizing over the POVMs $\{E_y\}_{y \in \mathcal{Y}}$, we obtain

$$\frac{1}{1 + \ln |\mathcal{X}|} \exp(H_{\frac{1}{2}}^{\dagger, M}(X|B)_{\rho}) \leq G(X|B)_{\rho} \quad (59)$$

$$\leq \exp(H_{\frac{1}{2}}^{\dagger, M}(X|B)_{\rho}), \quad (60)$$

where for $\alpha \in (0, 1) \cup (1, \infty)$, $H_{\alpha}^{\dagger, M}(X|B)_{\rho}$ is the B -measured conditional α -Rényi entropy, defined by

$$H_{\alpha}^{\dagger, M}(X|B)_{\rho} := \inf_{\{E_y\}_{y \in \mathcal{Y}}} H_{\alpha}^{\dagger}(X|Y)_p, \quad (61)$$

where $p_{XY}(x, y) = p_X(x) \text{tr}[E_y \rho_B^x]$ is the joint probability distribution obtained by measuring the B part of ρ_{XB} via $\{E_y\}_{y \in \mathcal{Y}}$.

Remark 3: We may expand this quantity as

$$H_{\alpha}^{\uparrow, M}(X|B)_{\rho} = \inf_{\{E_y\}_{y \in \mathcal{Y}}} \sup_{q_Y} [-D_{\alpha}(p_{XY} \| \mathbf{1}_X \otimes q_Y)], \quad (62)$$

where p_{XY} is induced by the measurement of ρ_{XB} . This quantity seems to be different from the *conditional entropy induced by the measured Rényi divergence*, namely

$$H_{D_{\alpha}^M}^{\uparrow}(X|B)_{\rho} := \sup_{\sigma_B} -D_{\alpha}^M(\rho_{XB} \| \mathbf{1}_X \otimes \sigma_B), \quad (63)$$

where the supremum is over states on the B system, and for any pair of states (ρ, σ) ,

$$D_{\alpha}^M(\rho \| \sigma) := \sup_{\{E_z\}_z} D_{\alpha}(\{\text{tr}[E_z \rho]\}_z \| \{\text{tr}[E_z \sigma]\}_z) \quad (64)$$

is the *measured α -Rényi divergence*. Indeed, the latter quantity may be expanded to obtain

$$\begin{aligned} H_{D_{\alpha}^M}^{\uparrow}(X|B)_{\rho} \\ = \sup_{\sigma_B} \inf_{\{E_z\}_z} [-D_{\alpha}(\{\text{tr}[E_z \rho_{XB}]\}_z \| \{\text{tr}[E_z \mathbf{1}_X \otimes \sigma_B]\}_z)]. \end{aligned} \quad (65)$$

From the min-max inequality, and the fact that collective measurements on XB can simulate measurements on B alone, we have

$$H_{D_{\alpha}^M}^{\uparrow}(X|B)_{\rho} \leq H_{\alpha}^{\uparrow, M}(X|B)_{\rho}. \quad (66)$$

B. Asymptotic Analysis

We can consider the asymptotic setting in which Bob receives a sequence of product states $\rho_B^{\vec{x}} := \rho_B^{x_1} \otimes \cdots \otimes \rho_B^{x_n}$, with probability $p_X(x_1) \cdots p_X(x_n)$ and aims to guess the full sequence $\vec{x} = (x_1, \dots, x_n)$. In this case, the problem is characterized by the c-q state $\rho_{XB}^{\otimes n}$. The 1-shot bounds (59) give us

$$-\frac{1}{n} \ln(1 + n \ln(|\mathcal{X}|)) + \frac{1}{n} H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}} \quad (67)$$

$$\leq \frac{1}{n} \ln G(X^n|B^n)_{\rho^{\otimes n}} \quad (68)$$

$$\leq \frac{1}{n} H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}} \quad (69)$$

where $H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}}$ can involve collective measurements on the system B^n . Taking $n \rightarrow \infty$, we obtain

$$\lim_{n \rightarrow \infty} \frac{1}{n} \ln G(X^n|B^n)_{\rho^{\otimes n}} = \lim_{n \rightarrow \infty} \frac{1}{n} H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}}, \quad (70)$$

assuming that the limit on the right-hand side exists.

Note that we can bound

$$\frac{1}{n} H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}} \leq \frac{1}{n} \inf_{\{E_y\}_{y \in \mathcal{Y}}} H_{\frac{1}{2}}^{\uparrow}(X^n|Y^n)_{\rho^{\otimes n}} \quad (71)$$

$$= \inf_{\{E_y\}_{y \in \mathcal{Y}}} H_{\frac{1}{2}}^{\uparrow}(X|Y)_p \quad (72)$$

$$= H_{\frac{1}{2}}^{\uparrow, M}(X|B)_{\rho} \quad (73)$$

where the first inequality follows from the fact that product measurements are a special case of collective measurements, and the first equality follows from the additivity of the classical Rényi entropy ([4, Proposition 1]), and the third by the definition of $H_{\frac{1}{2}}^{\uparrow, M}(X|B)_{\rho}$. Moreover, by the data-processing inequality [16],

$$\frac{1}{n} H_{\frac{1}{2}}^{\uparrow, M}(X^n|B^n)_{\rho^{\otimes n}} \geq \frac{1}{n} \tilde{H}_{\frac{1}{2}}^{\uparrow}(X^n|B^n)_{\rho^{\otimes n}} = \tilde{H}_{\frac{1}{2}}^{\uparrow}(X|B)_{\rho}, \quad (74)$$

where the conditional Rényi entropy $\tilde{H}_{\alpha}^{\uparrow}(C|D)_{\sigma}$ of a bipartite state σ_{CD} is defined as

$$\tilde{H}_{\alpha}^{\uparrow}(C|D)_{\sigma} = \sup_{\omega_D} \left[-\tilde{D}_{\alpha}(\sigma_{CD} \| \mathbf{1}_C \otimes \omega_D) \right], \quad (75)$$

with the optimization with respect to states ω_D and the sandwiched Rényi relative entropy defined as [17], [18]:

$$\tilde{D}_{\alpha}(X\|Y) = \frac{1}{\alpha - 1} \ln \text{tr}[(Y^{(1-2\alpha)/\alpha} X Y^{(1-2\alpha)/\alpha})^{\alpha}]. \quad (76)$$

The equality in (74) follows from the additivity of $\tilde{H}_{\frac{1}{2}}^{\uparrow}$ under tensor products (see, e.g., [19, Corollary 5.2]). Hence, we obtain

$$\tilde{H}_{\frac{1}{2}}^{\uparrow}(X|B)_{\rho} \leq \lim_{n \rightarrow \infty} \frac{1}{n} \ln G(X^n|B^n)_{\rho^{\otimes n}} \leq H_{\frac{1}{2}}^{\uparrow, M}(X|B)_{\rho}. \quad (77)$$

In the classical case, (5), both the left and right-hand sides reduce to

$$H_{\frac{1}{2}}^{\uparrow}(X|Y)_p \quad (78)$$

where p is the underlying classical distribution of (5). Hence, these bounds recover Proposition 5 of [4].

V. SEMI-DEFINITE OPTIMIZATION REPRESENTATIONS AND THEIR CONSEQUENCES

The task of calculating $G_{\vec{c}}(X|B)_{\rho}$ as defined in (50) can be written as a semi-definite optimization problem, as was found in [13]. In this section, we present a different derivation of that fact yielding in (87) a representation dual to the one found in [13]. In Section VII-A we use this representation to prove that the guesswork $G(X|B)_{\rho}$ is a concave function of the c-q state ρ_{XB} . In Section VII-B we likewise use this representation to obtain a Lipschitz continuity bound on the guesswork. Then in Section V-A we compute the dual SDP, recovering the one obtained in [13]. In Section V-B we use this dual representation to develop a simple algorithm to obtain upper bounds on the quantity. Lastly, in Section V-C, we formulate a mixed-integer SDP representation of the problem, whose number of variables and constraints scales polynomially with all the relevant quantities (at the cost of adding binary constraints). We also provide implementations of these SDP representations [20], using the Julia programming language [21] and the optimization library Convex.jl [22].

Consider an ordered strategy $\vec{G}$ with a set of POVMs $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}^K}$. Then since $p_{\vec{G}, X}(\vec{g}, x) = p_X(x) \text{tr}[E_{\vec{g}} \rho_B^x]$, we have

$$c_k p_{N(\vec{G}, X)}(k) = c_k \sum_{x \in \mathcal{X}} p_X(x) \sum_{\substack{\vec{g} \in \mathcal{X}^K \\ N(\vec{g}, x) = k}} \text{tr}[E_{\vec{g}} \rho_B^x] \quad (79)$$

and hence

$$E_{\vec{c}}(N(\vec{G}, X)) = \sum_{k=1}^K c_k \sum_{x \in \mathcal{X}} p_X(x) \sum_{\substack{\vec{g} \in \mathcal{X}^K \\ N(\vec{g}, x)=k}} \text{tr}[E_{\vec{g}} \rho_B^x] \quad (80)$$

$$+ c_\infty \sum_{x \in \mathcal{X}} p_X(x) \sum_{\substack{\vec{g} \in \mathcal{X}^K \\ N(\vec{g}, x)=\infty}} \text{tr}[E_{\vec{g}} \rho_B^x] \quad (81)$$

$$= \sum_{\vec{g} \in \mathcal{X}^K} \sum_{x \in \mathcal{X}} c_{N(\vec{g}, x)} p_X(x) \text{tr}[E_{\vec{g}} \rho_B^x] \quad (82)$$

$$= \sum_{\vec{g} \in \mathcal{X}^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}] \quad (83)$$

where we define $R_{\vec{g}} := \sum_{x \in \mathcal{X}} p_X(x) c_{N(\vec{g}, x)} \rho_B^x$ for $\vec{g} \in \mathcal{X}^K$. Thus,

$$\begin{aligned} G_{\vec{c}}(X|B)_\rho &= \text{minimize} && \sum_{\vec{g} \in \mathcal{X}^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}] \\ &\text{subject to} && E_{\vec{g}} \geq 0 \quad \forall \vec{g} \in \mathcal{X}^K \\ &&& \sum_{\vec{g} \in \mathcal{X}^K} E_{\vec{g}} = \mathbf{1}_B. \end{aligned} \quad (84)$$

The expression in (84) clarifies that $R_{\vec{g}}$ has an interpretation as a cost operator corresponding to the guessing outcome $\vec{g}$. Since $\sum_{\vec{g} \in \mathcal{X}^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}]$ is linear in each positive semi-definite (matrix) variable $E_{\vec{g}}$, $G_{\vec{c}}(X|B)_\rho$ admits an SDP representation, given in (84). This program has $|\mathcal{X}|^K$ variables (each $d_B \times d_B$ complex positive semi-definite matrices), subject to one constraint. Note, however, since the cost vector $\vec{c}$ is increasing, any guess $\vec{h} \in \mathcal{X}^K$ with repeated elements is a suboptimal guessing order. That is, if $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}^K}$ is a POVM with $E_{\vec{h}} \neq 0$, and $\vec{h}' \in \mathcal{X}^K$ only differs from $\vec{h}$ by replacing repeated elements such that $\vec{h}'$ has no repeated elements, then the POVM defined by

$$\tilde{E}_{\vec{g}} := \begin{cases} E_{\vec{g}} & \vec{g} \neq \vec{h} \text{ and } \vec{g} \neq \vec{h}' \\ 0 & \vec{g} = \vec{h} \\ E_{\vec{h}} + E_{\vec{h}'} & \vec{g} = \vec{h}' \end{cases} \quad (85)$$

has $\sum_{\vec{g} \in \mathcal{X}^K} \text{tr}[R_{\vec{g}} \tilde{E}_{\vec{g}}] \leq \sum_{\vec{g} \in \mathcal{X}^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}]$. Hence, we may restrict to the outcome space

$$\mathcal{X}_\neq^K := \{\vec{g} \in \mathcal{X}^K : g_i \neq g_j, \forall i \neq j\} \subseteq \mathcal{X}^K. \quad (86)$$

Note $|\mathcal{X}_\neq^K| = \frac{|\mathcal{X}|!}{(|\mathcal{X}|-K)!}$, and in the case in which $K = |\mathcal{X}|$, the set $\mathcal{X}_\neq^K$ is just the set of permutations of $\mathcal{X}$. Hence, (84) can be re-written as the following smaller problem (with $\frac{|\mathcal{X}|!}{(|\mathcal{X}|-K)!}$ instead of $|\mathcal{X}|^K$ constraints):

$$\begin{aligned} G_{\vec{c}}(X|B)_\rho &= \text{minimize} && \sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}] \\ &\text{subject to} && E_{\vec{g}} \geq 0 \quad \forall \vec{g} \in \mathcal{X}_\neq^K \\ &&& \sum_{\vec{g} \in \mathcal{X}_\neq^K} E_{\vec{g}} = \mathbf{1}_B. \end{aligned} \quad (87)$$

Note that in the case $c_\infty = \infty$ and $K < |\mathcal{X}|$, there exists a finite solution if and only if there exists a POVM $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}$ such that for all $x \in \mathcal{X}$ and $\vec{g} \in \mathcal{X}_\neq^K$ with $x \notin \vec{g}$, we have $\text{tr}[E_{\vec{g}} \rho_B^x] = 0$. Whether or not this holds depends

on the particular state ρ_{XB} . However, when $c_\infty < \infty$ or $K = |\mathcal{X}|$, for any state ρ_{XB} , the problem (87) has a finite solution. Moreover, for any POVM $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}$, the objective $\sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[R_{\vec{g}} E_{\vec{g}}]$ is finite. In the following, we restrict to those two cases.

Remark 4: This optimization problem has the same form as that of discriminating quantum states in an ensemble, as described in, e.g., [23, Section 3.2.1]. Note, however, that (1) the $R_{\vec{g}}$ are positive semi-definite but not normalized, and (2) the case of having two copies of the unknown state in the guessing framework does not correspond to $R_{\vec{g}}^{\otimes 2}$. Nevertheless, slight modifications to [23, Theorem 3.9] show that a POVM $\{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}$ is optimal for (87) if and only if

$$Y = \sum_{\vec{g} \in \mathcal{X}_\neq^K} R_{\vec{g}} E_{\vec{g}} \quad (88)$$

satisfies $Y \leq R_{\vec{g}}$ for all $\vec{g} \in \mathcal{X}_\neq^K$.

Remark 5: The set of POVMs is convex and since the objective function is linear, any minimizer for (87) may be decomposed into extremal POVMs which are also minimizers. By [24, Corollary 2.2], any extremal POVM on a Hilbert space of size d_B has at most d_B^2 non-zero elements. Hence, there exist minimizers of (87) with at most d_B^2 non-zero elements (even though $|\mathcal{X}_\neq^K|$ could be far larger than d_B^2). Let $S \subseteq \mathcal{X}_\neq^K$ be a set of d_B^2 points such that there exists $\{\tilde{E}_{\vec{g}}\}_{\vec{g} \in S}$ with $\tilde{E}_{\vec{g}} \geq 0$, $\sum_{\vec{g} \in S} \tilde{E}_{\vec{g}} = \mathbf{1}_B$, and

$$G_{\vec{c}}(X|B)_\rho = \sum_{\vec{g} \in S} \text{tr}[\tilde{E}_{\vec{g}} R_{\vec{g}}]. \quad (89)$$

Then (87) holds with $\mathcal{X}_\neq^K$ replaced by S , namely

$$\begin{aligned} G_{\vec{c}}(X|B)_\rho &= \text{minimize} && \sum_{\vec{g} \in S} \text{tr}[R_{\vec{g}} E_{\vec{g}}] \\ &\text{subject to} && E_{\vec{g}} \geq 0 \quad \forall \vec{g} \in S \\ &&& \sum_{\vec{g} \in S} E_{\vec{g}} = \mathbf{1}_B. \end{aligned} \quad (90)$$

Note the “ $\leq$ ” direction of the equality (90) is trivial, since given a minimizer $\{E_{\vec{g}}\}_{\vec{g} \in S}$ for (90), simply extending it by choosing $E_{\vec{g}} = 0$ for $\vec{g} \notin S$ gives a feasible point for the optimization problem on the right-hand side of (87). The “ $\geq$ ” direction follows from the existence of the $\{\tilde{E}_{\vec{g}}\}_{\vec{g} \in S}$ described above. We note here that it is nontrivial to identify the set S . In general, identifying the set S is as difficult as solving the original problem in (87). However, applying a heuristic method inspired by choosing a smaller set of constraints can lead to useful upper bounds on the guesswork, which we describe in Section V-B.

A. The Dual Problem

Next, we compute the dual problem to (87), in the case $K = |\mathcal{X}|$ or $c_\infty < \infty$. Consider the Lagrangian

$$\mathcal{L}((E_{\vec{g}})_{\vec{g} \in \mathcal{X}_\neq^K}, (\lambda_{\vec{g}})_{\vec{g} \in \mathcal{X}_\neq^K}, \nu) = \sum_{\vec{g} \in \mathcal{X}_\neq^K} \langle R_{\vec{g}}, E_{\vec{g}} \rangle \quad (91)$$

$$- \sum_{\vec{g} \in \mathcal{X}_{\neq}^K} \langle \lambda_{\vec{g}}, E_{\vec{g}} \rangle + \left\langle \nu, \sum_{\vec{g} \in \mathcal{X}_{\neq}^K} E_{\vec{g}} - \mathbf{1}_B \right\rangle \quad (92)$$

$$= \sum_{\vec{g} \in \mathcal{X}_{\neq}^K} \langle R_{\vec{g}} - \lambda_{\vec{g}} + \nu, E_{\vec{g}} \rangle - \text{tr}[\nu] \quad (93)$$

where we have introduced the Hilbert–Schmidt product $\langle A, B \rangle = \text{tr}[A^\dagger B]$, and where $\lambda_{\vec{g}} \geq 0$ is the dual variable to the inequality constraint $E_{\vec{g}} \geq 0$, and $\nu = \nu^\dagger$ is the dual variable to the equality constraint $\sum_{\vec{g} \in \mathcal{X}_{\neq}^K} E_{\vec{g}} = \mathbf{1}_B$. As shown in, e.g., [25], the primal problem (87) can be expressed as

$$\min_{(E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}} \max_{\lambda_{\vec{g}} \geq 0, \nu} \mathcal{L}((E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, (\lambda_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, \nu) \quad (94)$$

while the dual problem is given by

$$\max_{\lambda_{\vec{g}} \geq 0, \nu} \min_{(E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}} \mathcal{L}((E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, (\lambda_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, \nu). \quad (95)$$

If $R_{\vec{g}} - \lambda_{\vec{g}} + \nu \neq 0$ for any $\vec{g} \in \mathcal{X}_{\neq}^K$, then the inner minimization in (95) yields $-\infty$. Hence,

$$\begin{aligned} \min_{(E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}} \mathcal{L}((E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, (\lambda_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}, \nu) \\ = \begin{cases} -\infty & R_{\vec{g}} - \lambda_{\vec{g}} + \nu \neq 0 \quad \exists \vec{g} \in \mathcal{X}_{\neq}^K \\ -\text{tr}[\nu] & \text{else.} \end{cases} \end{aligned} \quad (96)$$

The constraint $\lambda_{\vec{g}} \geq 0$ and $R_{\vec{g}} - \lambda_{\vec{g}} + \nu = 0$ imply the semi-definite inequality $-\nu \leq R_{\vec{g}}$. Writing $Y = -\nu$ and maximizing over $\lambda_{\vec{g}} \geq 0$, (95) becomes

$$\begin{aligned} \text{maximize} \quad & \text{tr}[Y] \\ \text{subject to} \quad & Y = Y^\dagger \\ & Y \leq R_{\vec{g}} \quad \forall \vec{g} \in \mathcal{X}_{\neq}^K \end{aligned} \quad (97)$$

Since (87) is strictly feasible (e.g., $E_{\vec{g}} = \mathbf{1}_B \frac{1}{|\mathcal{X}_{\neq}^K|}$ is a strictly feasible point) by Slater's condition, strong duality holds. Hence, (97) obtains the same optimal value as (87). The formulation of the problem as given in (97) was previously found in the work [13, Proposition 3]. In contrast to the primal SDP (90), the dual problem has a single variable Y subject to $|\mathcal{X}_{\neq}^K|$ constraints.

B. A Simple Algorithm to Compute Upper Bounds

The dual form of the SDP can be used to generate upper bounds on $G_{\vec{c}}(X|B)_\rho$ simply by removing constraints. This provides an algorithm to find an upper bound on the objective function: Decide on some number of constraints κ to impose in total. Then,

- 1) Initialize an empty list $L = \{\}$ corresponding to constraints to impose.
- 2) Set Y to be the identity matrix, as a first guess at the optimal dual variable.
- 3) If Y satisfies $Y \leq R_{\vec{g}}$ for all $\vec{g} \in \mathcal{X}_{\neq}^K$, then Y is the maximizer of the dual problem (97), and the optimization is solved. Otherwise, find $\vec{g} \in \mathcal{X}_{\neq}^K$ such that $Y \not\leq R_{\vec{g}}$, and add $\vec{g}$ to the list L .

4) Solve the problem

$$\begin{aligned} \text{maximize} \quad & \text{tr}[Y] \\ \text{subject to} \quad & Y = Y^\dagger \\ & Y \leq R_{\vec{g}} \quad \forall \vec{g} \in L \end{aligned} \quad (98)$$

and set Y to be its maximizer.

5) Repeat steps 2 and 3 until the list L has length κ .

6) Solve the problem one last time, and return the output.

In order to find a constraint that Y violates, a heuristic technique such as simulated annealing can be used. Moreover, in the case that there are too many constraints to fit into memory or check exhaustively, using an iterative technique (such as simulated annealing) is essential. If this algorithm was continued (without imposing a limit on the total number of constraints κ to impose), it would eventually yield the true value $G_{\vec{c}}(\rho_{XB}, K)$. When a total number of constraints is limited, it provides an upper bound (since it is a relaxation of (97)).

However, even with a limit κ on the total number of constraints, this algorithm can in theory yield the true value $G_{\vec{c}}(X|B)_\rho$. Note that the dual problem to (90) is

$$\begin{aligned} \text{maximize} \quad & \text{tr}[Y] \\ \text{subject to} \quad & Y = Y^\dagger \\ & Y \leq R_{\vec{g}} \quad \forall \vec{g} \in S \end{aligned} \quad (99)$$

where $S \subseteq \mathcal{X}_{\neq}^K$ has $|S| = d_B^2$ and is described in the remark above. Hence, if L in (98) equals S , then the algorithm finds the true value $G_{\vec{c}}(X|B)_\rho$, not just an upper bound. Thus, $\kappa = d_B^2$ suffices if the constraints $\vec{g}$ can be chosen precisely to obtain $L = S$. In general, finding S is as difficult as solving the original problem. Nonetheless, this motivates why choosing a relatively small value of κ (such as d_B^2) can still yield good upper bounds.

C. A Mixed-Integer Reformulation

The problem of computing $G_{\vec{c}}(X|B)_\rho$ can be formulated another way as a mixed-integer SDP, i.e., an SDP that has additional integer or binary constraints. Consider a POVM $\{F_j\}_{j=1}^M$ with M outcomes. When outcome j is obtained, Bob guesses in some order $\vec{g}^{(j)} \in \mathcal{X}_{\neq}^K$. Then consider the problem

$$\begin{aligned} \text{minimize} \quad & \sum_{x \in \mathcal{X}, j=1, \dots, M} p_X(x) c_{N(\vec{g}^{(j)}, x)} \text{tr}[F_j \rho_B^x] \\ \text{subject to} \quad & F_j \geq 0 \quad j = 1, \dots, M, \\ & \vec{g}^{(j)} \in \mathcal{X}_{\neq}^K, j = 1, \dots, M, \\ & \sum_{j=1}^M F_j = \mathbf{1}_B. \end{aligned} \quad (100)$$

We note that in the above, the variables to be optimized over are both the POVM $\{F_j\}_{j=1}^M$ and the guessing orders $\{\vec{g}^{(j)}\}_j$ corresponding to each POVM outcome. This optimization is not an SDP, since the dependence on the optimization variables $\{\vec{g}^{(j)}\}_{j=1}^M$ and $\{F_j\}$ is not linear, and $\vec{g}^{(j)} \in \mathcal{X}_{\neq}^K$ is a discrete constraint. Consider, however, the case that $K = |\mathcal{X}|$. With this assumption, we will be able to remove the

nonlinearity although not the discrete variables. This yields a *mixed-integer* SDP: an optimization problem such that if all integer constraints were removed, the result would be an SDP. We proceed as follows.

Under the condition $K = |\mathcal{X}|$, we may restrict to considering guessing orders that are permutations without loss of generality; other guessing orders have repeated guesses, which can only increase the value of the objective function. In this case, the outcome ∞ never occurs, and for each $\vec{g} \in S_{|\mathcal{X}|}$, the quantity $(c_{N(\vec{g},x)})_{x \in \mathcal{X}}$ satisfies

$$(c_{N(\vec{g},x)})_{x \in \mathcal{X}} = \vec{g}^{-1}(c), \quad (101)$$

where $\vec{g}^{-1}$ is the inverse permutation to $\vec{g}$, and $c = (c_k)_{k=1}^K$ is the cost vector (without ∞). Here, S_n is the set of permutations on $\{1, \dots, n\}$. Let $P^{(j)}$ be an $|\mathcal{X}| \times |\mathcal{X}|$ matrix representation of the permutation $\vec{g}^{(j)-1}$. Then $(P^{(j)}c)_x = \sum_{y \in \mathcal{X}} P_{xy}^{(j)} c_y = c_{N(\vec{g},x)}$. Hence, the optimization (100) can be reformulated as

$$\begin{aligned} & \text{minimize} && \sum_{x,y \in \mathcal{X}, j=1, \dots, M} p_X(x) P_{xy}^{(j)} c_y \operatorname{tr}[F_j \rho_B^x] \\ & \text{subject to} && F_j \in \mathcal{M}_{d_B} \quad \forall j \in [M], \\ & && P_{xy}^{(j)} \in \{0, 1\}, \quad \forall j \in [M], x, y \in \mathcal{X} \\ & && F_j \geq 0 \quad \forall j \in [M], \\ & && \sum_{j=1}^M F_j = \mathbf{1}_B, \\ & && \sum_{x \in \mathcal{X}} P_{xy}^{(j)} = 1, \quad \forall j \in [M], y \in \mathcal{X} \\ & && \sum_{y \in \mathcal{X}} P_{xy}^{(j)} = 1, \quad \forall j \in [M], x \in \mathcal{X} \end{aligned} \quad (102)$$

Note that all the constraints are semi-definite or linear, except that each element $P_{xy}^{(j)}$ is a binary variable: $P_{xy}^{(j)} \in \{0, 1\}$, which is a particularly simple type of discrete constraint. The non-linearity in the objective function, however, persists. To remove this, we take advantage of the fact that the $P_{xy}^{(j)}$ are binary. In particular, [26, Equations (22)–(24)] provide a clever trick to turn objective functions with terms of the form zx where z is a binary variable and x a continuous variable into objective functions of a continuous variable y subject to four affine constraints (in terms of x and z), as long as x is bounded by known constants. We reproduce this argument in the following.

We first write the objective function entirely in terms of scalar quantities:

$$\begin{aligned} & \sum_{x,y \in \mathcal{X}, j \in [M]} p_X(x) P_{xy}^{(j)} c_y \operatorname{tr}[F_j \rho_B^x] \\ &= \sum_{k, \ell \in [d_B]} \sum_{x,y \in \mathcal{X}, j \in [M]} p_X(x) (\rho_B^x)_{k\ell} c_y P_{xy}^{(j)} (F_j)_{\ell k} \end{aligned} \quad (103)$$

Let $x = (F_j)_{\ell k}$ and $z = P_{xy}^{(j)} \in \{0, 1\}$. Then $|x| \leq \operatorname{tr}[F_j]/2 \leq d_B/2$. Then $x_L := -d_B/2$ and $x_U := d_B/2$ constitute lower and upper bounds to x , respectively. Hence,

the following four inequalities hold trivially:

$$\begin{aligned} z(x - x_L) &\geq 0, \\ (z - 1)(x - x_U) &\geq 0, \\ z(x - x_U) &\leq 0, \\ (z - 1)(x - x_L) &\leq 0. \end{aligned} \quad (104)$$

Now, let $y = xz$. Then we have

$$\begin{aligned} y - zx_L &\geq 0, \\ y - zx_U &\geq x - x_U, \\ y - zx_U &\leq 0, \\ y - zx_L &\leq x - x_L. \end{aligned} \quad (105)$$

On the other hand, let us remove the constraint $y = xz$, and consider y as another variable. Then if $z = 0$, the first equation of (105) implies that $y \geq 0$, while the third implies $y \leq 0$, so $y = 0$. On the other hand, if $z = 1$, then the second equation of (105) implies that $y \geq x$ while the fourth implies that $y \leq x$. Hence, either way, $y = xz$. Thus, (105) is equivalent to $y = xz$.

With this transformation, (102) can be reformulated as the following.

$$\begin{aligned} & \text{minimize} && \sum_{k, \ell \in [d_B]} \sum_{x,y \in \mathcal{X}, j \in [M]} p_X(x) (\rho_B^x)_{k\ell} c_y y_{xy\ell k j} \\ & \text{subject to} && F_j \in \mathcal{M}_{d_B} \quad \forall j \in [M], \\ & && y_{xy\ell k j} \in \mathbb{R}, \quad \forall x, y \in \mathcal{X}, \ell, k \in [d_B], j \in [M], \\ & && P_{xy}^{(j)} \in \{0, 1\}, \quad \forall j \in [M], x, y \in \mathcal{X} \\ & && F_j \geq 0 \quad \forall j \in [M], \\ & && \sum_{j=1}^M F_j = \mathbf{1}_B, \\ & && \sum_{x \in \mathcal{X}} P_{xy}^{(j)} = 1 \quad \forall j \in [M], y \in \mathcal{X}, \\ & && \sum_{y \in \mathcal{X}} P_{xy}^{(j)} = 1 \quad \forall j \in [M], x \in \mathcal{X}, \\ & && y_{xy\ell k j} + P_{xy}^{(j)} \frac{d_B}{2} \geq 0, \\ & && y_{xy\ell k j} - P_{xy}^{(j)} \frac{d_B}{2} \geq (F_j)_{\ell k} - \frac{d_B}{2}, \\ & && y_{xy\ell k j} - P_{xy}^{(j)} \frac{d_B}{2} \leq 0, \\ & && y_{xy\ell k j} + P_{xy}^{(j)} \frac{d_B}{2} \leq (F_j)_{\ell k} + \frac{d_B}{2}. \end{aligned} \quad (106)$$

where the last four constraints hold for $\forall x, y \in \mathcal{X}, \ell, k \in [d_B]$, and $j \in [M]$. This is a mixed-integer SDP, with a number of constraints and variables that is polynomial in $M, d_B, |\mathcal{X}|$. Moreover, if $M \geq d_B^2$, then as follows from the remark below (87), the mixed-integer SDP (106) obtains the same optimal value as (87), namely $G_{\vec{c}}(\rho_{XB}, |\mathcal{X}|)$, using that $K = |\mathcal{X}|$. Note, however, that mixed-integer SDPs are not in general efficiently solvable; they encompass mixed integer linear programs, which are NP-hard. However, in practice they can sometimes be quickly solved. Since the original SDP formulation (87) involves an exponential (in $|\mathcal{X}|$) number of variables (or an exponential number of constraints in its dual

formulation (97)), Eq. (106) may provide a more practical approach in some cases because it instead has a polynomial (in $|\mathcal{X}|$) number of variables. Mixed-integer SDPs can be solved in various ways; in the code accompanying this paper [20], the problem (106) is solved using the library Pajarito.jl [27], which proceeds by solving an alternating sequence of mixed-integer linear problems and SDPs.

VI. THE ELLIPSOID ALGORITHM

The ellipsoid algorithm (see, e.g., [28]) provides a theoretical proof that under a strict feasibility assumption, semi-definite programs can be solved in time that scales as a polynomial in: the number of scalar variables and constraints, the logarithm of a 2-norm bound on the feasible points, $\ln(1/\varepsilon)$ where ε is the solution tolerance, and the maximum bit length of the scalar entries of the objective and constraints (see e.g. [29, Theorem 4]).

In fact, the ellipsoid algorithm applies quite generally to the optimization of a linear objective function over a convex *feasible region* (which could be described by a domain and constraint functions). The ellipsoid algorithm only requires a *separation oracle* for the feasible region, a subroutine which either asserts that a given point lies within the feasible region, or provides a separating hyperplane between the given point and the feasible region. When the separation oracle can be evaluated in polynomial time, the overall ellipsoid algorithm runs in polynomial time as well (see [28, Corollary 4.2.7]).

In the case of a single positive semi-definite constraint, e.g. $Y \geq 0$, a simple separation oracle is given by computing the eigendecomposition of Y and checking if all of its eigenvalues are non-negative. If so, it returns that Y is indeed feasible, and otherwise returns the matrix $C := U \text{diag}(f(\lambda_1), \dots, f(\lambda_d)) U^\dagger$ where $Y = U \text{diag}(\lambda_1, \dots, \lambda_d) U^\dagger$ is the eigendecomposition of Y , U is unitary, $\lambda_1, \dots, \lambda_d$ are the eigenvalues, and $f(x) = 1$ if $x < 0$ and $f(x) = 0$ otherwise. This matrix has the properties that $C \geq 0$, $\|C\|_\infty = 1$, and $\text{tr}[C^\dagger Y] = \sum_{i=1}^d f(\lambda_i) \lambda_i = \sum_{i: \lambda_i < 0} \lambda_i < 0$.

In the case of the dual problem (97) with $K = |\mathcal{X}|$, we have $|\mathcal{X}|!$ positive semi-definite constraints. Thus, we cannot check all of them together in polynomial time.

The feasibility problem $Y \leq R_\pi$ for each $\pi \in S_{|\mathcal{X}|}$ can be written as the following mixed-integer non-linear problem,

$$\begin{aligned} \eta := \text{minimize} \quad & \langle \psi, \left(\sum_{x \in \mathcal{X}} (Pc)_x p_X(x) \rho_B^x - Y \right) \psi \rangle \\ \text{subject to} \quad & \sum_{i \in \mathcal{X}} P_{ij} = \sum_{j \in \mathcal{X}} P_{ij} = 1 \\ & P_{ij} \in \{0, 1\}, \quad i, j \in \mathcal{X} \\ & \psi \in \mathbb{C}^{d_B} \end{aligned}$$

where $\eta \geq 0$ if and only if $Y \leq R_\pi$ for all $\pi \in S_{|\mathcal{X}|}$, using that a matrix M satisfies $M \geq 0$ if and only if $\langle \psi, M \psi \rangle \geq 0$ for all $\psi \in \mathbb{C}^{d_B}$. Since the convex hull of the permutation matrices is given by the doubly stochastic matrices, the discrete constraints can be relaxed, yielding the

following reformulation

$$\begin{aligned} \eta = \text{minimize} \quad & \langle \psi, \left(\sum_{x \in \mathcal{X}} (Dc)_x p_X(x) \rho_B^x - Y \right) \psi \rangle \\ \text{subject to} \quad & \sum_{i \in \mathcal{X}} D_{ij} = \sum_{j \in \mathcal{X}} D_{ij} = 1 \\ & D_{ij} \geq 0, \quad i, j \in \mathcal{X} \\ & \psi \in \mathbb{C}^{d_B}. \end{aligned} \quad (107)$$

If $\eta \geq 0$, then Y is feasible. Otherwise, the optimal value D^* can be decomposed as a convex combination of permutations, $D^* = \sum_i \alpha_i P_i$, and we must have $\sum_x (P_i c)_x p_X(x) \rho_B^x \not\leq Y$ for some P_i , using that the objective is an affine function of D . The problem (107) can be solved by global non-linear optimization solvers such as EAGO.jl [30] or SCIP [31], but not in general in polynomial time.

At each iteration of the ellipsoid algorithm, one must evaluate the separation oracle for some Hermitian matrix Y . In order to avoid solving (107), one may attempt to prove the feasibility or infeasibility of a point Y by other means. For example, one may search over permutations π heuristically, in order to find R_π such that $Y \not\leq R_\pi$. If such a permutation can be identified, then Y is not feasible, and the problem (107) does not need to be solved. Likewise, if one can show that for some $k \in (1, \dots, |\mathcal{X}|)$,

$$Y \leq \sum_{i=1}^k c_{|\mathcal{X}|-i} p_{x_i} \rho_B^{x_i} + c_1 \sum_{x \in \mathcal{X} \setminus \{x_{i_1}, \dots, x_{i_k}\}} p_x \rho_B^x \quad \forall (x_1, \dots, x_k) \in \mathcal{X}_{\neq}^k \quad (108)$$

then Y must be feasible, and again (107) does not need to be solved. The number of comparisons required scales as $|\mathcal{X}|^k$; for small choices of k , this provides an efficient check for feasibility (which may, however, be inconclusive).

A. Numerical Comparisons

We compare numerical implementations of several of the above algorithms on a set of 12 test problems. The code for these experiments can be found at [20]. Each problem has $p \equiv u$, the uniform distribution $u := (1/|\mathcal{X}|, \dots, 1/|\mathcal{X}|)$, for simplicity. The states are chosen as

- 1) Two random qubit density matrices
- 2) Two random qutrit density matrices
- 3) Three pure qubits chosen equidistant within one plane of the Bloch sphere (the qubit trine states), i.e.

$$\cos(j \frac{2\pi}{3}) |0\rangle + \sin(j \frac{2\pi}{3}) |1\rangle, \quad j = 1, 2, 3$$

- 4) Three random qubit density matrices
- 5) Three random qutrit density matrices
- 6) The four BB84 states, $|0\rangle, |1\rangle$, and $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$

as well as the “tensor-2” case of

$$\{\rho \otimes \sigma : \rho, \sigma \in S\} \quad (109)$$

for each of the six sets S listed above, corresponding to the guesswork problem with quantum side information associated to $\rho_{XB}^{\otimes 2}$, where ρ_{XB} is the state associated to the original

guesswork problem with quantum side information. The random states were chosen uniformly at random (i.e. according to the Haar measure).

The exponentially-large SDP formulation (and its dual), the mixed-integer SDP algorithm, and the active set method were compared, with several choices of parameters and underlying solvers. The mixed-integer SDP formulation was evaluated with $M = d_B$ (yielding an upper bound), $M = d_B^2$ (yielding the optimal value), with the Pajarito mixed-integer SDP solver [27], using Convex.jl (version 0.12.7) [22] to formulate the problem. Pajarito proceeds by solving mixed-integer linear problems (MILP) and SDPs as subproblems, and thus uses both a MILP solver and an SDP solver as subcomponents. Pajarito provides two algorithms: an iterative algorithm, which alternates between solving MILP and SDP subproblems, and solving a single branch-and-cut problem in which SDP subproblems are solved via so-called lazy callbacks to add cuts to the mixed-integer problem. The latter is called “mixed-solver drives” (MSD) in the Pajarito documentation. We tested three configurations of Pajarito (version 0.7.0):

- (c1) Gurobi (version 9.0.3) as the MILP solver and MOSEK (version 8.1.0.82) as the SDP solver, with Pajarito’s MSD algorithm,
- (c2) Gurobi as the MILP solver and MOSEK as the SDP solver, with Pajarito’s iterative algorithm, with a relative optimality gap tolerance of 0,
- (o) Cbc [32] (version 2.10.3) as the MILP solver, and SCS [33] (version 2.1.1) as the SDP solver, with Pajarito’s iterative algorithm.

Here, ‘c’ stands for commercial, and ‘o’ for open-source. In the configuration (c1), Gurobi was set to have a relative optimality gap tolerance of 10^{-5} and in (c2), a relative optimality gap tolerance of 0. In both configurations, Gurobi was given an absolute linear-constraint-wise feasibility tolerance of 10^{-8} , and an integrality tolerance of 10^{-9} . These choices of parameters match those made in [27]. Cbc was given an integrality tolerance of 10^{-8} , and SCS’s (normalized) primal, dual residual and relative gap were set to 10^{-6} for each problem. The default parameters were used otherwise. Note the MSD option was not used with Cbc, since the solver does not support lazy callbacks.

For the (exponentially large) SDP primal and dual formulations, the problems were solved with both MOSEK and SCS, and likewise with the active-set upper bound.

The active set method uses simulated annealing to iteratively add violated constraints to the problem to find an upper bound, as described in Section V-B, and uses a maximum-time parameter $t_{\max}$ to stop iterating when the estimated time of finding another constraint to add would cause the running time to exceed the maximum-time.¹ This provides a way to compare the improvement (or lack thereof) of running the algorithm for more iterations. The algorithm also terminates when a violated constraint cannot be found after 50 runs of simulated annealing (started each time with different random

initial conditions). Here, the problems were solved with three choices of $t_{\max}$, 20 s, 60 s, and 240 s.

The exact answer was not known analytically for most of these problems, and so the average relative error was calculated by comparing to the mean of the solutions (excluding the active-set method and the MISDP with $M = d_B$, which only give an upper bound in general). For the cases of the BB84 states and the Y-states, where the solution is known exactly (see Section VIII), the solutions obtained here match the analytic value to a relative tolerance of at least 10^{-7} .

The problems were run sequentially on a four-core desktop computer (Intel i7-6700K 4.00GHz CPU, with 16 GB of RAM, on Ubuntu-20.04 via Windows Subsystem for Linux, version 2), via the programming language Julia [21] (version 1.5.1), with a 5 minute time limit. The results are summarized in Table I, and presented in more detail in Table II and Table III.

One can see that the MISDP problems were harder to solve than the corresponding SDPs for these relatively small problem instances. The MISDPs have the advantage of finding extremal solutions, however, in the case $M = d_B$, and may scale better to large instances. Additionally, the active-set upper bound performed fairly well, finding feasible points within 20% of the optimum in all cases, with only $t_{\max} = 20$ s, and often finding near-optimal solutions. It was also the only method able to scale to the largest instances tested, such as two copies of the BB84 states (which involves 16 quantum states in dimension four, and for which the SDP formulation has $16!$ variables.). In general, the commercial solvers performed better than the open source solvers, with the notable exception of the active-set upper bound with MOSEK, in which two more problems timed out than with SCS. This could be due to SCS being a first-order solver which can therefore possibly scale to larger problem instances than MOSEK, which is a second-order solver.

VII. PROPERTIES OF GUESSWORK WITH QUANTUM SIDE INFORMATION

A. Concavity of the Guesswork

Proposition 6: For each cost vector $\vec{c}$ and $K \leq |\mathcal{X}|$, the function

$$\rho_{XB} \mapsto G_{\vec{c}}(X|B)_{\rho} \quad (110)$$

from the set of c-q states of the form (4) to $\mathbb{R}_{\geq 0} \cup \{\infty\}$, is concave.

Proof: For $\vec{g} \in \mathcal{X}_{\neq}^K$, and ρ_{XB} a c-q state, the quantity $R_{\vec{g}}^{\rho} := \sum_{x \in \mathcal{X}} p_X(x) c_{N(\vec{g}, x)} \rho_B^x$ can be expressed as

$$R_{\vec{g}}^{\rho} = \text{tr}_X \left[\left(\sum_{x \in \mathcal{X}} c_{N(\vec{g}, x)} |x\rangle\langle x|_X \otimes I_B \right) \rho_{XB} \right] \quad (111)$$

and hence is linear in ρ_{XB} . Then for each POVM $(E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}$,

$$\rho_{XB} \mapsto \sum_{\vec{g} \in \mathcal{X}_{\neq}^K} \text{tr}[R_{\vec{g}}^{\rho} E_{\vec{g}}] \quad (112)$$

¹The maximum time can still be exceeded, since at least one iteration must be performed and the estimate can be wrong.

TABLE I

COMPARISON OF AVERAGE RELATIVE ERROR AND AVERAGE SOLVE TIME FOR THE 12 PROBLEMS DISCUSSED ABOVE. A PROBLEM IS CONSIDERED “TIMED OUT” IF AN ANSWER IS NOT OBTAINED IN 5 MINUTES, AND “ERRORED OUT” IF THE SOLUTION WAS NOT OBTAINED DUE TO ERRORS (SUCH AS RUNNING OUT OF RAM). THE AVERAGE RELATIVE ERROR, WHICH WAS ROUNDED TO TWO DECIMAL DIGITS, AND THE TIME TAKEN ARE CALCULATED ONLY OVER THE PROBLEMS WHICH WERE SOLVED BY THE GIVEN ALGORITHM AND CHOICE OF PARAMETERS. “MISDP (d_B)” REFERS TO THE CHOICE $M = d_B$, AND LIKEWISE “MISDP (d_B^2)” REFERS TO THE CHOICE $M = d_B^2$

Algorithm	Parameters	average rel. error	average time	number solved	number out	timed out	number errored out
MISDP (d_B)	Pajarito (c1)	0 %	23.74 s	6	6		0
	Pajarito (c2)	0 %	24.03 s	6	6		0
	Pajarito (o)	0 %	45.05 s	6	6		0
MISDP (d_B^2)	Pajarito (c1)	0 %	35.27 s	5	7		0
	Pajarito (c2)	0 %	27.97 s	4	8		0
	Pajarito (o)	0 %	131.35 s	4	8		0
SDP	MOSEK	0 %	8.99 s	8	3		1
	SCS	0 %	9.08 s	8	3		1
SDP (dual)	MOSEK	0 %	8.74 s	8	3		1
	SCS	0 %	8.59 s	8	3		1
Active set upper bound (MOSEK)	$t_{\max} = 20$ s	6.80 %	16.08 s	10	2		0
	$t_{\max} = 60$ s	6.79 %	19.03 s	10	2		0
	$t_{\max} = 240$ s	6.80 %	26.17 s	10	2		0
Active set upper bound (SCS)	$t_{\max} = 20$ s	6.09 %	33.24 s	12	0		0
	$t_{\max} = 60$ s	6.09 %	35.78 s	12	0		0
	$t_{\max} = 240$ s	6.09 %	34.30 s	11	1		0

TABLE II

THE INDIVIDUAL TIMINGS FOR EACH ALGORITHM AND CHOICE OF SETTINGS ON PROBLEMS (1)–(3), AND THE CORRESPONDING “TENSOR-2” PROBLEMS DISCUSSED AT (109). FOR EACH ALGORITHM, THE RUNNING TIME OF THE ORIGINAL PROBLEM IS GIVEN FOLLOWED BY THE RUNNING TIME ON THE “TENSOR-2” PROBLEM, E.G. THE SDP FORMULATION WITH MOSEK ON THE TWO RANDOM QUBITS PROBLEM WAS SOLVED IN 8.69 SECONDS, AND IN 8.84 SECONDS FOR THE CORRESPONDING TENSOR-2 PROBLEM. “TIMEOUT” IS WRITTEN WHENEVER THE PROBLEM WAS NOT SOLVED WITHIN 5 MINUTES. FOR THE ACTIVE SET ALGORITHMS, THE RELATIVE ERROR IS ALSO GIVEN FOR EACH PROBLEM IN PARENTHESIS. NOTE THAT THE MISDP FORMULATION WITH $M = d_B$ IS ALSO ONLY KNOWN TO BE AN UPPER BOUND, BUT A RELATIVE ERROR OF LESS THAN 10^{-5} IN EACH INSTANCE, SO THE RELATIVE ERRORS ARE OMITTED. LASTLY, THE RELATIVE ERROR IS WRITTEN AS ? % IN THE CASE THAT ONLY AN UPPER BOUND WAS OBTAINED

Algorithm	Parameters	Two random qubits	Two random qutrits	Y-states
MISDP (d_B)	Pajarito (c1)	23.63 s, timeout	23.60 s, timeout	23.56 s, timeout
	Pajarito (c2)	22.99 s, timeout	23.31 s, timeout	23.21 s, timeout
	Pajarito (o)	23.47 s, timeout	24.77 s, timeout	26.15 s, timeout
MISDP (d_B^2)	Pajarito (c1)	24.49 s (0.00 %), timeout	31.40 s (0.00 %), timeout	26.27 s (0.00 %), timeout
	Pajarito (c2)	25.02 s (0.00 %), timeout	31.39 s (0.00 %), timeout	29.08 s (0.00 %), timeout
	Pajarito (o)	26.54 s (0.00 %), timeout	212.79 s (0.00 %), timeout	141.14 s (0.00 %), timeout
SDP	MOSEK	8.69 s, 8.84 s	8.78 s, 9.23 s	9.33 s, timeout
	SCS	9.00 s, 8.90 s	8.44 s, 11.22 s	8.98 s, timeout
SDP (dual)	MOSEK	8.46 s, 8.63 s	8.54 s, 8.83 s	9.11 s, timeout
	SCS	8.76 s, 8.32 s	8.33 s, 9.20 s	8.74 s, timeout
Active set upper bound (MOSEK)	$t_{\max} = 20$ s	8.76 s (19.5 %), 10.41 s (1.5 %)	8.89 s (19.5 %), timeout	9.72 s (0 %), 34.25 s (? %)
	$t_{\max} = 60$ s	10.91 s (19.5 %), 10.41 s (1.9 %)	8.87 s (19.5 %), timeout	9.66 s (0 %), 31.00 s (? %)
	$t_{\max} = 240$ s	9.47 s (19.5 %), 10.40 s (1.5 %)	8.90 s (19.5 %), timeout	9.81 s (0 %), 30.26 s (? %)
Active set upper bound (SCS)	$t_{\max} = 20$ s	9.04 s (19.5 %), 10.92 s (1.5 %)	8.70 s (19.5 %), 101.06 s (1.1 %)	9.23 s (0 %), 82.84 s (? %)
	$t_{\max} = 60$ s	9.07 s (19.5 %), 10.22 s (1.9 %)	8.66 s (19.5 %), 32.94 s (1.2 %)	9.18 s (0 %), 50.79 s (? %)
	$t_{\max} = 240$ s	9.04 s (19.5 %), 10.02 s (1.9 %)	8.79 s (19.5 %), 22.69 s (1.2 %)	9.31 s (0 %), 37.36 s (? %)

is linear in ρ_{XB} . The arbitrary infimum of concave functions, and in particular linear functions, is concave, and hence

$$G_{\mathcal{E}}(X|B)_{\rho} \equiv \min_{(E_{\vec{g}})_{\vec{g} \in \mathcal{X}_{\neq}^K}} \sum_{\vec{g} \in \mathcal{X}_{\neq}^K} \text{tr}[R_{\vec{g}}^{\rho} E_{\vec{g}}], \quad (113)$$

where the minimum is taken over all POVMs on system B with outcomes in $\mathcal{X}_{\neq}^K$, is concave. ■

Remark 7: Theorem 6 carries over to guesswork without side information, $G(X)$, which simply corresponds to the case that $\rho_B^x \equiv \rho_B$ is independent of $x \in \mathcal{X}$. Since $G(X)$

TABLE III

THE INDIVIDUAL TIMINGS FOR EACH ALGORITHM AND CHOICE OF SETTINGS ON PROBLEMS (4)–(6). SEE TABLE II FOR A DESCRIPTION OF THE QUANTITIES SHOWN. HERE, “ERROR” MEANS THE SOLUTION WAS NOT OBTAINED DUE TO AN ERROR (SUCH AS RUNNING OUT OF MEMORY)

Algorithm	Parameters	Three random qubits	Three random qutrits	BB84 states
MISDP (d_B)	Pajarito (c1)	23.63 s, timeout	24.49 s, timeout	23.51 s, timeout
	Pajarito (c2)	23.17 s, timeout	26.50 s, timeout	25.01 s, timeout
	Pajarito (o)	27.11 s, timeout	95.01 s, timeout	73.80 s, timeout
MISDP (d_B^2)	Pajarito (c1)	25.82 s (0.00 %), timeout	timeout, timeout	68.35 s (0.00 %), timeout
	Pajarito (c2)	26.38 s (0.00 %), timeout	timeout, timeout	timeout, timeout
	Pajarito (o)	144.93 s (0.00 %), timeout	timeout, timeout	timeout, timeout
SDP	MOSEK	9.35 s, timeout	8.82 s, timeout	8.87 s, error
	SCS	9.09 s, timeout	8.46 s, timeout	8.54 s, error
SDP (dual)	MOSEK	9.14 s, timeout	8.55 s, timeout	8.62 s, error
	SCS	8.86 s, timeout	8.25 s, timeout	8.23 s, error
Active set upper bound (MOSEK)	$t_{\max} = 20$ s	9.73 s (1.3 %), 32.29 s (? %)	9.50 s (5.8 %), timeout	9.51 s (0 %), 27.72 s (? %)
	$t_{\max} = 60$ s	9.69 s (1.3 %), 24.12 s (? %)	9.45 s (5.8 %), timeout	9.77 s (0 %), 66.47 s (? %)
	$t_{\max} = 240$ s	9.69 s (1.3 %), 30.42 s (? %)	9.45 s (5.8 %), timeout	9.49 s (0 %), 133.81 s (? %)
Active set upper bound (SCS)	$t_{\max} = 20$ s	9.44 s (1.3 %), 35.66 s (? %)	9.67 s (5.8 %), 84.43 s (? %)	9.09 s (0 %), 28.76 s (? %)
	$t_{\max} = 60$ s	9.44 s (1.3 %), 54.60 s (? %)	9.11 s (5.8 %), 155.51 s (? %)	9.29 s (0 %), 70.57 s (? %)
	$t_{\max} = 240$ s	8.89 s (1.3 %), 75.43 s (? %)	9.01 s (5.8 %), timeout	9.15 s (0 %), 177.64 s (? %)

is manifestly symmetric under permutations of the density p_X , this proves that $G(X)$ is a *Schur concave* function of the distribution p_X (i.e., decreasing in the majorization pre-order; see, e.g., [34] for an overview of majorization and Schur concave functions). Consequently, the work [35] provides an algorithm to calculate local continuity bounds for $G(X)$.

B. Continuity of the Guesswork

Proposition 8: For each cost vector $\vec{c}$ and $K \leq |\mathcal{X}|$, such that either $c_\infty < \infty$ or $K = |\mathcal{X}|$, the function

$$\rho_{XB} \mapsto G_{\vec{c}}(X|B)_\rho \quad (114)$$

from the set of c-q states of the form (4) to $\mathbb{R}_{\geq 0}$, is Lipschitz continuous, satisfying the bound

$$|G_{\vec{c}}(X|B)_\rho - G_{\vec{c}}(X|B)_\sigma| \leq \kappa \|\rho_{XB} - \sigma_{XB}\|_1. \quad (115)$$

for any c-q states ρ_{XB} and σ_{XB} , where $\kappa = c_\infty$ if $K < |\mathcal{X}|$, and $\kappa = c_{|\mathcal{X}|}$ if $K = |\mathcal{X}|$.

Proof: Define

$$f(\rho_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) := \sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[R_{\vec{g}}^\rho E_{\vec{g}}]. \quad (116)$$

Then, by linearity (as discussed in the proof of Theorem 6),

$$\begin{aligned} f(\rho_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) - f(\sigma_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) \\ = \sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[R_{\vec{g}}^{\rho-\sigma} E_{\vec{g}}] \end{aligned} \quad (117)$$

$$= \sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[\text{tr}_X[C_{XB}^{(\vec{g})} \Delta_{XB}] E_{\vec{g}}] \quad (118)$$

using (111), where $C_{XB}^{(\vec{g})} := \sum_{x \in \mathcal{X}} c_{N(\vec{g}, x)} |x\rangle\langle x| \otimes I_B \geq 0$ and $\Delta_{XB} := \rho_{XB} - \sigma_{XB}$. Since $C_{XB}^{(\vec{g})}$ and Δ_{XB} commute,

using the c-q structure of each, we have

$$\begin{aligned} f(\rho_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) - f(\sigma_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) \\ = \sum_{\vec{g} \in \mathcal{X}_\neq^K} \text{tr}[C_{XB}^{(\vec{g})} \Delta_{XB} (I_X \otimes E_{\vec{g}})] \end{aligned} \quad (119)$$

$$= \text{tr} \left[\Delta_{XB} \sum_{\vec{g} \in \mathcal{X}_\neq^K} C_{XB}^{(\vec{g})} (I_X \otimes E_{\vec{g}}) \right]. \quad (120)$$

Set

$$F_{XB} := \sum_{\vec{g} \in \mathcal{X}_\neq^K} C_{XB}^{(\vec{g})} (I_X \otimes E_{\vec{g}}) = \sum_{x \in \mathcal{X}} \sum_{\vec{g} \in \mathcal{X}_\neq^K} c_{N(\vec{g}, x)} |x\rangle\langle x| \otimes E_{\vec{g}}. \quad (121)$$

Since $c_{N(\vec{g}, x)} \leq \kappa$ for each $x \in \mathcal{X}$ and $\vec{g} \in \mathcal{X}_\neq^K$, we have that $F_{XB} \leq \kappa \sum_{x \in \mathcal{X}} \sum_{\vec{g} \in \mathcal{X}_\neq^K} |x\rangle\langle x| \otimes E_{\vec{g}}$ in semi-definite order. Performing the sums, we have $F_{XB} \leq \kappa I_X \otimes I_B$ and hence $\|F_{XB}\|_\infty \leq \kappa$. Thus,

$$\begin{aligned} f(\rho_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) - f(\sigma_{XB}, \{E_{\vec{g}}\}_{\vec{g} \in \mathcal{X}_\neq^K}) \\ = \text{tr}[\Delta_{XB} F_{XB}] \end{aligned} \quad (122)$$

$$\leq \|\Delta_{XB} F_{XB}\|_1 \quad (123)$$

$$\leq \|\Delta_{XB}\|_1 \|F_{XB}\|_\infty \quad (124)$$

$$\leq \kappa \|\rho_{XB} - \sigma_{XB}\|_1 \quad (125)$$

using Hölder's inequality in the second to last inequality. Swapping ρ_{XB} and σ_{XB} completes the proof. ■

VIII. SIMPLE EXAMPLES

A. BB84 States as Side Information

As an example, we consider the problem of calculating guesswork when one has four uniformly distributed letters to

guess from, each correlated to one of the four BB84 states [36]. That is,

$$\rho_{XB} = \frac{1}{4} \sum_{k=1}^4 |x_k\rangle\langle x_k|_X \otimes |\psi_k\rangle\langle\psi_k|_B \quad (126)$$

with the four $|\psi_k\rangle$'s being chosen from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. This example is firmly in the quantum realm of guesswork, as more information about the side information system B can be obtained via a quantum measurement than a classical one (in the computational basis, that is).

We establish an analytic upper bound on the guesswork by considering a particular POVM and associated sequences of guesses. We consider the POVM consisting of two orthogonal projectors $|\theta\rangle\langle\theta|$ and $|\theta^\perp\rangle\langle\theta^\perp|$ with $|\theta\rangle := \sin\theta|0\rangle + \cos\theta|1\rangle$. If the outcome corresponding to $|\theta\rangle\langle\theta|$ is obtained, then we guess in the order corresponding to $(1, +, -, 0)$. Similarly, the guessing order corresponding to the other outcome $\theta^\perp$ is $(0, -, +, 1)$, which is the reverse order.

By calculating the objective function of (87), we obtain

$$G(X|B) \leq \frac{1}{2} \left(1 \cdot \cos^2\theta + 2 \cdot \frac{1}{2} (1 + \sin 2\theta) + 3 \cdot \frac{1}{2} (1 - \sin 2\theta) + 4 \cdot \sin^2\theta \right) \quad (127)$$

$$= 1.75 + \frac{3}{2} \sin^2\theta - \frac{1}{4} \sin 2\theta. \quad (128)$$

With the aim of minimizing the guesswork, we choose $\theta = \frac{1}{2} \arctan \frac{1}{3}$, and obtain the right-hand side of (127) as $\frac{1}{4} (10 - \sqrt{10}) \approx 1.709430$. Moreover, the SDP in (87) can be solved numerically to obtain the same value, providing a matching numerical lower bound; see [20] for the code involved, including a high-precision demonstration using the SDP solver SDPA-GMP [37] showing agreement to 200 digits.

We also consider a generalization of this example, where the side information states are chosen from the set $\{|0\rangle, |1\rangle, |\psi(\varphi)\rangle, |\psi(-\varphi)\rangle\}$ where $|\psi(\varphi)\rangle = \cos(\varphi/2)|0\rangle + \sin(\varphi/2)|1\rangle$. The BB84 states are a special case of this ensemble with $\varphi = \pi/2$. For each of these ensembles, we compute the guesswork using our SDP formulation in (87). The results are shown in Figure 2.

Furthermore, we can use this example to delineate the difference, or gap, between guesswork with classical and quantum information. There are two ways of reducing the example of BB84 states to a classical setting: (a) restricting to measurements in the standard basis $\{|0\rangle, |1\rangle\}$, or (b) replacing the side information states $|+\rangle$ and $|-\rangle$ with the maximally mixed qubit state π_2 . In both of these cases, the side information then takes the form of a random variable Y . The joint probability distribution of variables XY is given as follows:

X \ Y	Y	
	0	1
x_1	1/4	0
x_2	1/8	1/8
x_3	1/8	1/8
x_4	0	1/4

Given either value of Y , one needs an average of 1.75 guesses. Hence in this classical analogue of the BB84 states,

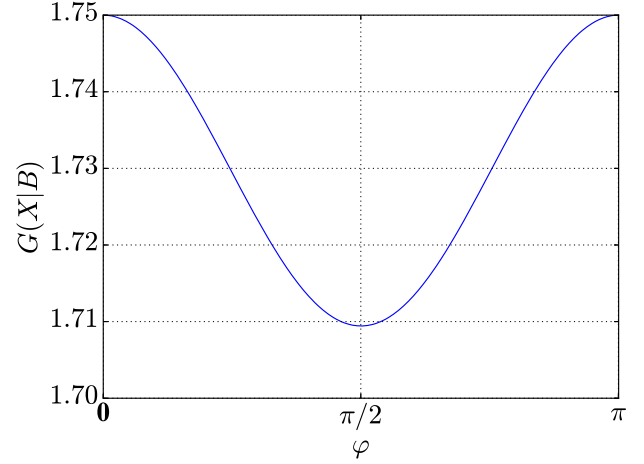


Fig. 2. The guesswork $G(X|B)$ as a function of the parameter φ , when X is uniformly distributed over $\{1, 2, 3, 4\}$ and the corresponding side information states are $\{|0\rangle, |1\rangle, |\psi(\varphi)\rangle, |\psi(-\varphi)\rangle\}$, where $|\psi(\varphi)\rangle = \cos(\varphi/2)|0\rangle + \sin(\varphi/2)|1\rangle$. We see that for classical states, i.e., when $\varphi = 0$ or $\varphi = \pi$, we obtain a maximum value of 1.75. For the BB84 states ($\varphi = \pi/2$), we achieve a minimum.

the guesswork is 1.75. This is higher than the lower value (≈ 1.709) that can be achieved by quantum measurements.

B. Qubit Trine States as Side Information

In this example, we consider the problem of calculating guesswork with three uniformly distributed letters to guess from, each correlated to one of the three qubit trine states [38]. That is,

$$\rho_{XB} = \frac{1}{3} \sum_{k=1}^3 |x_k\rangle\langle x_k|_X \otimes |\psi_k\rangle\langle\psi_k|_B \quad (129)$$

where $|\psi_k\rangle = \cos(k\frac{2\pi}{3})|0\rangle + \sin(k\frac{2\pi}{3})|1\rangle$.

As in the previous example of the BB84 states, here too we establish an analytic upper bound on the guesswork by considering a particular POVM. Consider that the measurement is characterized by two orthogonal projectors $|\theta\rangle\langle\theta|$ and $|\theta^\perp\rangle\langle\theta^\perp|$ with $|\theta\rangle := \cos\theta|0\rangle + \sin\theta|1\rangle$. For the sake of simplicity, we restrict to $\theta \in [0, \pi/2]$. For the outcome corresponding to θ , we guess in the order corresponding to (ψ_2, ψ_3, ψ_1) , and for the outcome corresponding to $\theta^\perp$, we guess in the order corresponding to (ψ_1, ψ_3, ψ_2) .

The objective function in (87) leads us to

$$\begin{aligned} G(X|B) &\leq \frac{1}{2} \cdot \frac{2}{3} \left(1 \cdot (\cos^2(\theta - 4\pi/3) + \sin^2(\theta - 2\pi/3)) \right. \\ &\quad \left. + 2 \cdot (\cos^2\theta + \sin^2\theta) \right. \\ &\quad \left. + 3 \cdot (\cos^2(\theta - 2\pi/3) + \sin^2(\theta - 4\pi/3)) \right) \\ &= \frac{4}{3} + \frac{2}{3} (\cos^2(\theta - 2\pi/3) + \sin^2(\theta - 4\pi/3)). \end{aligned} \quad (130)$$

The guesswork $G(X|B)$ is minimized by setting $f'(\theta) = 0$ where $f(\theta) = \cos^2(\theta - 2\pi/3) + \sin^2(\theta - 4\pi/3)$. This leads to $\theta = \pi/4$ and $G(X|B) = (2 - 1/\sqrt{3}) \approx 1.422649$. The SDP (87) is solved for this example as well, and the numerical result shows agreement with the analytic upper bound up to a relative tolerance of at least 10^{-7} .

Further, in this example, if we restrict to measuring in the standard basis (mimicking the classical analogue of the side information), then the average number of guesses needed is 1.5, in contrast to ≈ 1.4227 guesses needed using the optimal quantum measurement.

IX. GUESSWORK AS A SECURITY CRITERION: CERTIFYING AN IMPERFECT KEY STATE

A primitive in any cryptography scheme is the establishment of a secret key between two communicating parties. Quantum key distribution (QKD) protocols can produce a certifiably secure secret key by using pre-shared entanglement [39]. However, if the protocol is not implemented perfectly, as is the case in realistic scenarios, then some information can leak out to an eavesdropper. How secure is the key obtained in this “imperfect” scenario? In other words, if there is a small deviation from the ideal protocol, how does it affect the security of the key? We address this question considering the guesswork as a security criterion.

Consider two systems K and E , where K denotes the key system and encodes the secret key, and E is the system held by the eavesdropper. An ideal key state is of the form $\pi_K \otimes \rho_E$ where π_K refers to the maximally mixed state on the key system. This means that the eavesdropper can learn nothing about the key with access to the E system alone. An imperfect key, generally, is the joint state ρ_{KE} . Consider the promise that the imperfect key state is ε -close to an ideal one in normalized trace distance:

$$\frac{1}{2} \|\rho_{KE} - \pi_K \otimes \rho_E\|_1 \leq \varepsilon. \quad (131)$$

For an ideal key state, the expected guesswork for the eavesdropper is $\sum_k \frac{1}{|K|} k = \frac{|K|+1}{2}$ where $|K|$ indicates the cardinality of the alphabet associated to system K . We have the following result:

Theorem 9: For an imperfect key state satisfying the promise in (131), the following bound on guesswork holds

$$G(K|E) \geq \frac{|K|+1}{2} - |K|\varepsilon. \quad (132)$$

Proof: We apply the result of Lemma 10 below, which holds for the case of guesswork with classical side information. We know from Theorem 1 that a measured strategy for guesswork is equivalent to a quantum strategy. Using that fact, and by combining the promise $\frac{1}{2} \|\rho_{KE} - \pi_K \otimes \rho_E\|_1 \leq \varepsilon$ and the result in Lemma 10, we have (132). ■

Theorem 9 provides a robustness guarantee that imperfect key states continue to have near-maximal guesswork, if they remain close to an ideal key state in normalized trace distance.

Our proof of the lower bound in (132), as given above, is a consequence of the following extension of an analogous result pertaining to guesswork, due to Pliam [40, Theorem 3]. Pliam’s inequality states that for any random variable X with probability distribution p_X ,

$$\frac{|\mathcal{X}|+1}{2} - G(X) \leq \frac{1}{2} |\mathcal{X}| \|p_X - u_X\|_1, \quad (133)$$

where $G(X)$ denotes the guesswork and u_X denotes the uniform distribution.

Lemma 10: For random variables X and Y , the following bound holds for the guesswork:

$$\frac{|\mathcal{X}|+1}{2} - G(X|Y) \leq \frac{|\mathcal{X}|}{2} \|p_{XY} - u_X \otimes p_Y\|_1. \quad (134)$$

Proof: Consider the case of a joint distribution p_{XY} , with conditional distribution $p_{X|Y}$ and marginal distribution p_Y , and suppose that the value of y is fixed. Then we can invoke Pliam’s bound (133) to find that

$$\frac{|\mathcal{X}|+1}{2} - G(X|Y=y) \leq \frac{|\mathcal{X}|}{2} \|p_{X|Y=y} - u_X\|_1, \quad (135)$$

where the notation $G(X|Y=y)$ indicates the guesswork (without side information) of a random variable distributed according to $p_{X|Y}(\cdot|y)$. Taking the expectation of both sides with respect to the random variable Y , we find that

$$\frac{|\mathcal{X}|+1}{2} - \sum_y p_Y(y) G(X|Y=y) \quad (136)$$

$$\leq \frac{|\mathcal{X}|}{2} \sum_y p_Y(y) \|p_{X|Y=y} - u_X\|_1 \quad (137)$$

$$= \frac{|\mathcal{X}|}{2} \sum_y p_Y(y) \sum_x |p_{X|Y}(x|y) - u_X(x)| \quad (138)$$

$$= \frac{|\mathcal{X}|}{2} \sum_y \sum_x |p_{X|Y}(x|y) p_Y(y) - u_X(x) p_Y(y)| \quad (139)$$

$$= \frac{|\mathcal{X}|}{2} \sum_y \sum_x |p_{XY}(x,y) - u_X(x) p_Y(y)| \quad (140)$$

$$= \frac{|\mathcal{X}|}{2} \|p_{XY} - u_X \otimes p_Y\|_1. \quad (141)$$

Using the fact that

$$\sum_y p_Y(y) G(X|Y=y) = G(X|Y), \quad (142)$$

we can conclude the generalization of (133) in the presence of classical side information

$$\frac{|\mathcal{X}|+1}{2} - G(X|Y) \leq \frac{|\mathcal{X}|}{2} \|p_{XY} - u_X \otimes p_Y\|_1. \quad (143)$$

This concludes the proof. ■

Remark 11: Note that Theorem 8 gives the following continuity bound for the guesswork near $\pi_K \otimes \rho_E$:

$$|G(K|E)_\rho - G(K|E)_{\pi_K \otimes \rho_E}| \leq 2\varepsilon |K|, \quad (144)$$

and hence

$$G(K|E)_\rho \geq \frac{|K|+1}{2} - 2|K|\varepsilon. \quad (145)$$

Thus, the bound in (132) is slightly better than what we obtain by employing Theorem 8.

X. OPEN QUESTIONS

Guesswork presents an operationally-relevant method to quantify uncertainty, and has been relatively unexplored in the presence of quantum side information. We hope our investigation opens the door to further analysis of the guesswork and methods to compute it. In particular, our work leaves open the following questions:

- 1) Does equality hold in (74)? If so, the single-letter expression

$$\lim_{n \rightarrow \infty} \frac{1}{n} \ln G(X^n | B^n)_{\rho^{\otimes n}} = \tilde{H}_{\frac{1}{2}}^{\uparrow}(X|B)_{\rho} \quad (146)$$

holds, matching the classical case [4, Prop. 5].

- 2) Ref. [41] presented variational expressions for the measured Rényi divergences D_{α}^M and showed how those lead to efficient ways to compute the divergences. Are there similar variational formulas for $H_{\alpha}^{\uparrow, M}(X|Y)_{\rho}$? That could similarly provide an efficient way to compute the quantity.

ACKNOWLEDGMENT

Eric P. Hanson would like to thank Harsha Nagarajan for pointing out the transformation in [26, Equations (22)–(24)].

REFERENCES

- [1] J. Massey, “Guessing and entropy,” in *Proc. IEEE Int. Symp. Inf. Theory*, Trondheim, Norway, Jun. 1994, p. 204. [Online]. Available: <http://ieeexplore.ieee.org/document/394764/>
- [2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.
- [3] R. Lundin, T. Holleboom, and S. Lindskog, “On the relationship between confidentiality measures: Entropy and guesswork,” in *Proc. 5th Int. Workshop Secur. Inf. Syst.*, 2007, pp. 135–144.
- [4] E. Arikan, “An inequality on guessing and its application to sequential decoding,” *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 99–105, Jan. 1996. [Online]. Available: <http://ieeexplore.ieee.org/document/481781/>
- [5] E. Arikan and N. Merhav, “Guessing subject to distortion,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1041–1056, May 1998. [Online]. Available: <http://ieeexplore.ieee.org/document/669158/>
- [6] E. Arikan and N. Merhav, “Joint source-channel coding and guessing with application to sequential decoding,” *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1756–1769, Sep. 1998. [Online]. Available: <http://ieeexplore.ieee.org/document/705557/>
- [7] D. Malone and W. G. Sullivan, “Guesswork and entropy,” *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 525–526, Mar. 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1273661/>
- [8] R. Sundaresan, “Guessing under source uncertainty,” *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 269–287, Jan. 2007. [Online]. Available: <http://ieeexplore.ieee.org/document/4039677/>
- [9] M. K. Hanawal and R. Sundaresan, “Guessing revisited: A large deviations approach,” Aug. 2010, *arXiv:1008.1977*. [Online]. Available: <http://arxiv.org/abs/1008.1977>
- [10] M. M. Christiansen and K. R. Duffy, “Guesswork, large deviations, and Shannon entropy,” *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 796–802, Feb. 2013. [Online]. Available: <http://ieeexplore.ieee.org/document/6340341/>
- [11] I. Sason and S. Verdú, “Improved bounds on lossless source coding and guessing moments via Rényi measures,” *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4323–4346, Jun. 2018.
- [12] I. Sason, “Tight bounds on the Rényi entropy via majorization with applications to guessing and compression,” *Entropy*, vol. 20, no. 12, p. 896, Nov. 2018. [Online]. Available: <http://www.mdpi.com/1099-4300/20/12/896>
- [13] W. Chen, Y. Cao, H. Wang, and Y. Feng, “Minimum guesswork discrimination between quantum states,” *Quantum Inf. Comput.*, vol. 15, nos. 9–10, pp. 737–758, Jul. 2015, doi: [10.5555/2871422.2871424](https://doi.org/10.5555/2871422.2871424).
- [14] R. König, R. Renner, and C. Schaffner, “The operational meaning of min- and max-entropy,” *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4337–4347, Sep. 2009.
- [15] S. Fehr and S. Berens, “On the conditional Rényi entropy,” *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6801–6810, Nov. 2014.
- [16] R. L. Frank and E. H. Lieb, “Monotonicity of a relative Rényi entropy,” *J. Math. Phys.*, vol. 54, no. 12, p. 122201, Dec. 2013.
- [17] M. Müller-Lennert, F. Dupuis, O. Szechr, S. Fehr, and M. Tomamichel, “On quantum Rényi entropies: A new generalization and some properties,” *J. Math. Phys.*, vol. 54, no. 12, Dec. 2013, Art. no. 122203.
- [18] M. M. Wilde, A. Winter, and D. Yang, “Strong converse for the classical capacity of entanglement-breaking and Hadamard channels via a sandwiched Rényi relative entropy,” *Commun. Math. Phys.*, vol. 331, no. 2, pp. 593–622, Oct. 2014.
- [19] M. Tomamichel, *Quantum Information Processing With Finite Resources—Mathematical Foundations*, vol. 5. Cham, Switzerland: Springer, 2016.
- [20] E. P. Hanson, “Ericphanson/guessworkquantumsideinfo.jl: V0.1.2,” Tech. Rep., Sep. 2020, doi: [10.5281/zenodo.4018151](https://doi.org/10.5281/zenodo.4018151).
- [21] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, 2017, doi: [10.1137/141000671](https://doi.org/10.1137/141000671).
- [22] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd, “Convex optimization in Julia,” in *Proc. 1st Workshop High Perform. Tech. Comput. Dyn. Lang.*, Nov. 2014, pp. 18–28.
- [23] J. Watrous, *The Theory of Quantum Information*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, Apr. 2018.
- [24] K. R. Parthasarathy, “Extremal decision rules in quantum hypothesis testing,” *Infinite Dimensional Anal., Quantum Probab. Rel. Topics*, vol. 2, no. 4, pp. 557–568, Dec. 1999.
- [25] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004, Art. no. 40706.
- [26] S. Bhela, D. Deka, H. Nagarajan, and V. Kekatos, “Designing power grid topologies for minimizing network disturbances: An exact MILP formulation,” Mar. 2019, *arXiv:1903.08354*. [Online]. Available: <http://arxiv.org/abs/1903.08354>
- [27] C. Coey, M. Lubin, and J. P. Vielma, “Outer approximation with conic certificates for mixed-integer convex problems,” Aug. 2018, *arXiv:1808.05290*. [Online]. Available: <http://arxiv.org/abs/1808.05290>
- [28] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization* (Algorithms and Combinatorics), vol. 2. Berlin, Germany: Springer, 1993, doi: [10.1007/978-3-642-78240-4](https://doi.org/10.1007/978-3-642-78240-4).
- [29] J. Watrous, “Semidefinite programs for completely bounded norms,” *Theory Comput.*, vol. 5, pp. 217–238, Nov. 2009. [Online]. Available: <https://theoryofcomputing.org/articles/v005a011/>
- [30] M. Wilhelm and M. D. Stuber, “EAGO.jl: Easy advanced global optimization in Julia,” in *Proc. Optim. Methods Softw.*, 2020, pp. 1–26, doi: [10.1080/10556788.2020.1786566](https://doi.org/10.1080/10556788.2020.1786566).
- [31] G. Gamrath et al., “The SCIP optimization suite 7.0,” Optim. Online, Tech. Rep. ZIB-Report 20-10, Mar. 2020. [Online]. Available: <https://scipopt.org> and <https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/7802>
- [32] S. V. Johnjforrest et al. (Mar. 2020). *Coin-or/CBC: Version 2.10.5*. [Online]. Available: https://www.zenodo.org/record/3700700#.YV_wc33hXIU, doi: [10.5281/zenodo.3700700](https://doi.org/10.5281/zenodo.3700700).
- [33] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd. (Nov. 2019). *SCS: Splitting Conic Solver, Version 2.1.2*. [Online]. Available: <https://github.com/cvxgrp/scs>
- [34] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: Theory of Majorization and its Applications* (Springer Series in Statistics), 2nd ed. New York, NY, USA: Springer-Verlag, 2011. [Online]. Available: <http://www.springer.com/gb/book/9780387400877>
- [35] E. P. Hanson and N. Datta, “Maximum and minimum entropy states yielding local continuity bounds,” *J. Math. Phys.*, vol. 59, no. 4, Apr. 2018, Art. no. 042204.
- [36] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” in *Proc. IEEE Int. Conf. Comput., Syst., Signal Process.*, Bangalore, India, Dec. 1984, pp. 175–179.
- [37] M. Yamashita et al., “A high-performance software package for semi-definite programs: SDPA 7,” Dept. Math. Comput. Sci., Tokyo Inst. Technol., Tokyo, Japan, Res. Rep. B-460, 2010, p. 26.
- [38] A. S. Holevo, “Information-theoretical aspects of quantum measurement,” *Problemy Peredachi Informatsii*, vol. 9, no. 2, pp. 31–42, 1973.
- [39] A. K. Ekert, “Quantum cryptography based on Bell’s theorem,” *Phys. Rev. Lett.*, vol. 67, no. 6, pp. 661–663, Aug. 1991.
- [40] J. Pliam, “The disparity between work and entropy in cryptology,” Cryptol. ePrint Arch., Tech. Rep. 1998/024, 1998. [Online]. Available: <https://eprint.iacr.org/about.html>
- [41] M. Berta, O. Fawzi, and M. Tomamichel, “On variational expressions for quantum relative entropies,” *Lett. Math. Phys.*, vol. 107, no. 12, pp. 2239–2265, Dec. 2017.