Cheat-Penalised Quantum Weak Coin-Flipping

Atul Singh Arora,[*] Carl A. Miller,[†] Mauro E.S. Morales,[‡] Jamie Sikora[§]

October 3, 2025

Abstract

Coin-flipping is a fundamental task in two-party cryptography where two remote mistrustful parties wish to generate a shared uniformly random bit. While quantum protocols promising near-perfect security exist for *weak* coin-flipping—when the parties want opposing outcomes—it has been shown that they must be inefficient in terms of their round complexity, and it is an open question of how space efficient they can be. In this work, we consider a variant called *cheat-penalised* weak coin-flipping in which if a party gets caught cheating, they lose Λ points (compared to 0 in the standard definition). We find that already for a small cheating penalty, the landscape of coin-flipping changes dramatically. For example, with $\Lambda = 0.01$, we exhibit a protocol where neither Alice nor Bob can bias the result in their favour beyond $1/2 + 10^{-8}$, which uses 24 qubits and 10^{16} rounds of communication (provably 10^7 times better than any weak coin-flipping protocol with matching security). For the same space requirements, we demonstrate how one can choose between lowering how much a malicious party can bias the result (down to $1/2 + 10^{-10}$) and reducing the rounds of communication (down to $25,180$), depending on what is preferred. To find these protocols, we make two technical contributions. First, we extend the point game-protocol correspondence introduced by Kitaev and Mochon, to incorporate: (i) approximate point games, (ii) the cheat-penalised setting, and (iii) round and space complexity. Second, we give the first (to the best of our knowledge) numerical algorithm for constructing (approximate) point games that correspond to high security and low complexity. Our results open up the possibility of having secure and practical quantum protocols for multiparty computation.

[*]Centre for Quantum Science and Technology (CQST), IIIT Hyderabad
Joint Center for Quantum Information and Computer Science (QuICS)
Department of Computer Science, University of Maryland, College Park
atul.singh.arora@gmail.com
[†]Joint Center for Quantum Information and Computer Science (QuICS)
University of Maryland, College Park
camiller@umd.edu
[‡]Joint Center for Quantum Information and Computer Science (QuICS)
Department of Computer Science, University of Maryland, College Park
mauroms@umd.edu
[§]Virginia Tech Center for Quantum Information Science and Engineering (VTQ)
Department of Computer Science, Virginia Tech
sikora@vt.edu
The authors are listed in alphabetical order.

Contents

1 Introduction

Can two parties who are communicating at a distance, flip a coin in such a way that both are assured that the coin-flip was fair? In other words, is there an interactive protocol that allows two parties who do not trust one another to create a shared random bit? Classically, this task is impossible in the unconditional setting,[1] although it becomes possible if computational hardness assumptions are made (e.g., [MNS09]). In the quantum setting, however, there are secure protocols for coin-flipping that do not make computational hardness assumptions and rely only on minimal physical assumptions [ATSVY00, SR01, SR02, Amb04, Moc04, Moc05, Moc07]. Coin-flipping is a prime example of the unique capabilities of quantum mechanics for cryptographic tasks, and it was one of the original problems that started the field of quantum cryptography [BB84].

Quantum coin-flipping is part of the larger landscape of *two-party cryptography*, in which two persons who do not trust one another try to accomplish a cooperative task while simultaneously guarding against cheating by the other person. Other two-party primitives, such as bit commitment and oblivious transfer, automatically imply an ability to perform two-party coin-flipping. The task of coin-flipping is thus a natural watermark for measuring the power and the limitations of quantum mechanics in two-party cryptography.

The body of work in two-party cryptography offers many results that are beautiful from a theoretical standpoint but daunting from an experimental standpoint. Secure two-party computation, bit commitment, oblivious transfer, and strong coin-flipping have all been proved to be impossible to perform [Lo97, LC98, ABDR04, LC97, May97]—the best that one can achieve for those tasks are protocols with a fixed constant bias. For example, Kitaev proved that for strong coin-flipping—a coin-flipping problem where players are oblivious to the preference of the other—one of the parties can force a particular outcome with probability $\frac{1}{\sqrt{2}}$, meaning that the bias (i.e., the gap between $\frac{1}{2}$ and the probability of successful cheating) is at least $\frac{1}{\sqrt{2}} - \frac{1}{2}$. In 2007, Mochon [Moc07] offered a ray of hope by proving that *weak* coin-flipping—coin-flipping in which the desired outcome for each party is known to all a priori—is possible with bias arbitrarily close to zero.

However, Mochon's constructions are rather involved and complex. Specifically, the round complexity (the number of rounds of communication between the parties) required to achieve bias ϵ is very high. The best known upper bound on the number of communication rounds in Mochon's construction is $(1/\epsilon)^{O(1/\epsilon)}$ [ACG+14]. In fact, the situation was much worse because he used a non-constructive step in his proof, making even the circuit description of his protocols elusive. This was resolved in subsequent works that further improved upon and enriched Mochon's work [ACG+14, CGS16, CK17, Gan17, ARW19, ARV21, ARVW24]. Yet, none of them managed to improve the communication complexity. In 2020, Miller [Mil20] proved that any weak coin-flipping protocol with bias ϵ must have at least $\exp(\Omega(1/\sqrt{\epsilon}))$ round complexity. Consequently, there is no efficient protocol—i.e., no protocol in which the time-complexity is polynomial in $(1/\epsilon)$—for the original quantum coin-flipping problem. Unlike round complexity, no space-complexity lower bounds (i.e., lower bounds on the number qubits needed) are known for weak coin-flipping protocols. Therefore, even though Mochon's constructions are known to achieve bias ϵ with $O(\log(1/\epsilon))$ qubits [ACG+14], this is not known to be optimal. It is worth noting that Mochon's so-called *Dip-Dip-Boom* protocol approaches bias $1/6$ using only two qutrits and a qubit. In fact, Dip-Dip-Boom may be seen as the first protocol in Mochon's family of protocols, parametrised by k, that approach bias $\epsilon_{\text{Moch}}(k) := 1/(4k+2)$. Interestingly, beyond $k = 1$, no protocol is known to approach bias $\epsilon_{\text{Moch}}(k)$ with *constant* space complexity.

Two party-cryptography therefore seems challenged from the start. And yet, two-party cryptography was identified in [WEH18] (along with QKD and position verification) as one of three potential first-stage applications of quantum networks, and many experimental works have been done on the topic (see [BCWW24] for a summary). There is hope for two-party cryptography if certain theoretical obstacles can be overcome.

The simplest way to circumvent an impossibility proof such as [Mil20] is to change the mathematical model on which the proof is based. Fortunately, in the case of coin-flipping, existing literature [ABDR04,

[1] For any classical coin-flipping strategy, von Neumann's mini-max theorem implies that one of the parties will have a way to force a particular outcome. See [Mil22].

Moc07] gives us one natural and effective way of doing so—namely, by modifying the rules for coin-flipping to include a penalty $\Lambda \geq 0$ for cheating.

In this extended model, Alice and Bob each have three possible outputs: 0 (Alice wins the coin toss), 1 (Bob wins the coin toss), and $\perp$ (indicates the other party has been caught cheating). If Alice wins, she gains one point (Bob gains nothing); if Bob wins, he gains one point (Alice gains nothing); and if either party is caught cheating, they lose Λ points. This model, which we call Λ-*penalty weak coin-flipping*, or Λ-**penWCF** for short, makes a distinction between a party honestly losing the exchange (and receiving a score of 0) and a party getting caught cheating (and receiving a score of $-\Lambda$). This distinction is natural when considered as part of a broader scheme where parties engage in multiple interactions, introducing disincentive for malicious behaviour. There are a few key differences between the previously studied version of weak coin-flipping and the cheat-penalised setting. Specifically, in the cheat-penalised setting, each party wishes to maximise their respective *expected reward*, not just the probability that they will win. Thus, we want protocols where neither Alice nor Bob can cheat and obtain an expected reward more than $1/2 + \epsilon$, for a preferably small value of $\epsilon > 0$. In this work, we not only want the bias ϵ to be small, but also the round complexity rc and the space complexity sc. This motivates the following questions.

To what extent can introducing a cheat penalty improve the efficiency of weak coin-flipping protocols? How large must the cheat penalty be to achieve this?

Ambainis, Buhrman, Dodis and Roehrig [ABDR04] considered the cheat-penalised setting for multipartite coin-flipping. In particular, their results show that for $\Lambda \geq 4$, one can construct (bipartite) cheat-penalised weak coin-flipping protocols with constant space and round complexity, where the bias ϵ vanishes as $\Lambda \to \infty$. Mochon [Moc07], in a brief detour, gave a cheat-penalised protocol (a variant of Dip-Dip-Boom) for any $\Lambda \geq 0$ that uses constant space. He asserted, heuristically, that its bias also vanishes in the limit of both the round complexity and the cheating parameter Λ going to ∞. Besides these two works, we are unaware of any results in this direction.

By extending the formalism of point games (which we briefly recall below) to the cheat-penalised setting and developing a new numerical algorithm for constructing approximate point games, we obtain protocols that significantly outperform prior constructions in the following precise sense.

Main Result: The existence of more efficient, secure cheat-penalised weak coin-flipping

There exist weak coin-flipping protocols with cheat penalty $\Lambda = 0.01$, space complexity (number of qubits) $\mathsf{sc} = 24$ and the following trade-offs between the bias ϵ and the round complexity rc.

(i) More round efficient than WCF.	Bias $\epsilon = 10^{-8}$, round complexity $\mathsf{rc} = 10^{16}$ (rc is still 10^7 times better than any possible WCF protocol with a matching bias)
(ii) Constant space with low bias.	Bias $\epsilon = 10^{-10}$, round complexity $\mathsf{rc} = 10^{18}$, 24 qubits (rc is still 10^5 times better)

To obtain a better trade-off, we use $\Lambda = 1$ below.

(iii) Potentially amenable to experiments.	Bias $\epsilon = 0.09$, round complexity $\mathsf{rc} = 25,180$ ($\epsilon < \epsilon_{\text{Moch}}(2) = 1/10$)

In all the protocols above, Alice and Bob hold 8 qubits each and they exchange messages using a 8 qubit register. The comparison of Protocol (i) above with (non-cheat-penalised) WCF protocols is based on Miller's work [Mil20] that show that any WCF protocol with bias $\epsilon = 10^{-8}$, must have round complexity at least

$\mathsf{rc} = 10^{19}$ (these explicit numbers were computed later in [AM25]). Protocol (ii) shows that with 24 qubits, one can get very close to zero bias—i.e., we have a constant space protocol that can go as low as $\epsilon = 10^{-10}$. The only remotely comparable protocol in the (non-cheat-penalised) coin-flipping setting is Mochon's Dip-Dip-Boom that uses two qutrits and a qubit, to approach bias $\epsilon = 1/6$ in the limit of infinite round complexity. Protocol (ii) strongly suggests that in the cheat-penalised setting, there may be a constant space protocol that has bias arbitrarily close to zero. Protocol (iii) may be a reasonable compromise between the bias and the round complexity. Crucially, all our protocols are such that the message register can be discarded after every two rounds (i.e., after two messages have been exchanged).[2] Consequently, one only needs to keep the local quantum memories of the parties coherent throughout the execution of the protocol. These features suggest that Protocol (iii) may *potentially* be amenable to experiments. Figure 1 compares known protocols and our results, in terms of the bias and space complexity while Figure 11 (in the main text) compares bias and round complexity.

Our work makes several technical contributions in order to achieve this main result. To describe these, we briefly introduce some of the key concepts we use.

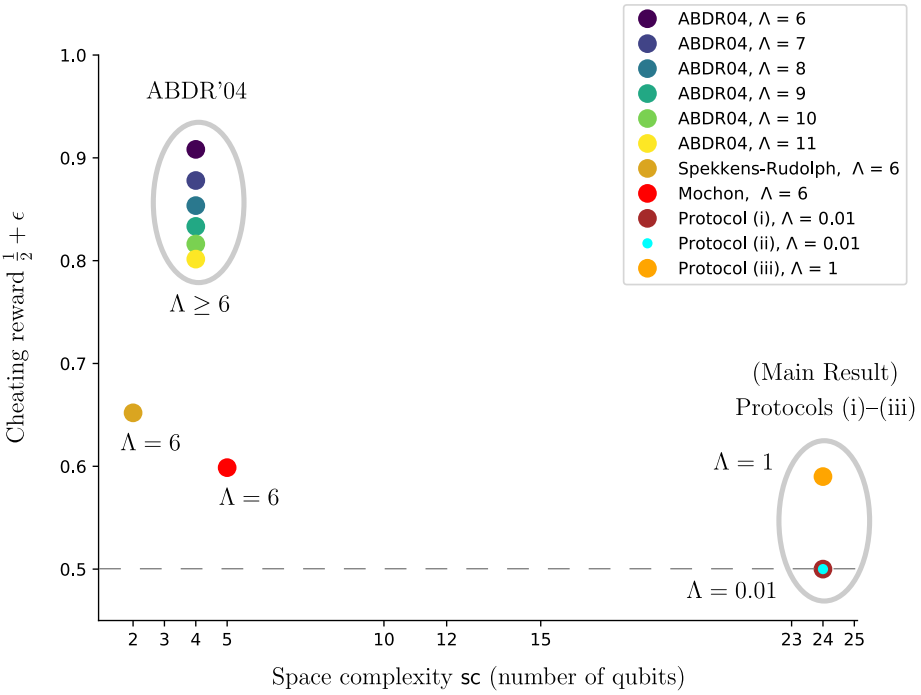


Figure 1: Comparison of cheat-penalised weak coin flipping protocols in terms of the greatest expected reward, i.e. $\frac{1}{2} + \epsilon$, and the number of qubits used in the protocol. We compare our Protocols (i)–(iii) to that of Ambainis, Buhrman, Dodis and Roehrig [ABDR04], the Spekkens-Rudolph protocol [SR02] (which we extend to the cheat-penalty setting), and Mochon's Dip-Dip-Boom cheat-penalised version [Moc07] (we rigorously derive and extend his heuristic bounds on the bias). Protocols (i)–(iii) are constructed using abstract objects we call cheat-penalised Time-Independent Point Games (penTIPGs). Protocols (i) and (ii) have $\mathsf{rc} = 10^{16}$ and $\mathsf{rc} = 10^{18}$ respectively and are based on penTIPG 3 which is illustrated in Figure 2 (not to scale). Protocol (iii) has round complexity $\mathsf{rc} = 25,180$ and is based on penTIPG 3. All these penTIPGs and related details appear in Section 9.

[2]We clarify that we count one message from one party to the other as constituting one round. In other works, sometimes one sees the convention where two messages are treated as one round of communication.

Point Games. Kitaev and Mochon [Moc07] introduced various so-called *point games* to design and analyse coin-flipping protocols. For a full introduction to this formalism we refer the reader to [Moc07, ACG+14]. Here, we focus on two variants: *time-independent point games* (**TIPGs**) and *time-dependent point games* (**TDPGs**). A TIPG is specified by two bivariate functions (h, v) where each encodes a finite set of weighted configurations of points on a two-dimensional plane. We use the notation $p[\![x, y]\!]$ to indicate that the point (x, y) has weight p where $p \in \mathbb{R}$. Slightly more formally, $[\![x, y]\!]$ is a bivariate function that is zero everywhere except on input (x, y) where it evaluates to one. A TIPG (h, v) for (non-penalised) weak coin-flipping has bias ϵ if it satisfies

$$h + v = \underbrace{\left[\!\left[\frac{1}{2} + \epsilon, \frac{1}{2} + \epsilon\right]\!\right]}_{\text{final config.}} - \underbrace{\left(\frac{1}{2}[\![0, 1]\!] + \frac{1}{2}[\![1, 0]\!]\right)}_{\text{initial config.}} \tag{1}$$

together with extra "validity conditions" that we are suppressing for now. The functions h and v encode a protocol for weak coin-flipping, i.e., for any such TIPG, there exist weak coin-flipping protocols that approach bias ϵ (up to arbitrary precision). As alluded to earlier, Mochon [Moc07] constructed a family of TIPGs parametrised by k that approach bias $\epsilon = 1/(4k + 2)$, thereby establishing the existence of weak coin-flipping with vanishing bias, $\epsilon \to 0$. However, in 2020, Miller showed that there is a relationship between the round complexity of weak coin-flipping protocols and the norms of h and v in any point game, leading to the conclusion that any weak coin-flipping protocol must have at least $\exp\big(\Omega(\epsilon^{-1/2})\big)$ rounds of communication. This renders them necessarily inefficient as $\epsilon \to 0$. A TDPG provides an alternative way of specifying a weak coin-flipping protocol. Instead of being defined by two functions, it is specified by a sequence of bivariate functions $(p_0, p_1, \ldots, p_n)$ such that $p_0 = \frac{1}{2}[\![0, 1]\!] + \frac{1}{2}[\![1, 0]\!]$, $p_n = \left[\!\left[\frac{1}{2} + \epsilon, \frac{1}{2} + \epsilon\right]\!\right]$ and each pair of functions (p_i, p_{i+1}) fulfil a validity condition, analogous to the one imposed in the TIPG formalism. While TIPGs encode protocols in a time-independent fashion—capturing only the start and end conditions with an implicit validity constraint—TDPGs represent the protocol as a sequence of valid transitions across rounds. This representation is therefore more directly connected to the round complexity of the protocol.

Contribution 1: Approximate cheat-penalised TIPGs and how to find them. We revisit the construction of point games in [Moc07], and define the notion of a *cheat-penalised TIPG*, which we denote by **penTIPG**. In essence, here one translates both the "initial" and "final configurations" by Λ (along both axes). While Mochon already informally considered such penTIPGs, his focus was establishing security for weak coin-flipping (without penalty) and not efficiency. Consequently, he did not study such games in further detail. In our work, we are concerned with *efficiency* as well, so our starting point is finding penTIPGs with small norm which can potentially give more efficient protocols. It does not take long to realise that a brute force search is hopeless, as the search space includes the space of all bivariate functions (h, v). Our first contribution is a numerical algorithm to find such point games. Not only does our algorithm yield solutions, it does so even for small values of Λ. However, these solutions are *approximate* and we therefore need to slightly relax the definition of penTIPG as follows.

Definition 1: $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG—Approximate cheat-penalised TIPGs (Definition 35 simplified)

We say (h, v) is a $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG with bias ϵ if it satisfies

$$\left\| h + v - \left(\underbrace{[\![\Lambda + \frac{1}{2} + \epsilon, \Lambda + \frac{1}{2} + \epsilon]\!]}_{\text{final config.}} - \underbrace{\left(\frac{1}{2} [\![\Lambda + 1, \Lambda]\!] + \frac{1}{2} [\![\Lambda, \Lambda + 1]\!] \right)}_{\text{initial config.}} \right) \right\|_1 \leq \varepsilon_{\mathsf{approx}} \qquad (2)$$

in addition to the "validity conditions" mentioned previously. By the *number of points* of (h, v), we mean the number of input pairs that are assigned non-zero weight by h and v, i.e., $|\mathsf{supp}\{h, v\}|$. By the *norm of* (h, v) we mean $\max\{\|h\|_1, \|v\|_1\}$.

We obtain various solutions but we highlight the one that results in a cheat-penalised WCF protocol with final bias $\epsilon = 10^{-8}$, where the cheat penalty is $\Lambda = 0.01$. As for efficiency, it uses 24 qubits and has round complexity more than *seven orders of magnitude* lower compared to Miller and Alnawaktha's [Mil20, AM25] lower bound for the analogous WCF protocol.

Theorem 1: Existence of approximate cheat-penalised time-independent point games with small norm (see Section 8)

For $\Lambda = 0.01$ and $\varepsilon_{\mathsf{approx}} = 10^{-18}$, there exists a $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG with bias $\epsilon = 10^{-14}$ (see Figure 2). Furthermore, it has norm 1.05, and uses at most 64 points.

Although we leave a rigorous performance analysis of our algorithm for future work, we note that once our algorithm finds a solution, the validity of the functions h and v can be easily checked. The code, together with other penTIPG solutions, are on GitHub [AMMS25]. It is important to emphasise that this result, on its own, does not resolve the central question of achieving efficient and secure weak coin-flipping. We need to tackle the reduction from, approximate penTIPG to protocols. This turns out to be subtle and requires new ideas. We proceed in two steps which we summarise next.

Contribution 2: From approximate penTIPGs to protocols. We first map the approximate penTIPGs to (exact) penTDPGs and then map the penTDPGs to protocols, for which we also give round and space complexities.

Mapping approximate time-independent point games to (exact) time-dependent point games. First, we convert an *approximate* cheat-penalised TIPG (penTIPG) into an *(exact) cheat-penalised time-dependent point game* (penTDPG). We defer its description to the Technical Overview (see Section 2.2) but note that it serves a similar purpose here as TDPGs do in the (non-penalised) WCF setting. Unlike a Λ-penTIPG, a **Λ-penTDPG** with bias ϵ is specified by a sequence of n configurations with positive weights $(p_0, p_1, p_2, \ldots, p_n)$ starting with p_0 which is the "initial configuration" and p_n which is the "final configuration" as in Equation 2. The intermediate configurations are required to satisfy "validity conditions" similar to those of TIPGs. As before, each change in configuration, intuitively, corresponds to one round of interaction.

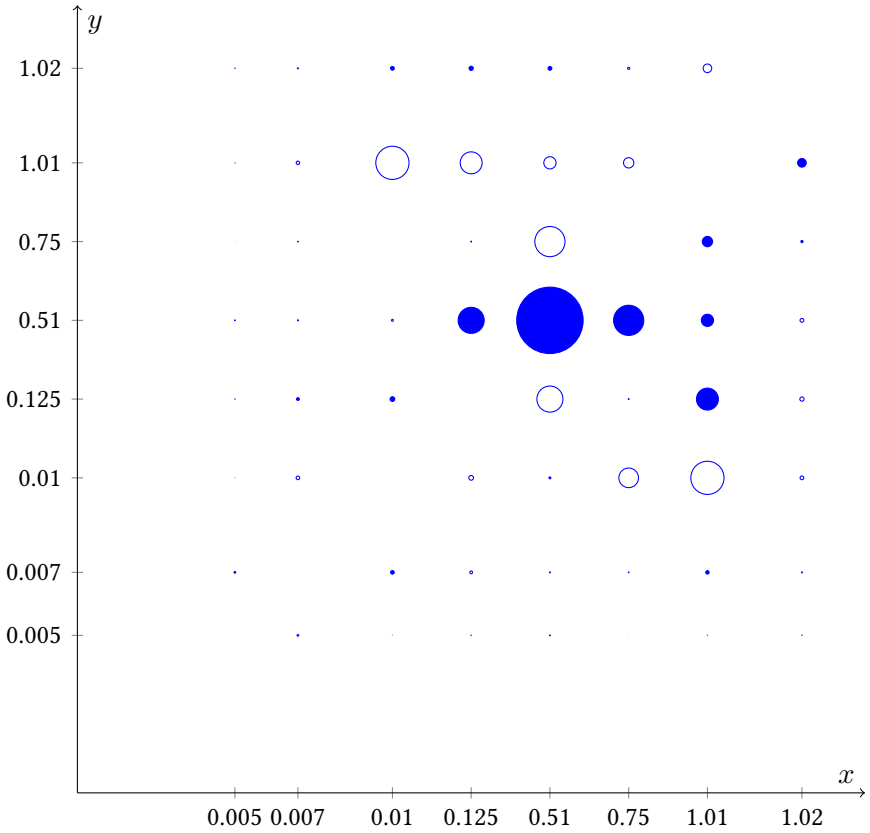


Figure 2: Graphical depiction (axes not to scale) of the parameters that entirely specify our bias $\epsilon = 10^{-10}$ protocol—except for the parameter that controls the trade-off with the round complexity. More precisely, the graph shows the function h of a time-independent point game (h, v) with cheat penalty $\Lambda = 0.01$ and approximation error $\varepsilon_{\mathsf{approx}} = 10^{-18}$, referenced above as a $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG. The filled circles correspond to positive weights and unfilled circles to negative weights while the radius indicates the magnitude of the weight.

Theorem 2: Mapping approximate penTIPGs to (exact) penTDPGs (Theorem 36 simplified)

Let (h, v) be a $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG with bias ϵ. Then, one can construct an (exact) Λ-penTDPG $(p_0, \ldots, p_n)$ with bias $\epsilon + \mathsf{err}$ where the trade-off between n and err can be tuned using two parameters (details suppressed). However, using these parameters, err can only be made to approach zero if $\varepsilon_{\mathsf{approx}} = 0$. Finally, the number of points in the (exact) penEBM point game $(p_0, \ldots, p_n)$ is at most the number of points in the approximate penTIPG (h, v), i.e., $\max_i\{|\mathsf{supp}(p_i)|\} \leq |\mathsf{supp}\{h, v\}|$.

We emphasise that the above deals with *approximate* point games and that such a reduction is new; it has not been studied in the non-cheat-penalised setting and could find application in other coin-flipping or quantum multiparty frameworks. Note that even if the bias ϵ is small, the error term err could dominate, thereby making the final bias of the protocol higher.

Mapping time-dependent point games to protocols. In Miller [Mil20], it was shown that an efficient WCF protocol leads to a point game with small norm (which he shows cannot exist). What we want is the converse in the cheat-penalised setting: point games with small norm lead to efficient protocols—together with explicit bounds on the resources required. While the former is a simple application of the triangle inequality, the latter turns out to be more delicate and involved (even given the point game analysis for

non-cheat-penalised WCF). As a result, we get the following theorem.

Theorem 3: Mapping time-dependent point games to protocols (Theorem 34 simplified)

Let $(p_0, \ldots, p_n)$ be a Λ-penTDPG with bias ϵ. Then there exists a Λ-penWCF protocol with the same bias, ϵ, that has round-complexity $\mathsf{rc} = 2n$ and space-complexity (number of qubits) $\mathsf{sc} = 3 \cdot \lceil \log_2(2\mu + 1) \rceil$ where μ is the greatest number of points in a configuration.

Combining these, we obtain the main result we stated earlier.

Significance. Weak coin-flipping stands out as one of the few cryptographic primitives that admits *unconditional security* in the quantum multiparty setting—a rare property in this field. This alone makes it an essential object of study, and our work shows that it can now be done more efficiently, giving hope that these can be implemented *and used* on real hardware. Moreover, optimal weak coin-flipping protocols are known to underpin a broad class of other cryptographic constructions. In particular, the only known optimal quantum protocols for strong coin-flipping [CK09], bit commitment [CK11], and oblivious transfer [CGS16] all critically rely on having access to optimal weak coin-flipping subroutines. As a consequence, the inefficiency of weak coin-flipping directly limits the practicality of these higher-level protocols. By revisiting the question of efficient and secure weak coin-flipping, we not only address an open problem but potentially unlock new avenues for secure quantum multiparty computation.

Organisation. The remainder of this article is organised as follows. Section 2 provides a technical overview of our results. Sections 3 through 6 present our first main contribution: extending the point game formalism to the cheat-penalised setting. Sections 7 through 10 then describe our second main contribution: a numerical algorithm for constructing point games, together with an analysis of the resulting protocols.

For the first contribution, we introduce Λ-penalty weak coin-flipping (Λ-penWCF) protocols and the point game formalism associated with them. We present the formulation of Λ-penWCF protocols as semidefinite programs (SDPs) in Section 3. Then, we present the formalism of EBM point games in the cheat penalise setting in Section 4. Sections 5 and 6 introduce time-dependent point games and time-independent point games respectively. We show how all these formulations are equivalent to Λ-penWCF protocols.

For the second contribution, Section 7 presents our numerical algorithm along with the intuition underlying its design. Section 8 illustrates some of the TIPGs obtained using this algorithm. Section 9 discusses cheat-penalised versions of existing WCF protocols from the literature. Finally, Section 10 compares the protocols obtained via our numerical approach with previously known constructions.

2 Technical Overview

In this section, we begin by presenting Kitaev's formalism for two-party protocols for (non-cheat-penalised) weak coin-flipping and its cheat-penalised variant. We then introduce the point game—explaining first time-dependent point games and subsequently time-independent point games—for both the non-cheat-penalised and cheat-penalised cases. We conclude by describing our numerical algorithm to find good point games.

2.1 Problem Statement

Weak coin-flipping (non-cheat-penalised)

Let n be an even positive integer. An n-message *weak coin flipping protocol* (**WCF**) may be described as follows. Suppose that two parties, Alice and Bob, wish to flip a coin with the outcome being either 0 ("heads") or 1 ("tails"). We assume that the parties have opposite preferences: Alice "wins" if the outcome is 0 and Bob "wins" when the outcome is 1. Alice possesses a local (quantum) memory register $\mathcal{A}$, and Bob possesses a local (quantum) memory register $\mathcal{B}$ and there is a third quantum message register $\mathcal{M}$ that is passed between the two parties. At the end of the protocol, each party outputs a bit representing what they believe to be the outcome of the coin flip. The protocol proceeds as follows.

1. Before round 1, the state of $(\mathcal{A}, \mathcal{B}, \mathcal{M})$ is a pure tripartite product state.
2. Alice possesses $\mathcal{M}$ on odd rounds, and Bob possesses $\mathcal{M}$ on even rounds.
3. For i odd, on the ith round Alice first applies a binary measurement to $(\mathcal{A}, \mathcal{M})$ to determine whether Bob has cheated. If Alice detects cheating, she aborts the protocol and outputs 0 (her desired outcome). If she does not detect cheating, she applies a unitary operator to $(\mathcal{A}, \mathcal{M})$ and then sends $\mathcal{M}$ to Bob.
4. For i even, Bob performs an analogous (possibly different) operation on the registers $(\mathcal{B}, \mathcal{M})$ and then sends $\mathcal{M}$ back to Alice.
5. After the nth round, Alice performs a binary measurement on $\mathcal{A}$ to obtain her output bit a, and Bob performs a binary measurement on $\mathcal{B}$ to obtain his output bit b.

Each player has a prescribed (honest) behaviour during round i, which is represented mathematically by binary measurements $\{E_i, \mathbb{I} - E_i\}$ and unitary operators U_i (both on $\mathcal{A} \otimes \mathcal{M}$ or $\mathcal{B} \otimes \mathcal{M}$ depending on whether i is even or odd) and final projection operators $\{\Pi_A^{(0)}, \Pi_A^{(1)}\}$ and $\{\Pi_B^{(0)}, \Pi_B^{(1)}\}$ (on $\mathcal{A}$ and $\mathcal{B}$ respectively). These operators define the coin flipping protocol. We assume that these operators are chosen so that if both players behave honestly, then the protocol behaves correctly, i.e.,

$$\Pr(a = b = 1) = \Pr(a = b = 0) = \frac{1}{2}. \tag{3}$$

This condition enforces that the outputs of Alice and Bob are the same and generated uniformly by the protocol.

A malicious party may deviate arbitrarily[3] (except that it cannot influence registers held by the honest party) from the prescribed protocol to bias the output towards their preferred value. Denote by P_A^* the greatest probability $\Pr(b = 0)$ with which a malicious Alice interacting with an honest Bob, can make him output $b = 0$. Denote by P_B^* the analogous quantity where Bob is malicious and interacts with an honest Alice.

[3] No bounds on their computational abilities are assumed.

Given a WCF protocol, one can cast both P_A^* and P_B^* as semidefinite programs. The bias of the protocol is given by

$$\epsilon \;:=\; \max\{P_A^*, P_B^*\} - \frac{1}{2} \;\in\; [0, 1/2] \tag{4}$$

which is the maximum probability one of the parties can force their desired outcome above $1/2$, the honest outcome probability.

Cheat-penalised weak coin-flipping

The structure of the problem changes significantly if we (following [ABDR04]) introduce a third parameter Λ, the *cheat penalty*. Suppose that we allow Alice and Bob to output, in addition to 0 and 1, an abort symbol $\perp$ that indicates cheating has been detected. Intuitively, we still want outcomes $0, 1$ to determine who "wins" but the outcome $\perp$ means that the malicious party not only "loses" but is penalised. To make this quantitative, on outcomes 0 and 1, we assign 1 point to the winner and 0 to the loser, while on outcome $\perp$, we penalise the cheating party by subtracting Λ points. We assume that each party wants to maximise their expected reward. This discussion leaves room for confusion—who is the "we" in the description above, what happens if both parties output $\perp$ and so on. To clarify that, we need to be slightly more formal. We first modify the protocol to accommodate three outcomes.

- For steps $3-4$, if Alice or Bob detects cheating, they abort the protocol and return the symbol $\perp$ (instead of just declaring themselves the winner).
- On step 5, Alice performs a ternary measurement $\{\Pi_A^{(0)}, \Pi_A^{(1)}, \Pi_A^{(\perp)}\}$ and returns the result. Bob likewise applies $\{\Pi_B^{(0)}, \Pi_B^{(1)}, \Pi_B^{(\perp)}\}$ and returns the result.

We enforce the same correctness condition (as in Equation 3). As a consequence, we note that when both parties are honest, the outcome $\perp$ never occurs, and one of the parties receives 1 point, each with equal probability.

As for security, we consider the setting where one party is malicious and the other honest (i.e., follows the protocol).

- Alice is honest, Bob is malicious.
  - If Alice outputs 1 (i.e., "Bob wins"), Bob gets 1 point.
  - If Alice outputs 0 (i.e., "Alice wins"), Bob gets 0 points.
  - If Alice outputs $\perp$ (i.e., "cheating detected"), Bob loses Λ points.
- Bob is honest, Alice is malicious.
  - If Bob outputs 0 (i.e., "Alice wins"), Alice gets 1 point.
  - If Bob outputs 1 (i.e., "Bob wins"), Alice gets 0 points.
  - If Bob outputs $\perp$ (i.e., "cheating detected"), Alice loses Λ points.

Notice that this clarifies the issues raised earlier—the goal for a malicious party is to convince an honest party to give them as many points as possible. Denote by R'_A the supremum of the average scores achieved by a malicious Alice when interacting with an honest Bob. Similarly, denote by R'_B the analogous quantity for malicious Bob interacting with an honest Alice. Then, the bias of the protocol is given by

$$\epsilon \;:=\; \max\{R'_A, R'_B\} - \frac{1}{2} \;\in\; [0, 1/2]. \tag{5}$$

It is also elementary to note that for $\Lambda = 0$, we reduce to (non-cheat-penalised) WCF. To see this, notice that the reward for malicious Bob is the same when Alice outputs 0 or if she outputs $\perp$. One can argue similarly for malicious Alice and honest Bob. Thus, the distinction between the outcome $\perp$ and the bit corresponding to "losing", vanishes, bringing us back to (non-cheat-penalised) WCF where instead of $\perp$ the honest party declares themselves to be the winner.

Cheat-penalised weak coin-flipping, reformulated

It turns out that, for technical reasons which we explain shortly, it is easier to work with a "translated" scoring convention where the "cheating detection" outcome $\perp$ corresponds to getting 0 points. More precisely, we add Λ points globally, to get the following convention.

- Alice is honest, Bob is malicious.
  - If Alice outputs 1 (i.e., "Bob wins"), Bob gets $\Lambda + 1$ points.
  - If Alice outputs 0 (i.e., "Alice wins"), Bob gets Λ points.
  - If Alice outputs $\perp$ (i.e., "cheating detected"), Bob gets 0 points.
- Bob is honest, Alice is malicious.
  - If Bob outputs 0 (i.e., "Alice wins"), Alice gets $\Lambda + 1$ points.
  - If Bob outputs 1 (i.e., "Bob wins"), Alice gets Λ points.
  - If Bob outputs $\perp$ (i.e., "cheating detected"), Alice gets 0 points.

Denoting by R_A^* (resp. R_B^*) the greatest average scores a malicious Alice (resp. Bob) gets against an honest Bob (resp. Alice), we can rewrite the bias as

$$\epsilon \;=\; \max\{R_A^*, R_B^*\} - \Lambda - \frac{1}{2} \;\in\; [0, 1/2]. \tag{6}$$

Just as one can cast P_A^* and P_B^* from (non-cheat-penalised) WCF as a semidefinite program (SDP), one can also cast R_A^* and R_B^* as an SDP. Having non-negative reward helps not only because one can still work with positive semidefinite matrices easily, but also because the way the "cheat detection projectors" $\{E_i, \mathbb{I} - E_i\}$ appear in the SDP remains largely unchanged—in the (non-cheat-penalised) setting, these projectors project onto a subspace that never contributes to the support of the objective being maximised, and with this convention, it continues to be the case. Henceforth, we stick with the Λ-translated scoring convention to compute the rewards.

2.2 The point game formalism

We now return to (non-cheat-penalised) WCF and describe the point game formalism Kitaev and Mochon [Moc07] introduced to build secure protocols, along with its generalisation to the cheat-penalised setting. We begin by presenting time-dependent point games and then simplify them further to time-independent point games.

To appreciate the origin of point games, it is instructive to recall the connection between SDPs and WCF. As we already mentioned, the cheating probabilities P_A^* and P_B^* for Alice and Bob can each be cast as an SDP. Using SDP duality, one can establish upper bounds on P_A^* and P_B^* by finding a feasible solution to the dual SDP. The key insight of Kitaev and Mochon was to combine the spectra of these dual feasible solutions with the honest state of the protocol, and distil them into "configurations of points on a plane" subject to specific rules that correspond to the constraints of the dual SDP. Given the structure of this dual SDP (as imposed by WCF), one can go beyond analysing a specific protocol and consider such constraints more generically. Viewed this

way, they can be shown to have an alternate characterisation in terms of operator monotone functions. Since operator monotone functions have been studied separately and admit a simple characterisation themselves, one can use this characterisation to simplify the constraints on "configurations" into what are called "validity conditions". These sequences of "configurations" are what we have been calling *time-dependent point games*. Since these "configurations" still correspond to the protocol, there are as many of them as there are interactions in the protocol—and hence the adjective "time dependent". These are then simplified further by, abstractly, merging all of Alice's steps into one step, and Bob's steps into another step to obtain a much simpler object called a *time-independent point game* that is specified using only two "moves" (captures the transition from one configuration to the next). Going from protocols to time-dependent point games and finally time-independent point games is relatively straightforward—in hindsight—but going from time-independent point games all the way back to protocol, is highly non-trivial [Moc07, ACG+14].

2.2.1 Time-Dependent Point Games

Fortunately, the description of these point games is straightforward and that suffices for our purposes here. We start by defining the term *one-dimensional configuration* (later we use *frame* and configuration interchangeably) to mean a function $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that has finite support (i.e., $f(x) = 0$ for all but a finite number of values of x). Given two such configurations f_1 and f_2, we say that the transition $f_1 \to f_2$ is *valid* if:

$$\sum_x f_1(x) = \sum_x f_2(x) \tag{7}$$

and for all $\lambda > 0$,

$$\sum_x f_1(x) \left(\frac{x\lambda}{x+\lambda}\right) \leq \sum_x f_2(x) \left(\frac{x\lambda}{x+\lambda}\right). \tag{8}$$

A *two-dimensional configuration* is a function $p\colon \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ (i.e., a non-negative function on the upper right-hand quadrant of a Cartesian coordinate system) that has finite support. (If we use the word "configuration" or "frame" by itself, we mean a two-dimensional configuration.) A transition $p_1 \to p_2$ is *horizontally valid* if it is valid on all horizontal lines in the Cartesian coordinate system, and it is *vertically valid* if it is valid on all vertical lines in the Cartesian coordinate system. An example of a horizontally valid move is shown in Figure 3.

We continue using the $[\![x, y]\!]$ notation: for any pair of non-negative real numbers (x, y), we denote by $[\![x, y]\!]$ the function from $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ that maps (x, y) to 1 and all other points to 0. With this, we define the start-configuration as $s := \frac{1}{2}[\![0, 1]\!] + \frac{1}{2}[\![1, 0]\!]$ and the end-configuration as $e := [\![\beta, \alpha]\!]$. Now, a (valid) *time-dependent point game* (**TDPG**) with final point (β, α) is a sequence of two-dimensional configurations $(p_0 := s, p_1, p_2, \ldots, p_{n-1}, p_n := e)$ such that $p_i \to p_{i+1}$ is horizontally valid for all even i and vertically valid for all odd i.

In [Moc07], it was proved that time-dependent point games are equivalent to weak coin-flipping protocols in the following sense.

Theorem 1 (WCF protocol–TDPG equivalence (informal; [Moc07]))**.** *The following holds.*

WCF $\implies$ ***TDPG.*** *Given any $\delta > 0$, and a weak coin-flipping protocol with cheating probabilities P_A^* and P_B^*, using n rounds of interaction, there exists a time-dependent point game $(p_0, \ldots, p_n)$ with final point β, α where $\beta = P_B^* + \delta$ and $\alpha = P_A^* + \delta$.*

WCF $\impliedby$ ***TDPG.*** *Given any $\delta > 0$ and any time-dependent point game $(p_0, \ldots, p_n)$ with final point (β, α), there exists a weak coin-flipping protocol with cheating probabilities $P_A^* = \alpha + \delta$ and $P_B^* = \beta + \delta$.*

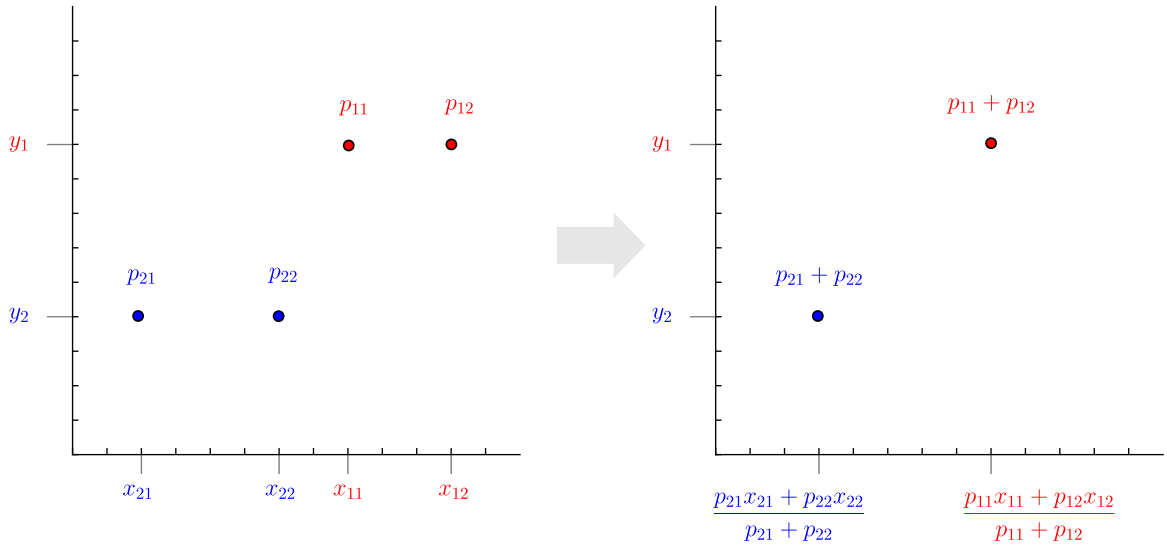


Figure 3: An example of a horizontally valid transition. The red points along $y = y_1$ are *merged* into single point at the average x-coordinate of the initial two red points. Similarly, blue points along $y = y_2$ are *merged* into a single point at the average x-coordinate of the two initial blue points. Such *merge* operations are known to always satisfy the validity conditions in Equations 7 and 8.

These were established in [Moc07, ACG+14] and one of our main technical contributions is to adapt to these to the cheat-penalised settings. We make a few small remarks about these results since our focus here is efficiency.

- We emphasise that the "TDPG $\implies$ WCF" in these works relied on a *non-constructive* step. As a consequence, for a given a TDPG, it was known that a protocol exists but there was no way to find the unitaries corresponding to this protocol—besides doing a brute-force search. This was later remedied in [ARW19] who gave a general numerical algorithm, they called the EMA algorithm, for finding the required unitaries corresponding to any valid transition. We mention this here because below, when we consider the cheat-penalised setting, the start-configuration and end-configurations change. However, the notion of valid transitions stays unchanged and therefore one can still use the EMA algorithm to construct *explicit unitaries* for these protocols in the cheat-penalised setting.

- The original proofs in [Moc07, ACG+14] produced protocols with round-complexity n (the length of the TDPG) where the message register had to kept coherent throughout the execution of the protocol. The proof in [ARW19] produced protocols where the message register can be discarded after every interaction—however, the round complexity of this protocol was far from optimal. We omit the details but briefly, one could only make transitions line by line (and not "frame by frame") leading to inefficiencies. When we consider the cheat-penalised setting below, we show how to get the best of both approaches.[4]

- In [ARW19] it was also noted that one can have $\delta = 0$ in the "TDGP $\implies$ WCF" step by using projectors after the unitaries (instead of before, as in [ACG+14]) appropriately. We therefore use that convention

[4] In fact, this improvement also applies to (non-cheat-penalised) weak coin-flipping.

here. In the cheat-penalised, with the translated scoring system, we are able to preserve the benefits of placing the projectors after.

Fix any $\Lambda \geq 0$. A Λ-*penalty time-dependent point game*, denoted by Λ-**penTDPG**, with final point (β, α) is a sequence of configurations $(p_0, \ldots, p_n)$ such that $p_0 = \frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!]$, $p_n = [\![\beta, \alpha]\!]$ and the transitions $p_i \to p_{i+1}$ are valid (exactly as in the (non-cheat-penalised) WCF setting; see Definition 32). Skipping the "easy direction"[5] for brevity, below we state a stronger variant of Theorem 3 that we prove holds in the cheat-penalised setting. In fact, this result also applies to (non-cheat-penalised) WCF when $\Lambda = 0$.

Theorem 2 (Λ-penTDPG $\implies$ Λ-penWCF protocol (Theorem 34 simplified))**.** *Fix any $\Lambda \geq 0$ (also works for $\Lambda = 0$). Let $(p_0, \ldots, p_n)$ be a Λ-penalty Time-Dependent Point Game that has (β, α) as the final point. Then, there exists a cheat-penalised WCF protocol with penalty Λ whose "cheating rewards" are bounded as $R_A^* \leq \alpha$ and $R_B^* \leq \beta$. Furthermore, it uses $2n$ rounds of communication and $3 \cdot \log\lceil(\max_j (2\mu_j) + 1)\rceil$ qubits where $\mu_j = |\mathsf{supp}(p_j)|$ is the number of points with non-zero weight in the configuration p_j.*

To prove the theorem, we overcome the aforementioned challenges in Section 4.3 and Section 5.2. In fact, we show the equivalence by going through an intermediate object called "Expressible-by-Matrices" (EBM) point games. We defer these details to Section 4.

2.2.2 Time-Independent Point Games

Moving on, as we already suggested, one can simplify time-dependent point games even further by considering time-independent point games. To this end, we first consider the (non-cheat-penalised) WCF setting and introduce some notation. Let us say that a *one-dimensional move* is a (not necessarily non-negative) function from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}$ with finite support, and that a *two-dimensional move* is a function from $\mathbb{R}^2_{\geq 0}$ to $\mathbb{R}$ with finite support. A one-dimensional move f is *valid* if

$$\sum_{x \in \mathbb{R}_{\geq 0}} f(x) \quad = \quad 0 \tag{9}$$

and

$$\sum_{x \in \mathbb{R}_{\geq 0}} f(x) \left(\frac{x\lambda}{x+\lambda}\right) \quad \geq \quad 0 \quad \forall \lambda \geq 0. \tag{10}$$

A two-dimensional move is horizontally valid if its rows are valid, and vertically valid if its columns are valid. A *time-independent point game* (**TIPG**) with final point (β, α) is a pair (h, v) of two-dimensional moves such that h is horizontally valid and v is vertically valid such that $h + v = e - s$ (where recall $s := 1/2\,[\![0, 1]\!] + 1/2\,[\![1, 0]\!]$ and $e = [\![\beta, \alpha]\!]$).

It is straightforward to observe that given a time-dependent point game $(p_0, p_1, \ldots, p_n)$ from s to e, one can construct a time-independent point game from s to e as

$$h \quad := \quad -p_0 + p_1 - p_2 + p_3 + \ldots + p_{n-1} \tag{11}$$

$$v \quad := \quad -p_1 + p_2 - p_3 + p_4 - \ldots + p_n. \tag{12}$$

The converse also holds [Moc07, ACG+14] although establishing it takes much more work—and it is closely related to the round complexity.

[5] In fact, we realised that one of the steps in [ACG+14] that is used to establish this "easy direction" for (non-cheat-penalised) WCF—specifically, the proof of Proposition 9—contains an error. The statement, fortunately, is still correct. Very briefly, the issue is that they implicitly assume that the spectrum of a matrix remains unchanged under certain kinds of projections. This is easily fixed by deferring all projectors to the end. See Proposition 17 which also holds for $\Lambda = 0$.

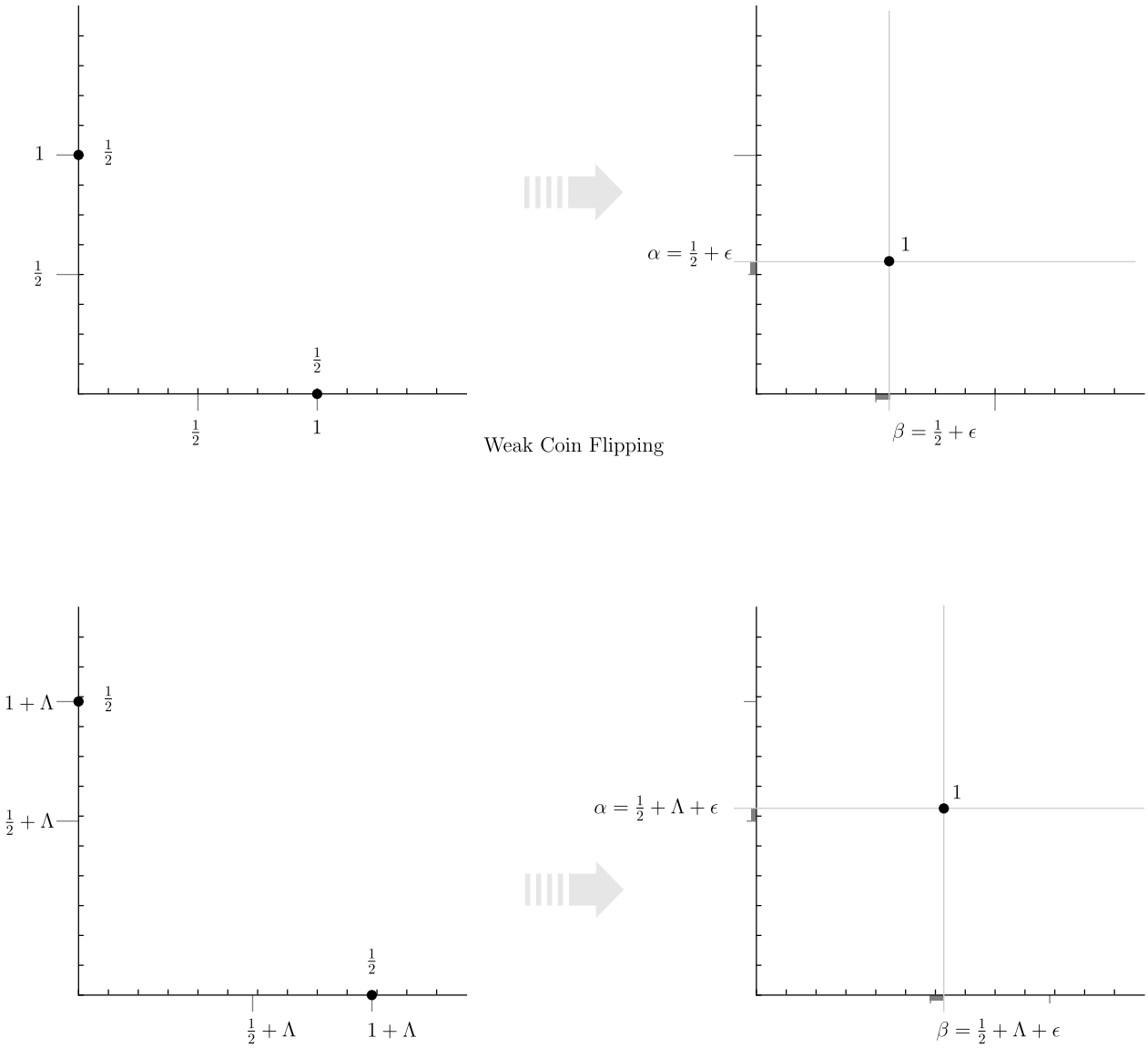


Figure 4: The point game problems corresponding to ordinary weak coin-flipping (top) and Λ-penalty weak coin-flipping (bottom). The points in the initial configurations each have weight $1/2$, and the points in the final configurations each have weight 1.

Theorem 3 (TDPG–TIPG equivalence (informal [Moc07, ACG+14])). *The following holds.*

TDPG $\implies$ ***TIPG.*** *Given a time-dependent point game (TDPG) with final point* (β, α) *there is a time-independent point game (TIPG) with final point* (β, α).

TDPG $\impliedby$ ***TIPG.*** *For every* $\delta > 0$, *given a time-independent point game (TIPG) with final point* (β, α) *there is a time-dependent point game TDPG* $(p_0, \ldots, p_n)$ *with final point* $(\beta + \delta, \alpha, +\delta)$ *where* n *grows as* δ *shrinks.*

At a high level, the key difficulty in converting a TIPG into a TDPG is that there is no way to "start the TIPG". Specifically, the starting configuration for a TDPG is always s. However, to use the move h as a transition in the TDPG, one must ensure that there are points at h^- already present in the previous configuration. Here h^- is the negative part of h and encodes all the points with negative weight. To remedy this, Mochon introduces the notion of a *catalyst state*. The idea is to take some small amount of weight from s and deposit it as per h^-. This process also creates a few extra points with high coordinates. However, since the total weight of these points is small, it does not influence the final point of the TDPG too much. With a scaled down version of h^- present, one can apply a scaled down version of the h move, followed by the v, repeatedly until most of the weight has accumulated at the point (β, α). Finally, all remaining points are "absorbed" into this $(\beta, \alpha$ point to obtain the final point $(\beta + \delta, \alpha + \delta)$ where δ depends on a variety of parameters. Intuitively, smaller the δ, smaller the weight of the catalyst state—and thus greater the number of transitions in the TDPG.

We briefly remark on some of the challenges that arise when one introduces cheat penalty and considers cheat-penalised TDPGs (penTDPGs) and TIPGs (penTIPGs) that we describe shortly.

- For the cheat-penalised version, like we explained in the introduction, we need to allow the TIPG to be approximate—otherwise we cannot use results from our numerical algorithm (explained in Section 2.3 below). This means that not only does one need to prepare the catalyst, but one also needs to deposit weight on the points that arise because of this approximation. While the weight on the catalyst in the non-cheat-penalised setting is essentially a free parameter, it is unclear if this is still the case in view of the approximation error.

- In (non-cheat-penalised) TDPGs and TIPGs, no point has either coordinate below 0. In the cheat-penalised setting, everything gets translated by (Λ, Λ). One might therefore suspect that in this setting, no point in the point game can have either coordinate below Λ. However, this is not the case—and as we explain later, numerical evidence suggests that being able to use points with coordinates below Λ seems to be crucial to obtain improvements. Why is this pertinent to the connection between cheat-penalised TDPGs and TIPGs? It turns out that Mochon's original construction for catalyst state, when applied to the cheat-penalised setting, cannot handle points that have coordinates lower than Λ. We use a slightly different route to circumvent this issue (but assume $\Lambda > 0$ to simplify the analysis slightly; note that $\Lambda = 0$ works for the previous results).

- Finally, the analysis in [Moc07] is not concerned with resource usage while the analysis in [ACG+14] is only considers asymptotic dependence of the round complexity on the bias. This is not enough to compute concrete numbers for specific protocols. We need to compute explicit formulae relating the bias and round complexity, in terms of various parameters (including approximation error)—while preserving properties that control the space-complexity.

Cumulatively, when deriving the analogous result (i.e., cheat-penalised TIPG $\implies$ cheat-penalised TDPG), we need to be much more careful about our parametrisation and even so, we find that δ cannot be arbitrarily small. This, in turn, means that the approximation factor limits the bias, regardless of how many rounds of communication are allowed.

In the technical overview, we have not yet detailed penTIPGs. Unsurprisingly—at least at this point—a Λ-*penalty, time-independent point game with approximation error* $\varepsilon_{\mathsf{approx}}$, or more briefly, $(\Lambda, \varepsilon_{\mathsf{approx}})$-**penTIPG** is a pair of moves (h, v) where h is horizontally valid, v is vertically valid and they satisfy $\|h + v - (e - s)\|_1 \leq \varepsilon_{\mathsf{approx}}$ for $e := [\![\beta, \alpha]\!]$, and $s := 1/2\,[\![\Lambda + 1, \Lambda]\!] + 1/2\,[\![\Lambda, \Lambda + 1]\!]$. We show that these approximate time-independent point games can also be converted into time-dependent point games with a slightly increased final coordinate. This involves various parameters so we first state the result and specify the parameters afterwards.

Theorem 4 ($\Lambda, \varepsilon_{\mathsf{approx}}$-pen TIPG $\implies$ Λ-pen TDPG (Theorem 36 reformulated)). *Fix any $\Lambda > 0$ (cannot have $\Lambda = 0$). Given a symmetric $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG (h, v), i.e., $h = v^T$, with final point (β, α), choose any*

$$c \in \left(0, \frac{m_{\mathsf{min}}^2}{(\Lambda + 1) \cdot \Lambda}\right)$$

where $m_{\mathsf{min}} \geq 0$ is the minimum coordinate in (h, v). The choice of c specifies a $\delta_{\min} < 1$. Then, for every $\delta \in (\delta_{\min}, 1)$ there is a Λ-penalty TDPG $(p_0, \ldots, p_n)$ with final point $(\beta + \mathsf{err}, \alpha + \mathsf{err})$ where

$$\mathsf{err} = \sqrt{\delta \cdot (m_{\mathsf{max}} - \alpha)(m_{\mathsf{max}} - \beta)},$$

$n = 10 + 2/\eta_{\mathsf{clyst}}$. *Here m_{max} is closely related with the maximum coordinate in (h, v) and the catalyst state and η_{clyst} is proportional to the weight assigned to the catalyst state (controlled by δ) but inversely proportional to $\|h^-\|$. Explicit formulae for these are listed below. The number of points in p_i is at most the number of points in (h, v), i.e., $|\mathsf{supp}(p_i)| \leq |\mathsf{supp}\{h\} \cup \mathsf{supp}\{v\}|$ for all $i \in \{0, \ldots n\}$.*

To describe the parameters exactly, we observe that there exist $\varepsilon_1, \varepsilon_2 \leq \varepsilon_{\mathsf{approx}}$ such that negative part of $h + v$ equals $(1 - \varepsilon_1)\, s + \varepsilon_1 \cdot s_{\mathsf{error}}$ and the positive part of $h + v$ equals $(1 - \varepsilon_2)e + \varepsilon_2 \cdot e_{\mathsf{error}}$ where $\|s_{\mathsf{error}}\|_1, \|e_{\mathsf{error}}\|_1 = 1$. Recall that c was already fixed above (in the theorem). Using these, we have:

- $\delta_{\min} := (1 - \varepsilon_2) \cdot \frac{c'\varepsilon_1}{1 + c'\varepsilon_1} + \varepsilon_2$, with $c' := c^{-1} - 1$,
- $m_{\mathsf{max}} := \max\{\mathsf{max\text{-}coordinate}(h), \tilde{m}_{\mathsf{max}}\}$ with
- $\tilde{m}_{\mathsf{max}} := \min\left\{\left|(1 - w)\left(\frac{1}{(\Lambda+1)} - \frac{w}{m_{\mathsf{min}}}\right)^{-1}\right| : w \in \{w^{\pm}\}\right\}$ for

$$w^{\pm} = \frac{\sqrt{8c\Lambda^2(\Lambda + 1)^2 + m_{\mathsf{min}}^2(8c\Lambda(\Lambda + 1) + 1) - 8c\Lambda\left(2\Lambda^2 + 3\Lambda + 1\right) m_{\mathsf{min}}} \pm m_{\mathsf{min}}}{2(\Lambda + 1)(m_{\mathsf{min}} - \Lambda)},$$

and finally,

- $\eta_{\mathsf{clyst}} := \frac{\delta_{\mathsf{clyst}}}{\|h^-\|} \cdot \frac{c(1-\varepsilon_1)+\varepsilon_1}{(1-\delta_{\mathsf{clyst}})}$ with $\delta_{\mathsf{clyst}} = 1 - ((1 - \varepsilon_1) + \varepsilon_1/c) \cdot \frac{1-\delta}{1-\varepsilon_2}$.

In Section 6, we prove the theorem by overcoming the challenges listed above it. We note that our result relating time-independent point games and time-dependent point games (in the cheat-penalised setting) assumes the point game is symmetric (i.e., $h = v^T$) and $\Lambda > 0$ because otherwise some of the analysis gets more involved. We leave the relaxation of these assumptions to future work.

2.3 Good TIPGs and how to find them

Even using all the point game machinery discussed so far, and its extension to the cheat-penalised setting, it is unclear how to design good point games. Any point game for weak coin-flipping can be immediately lifted to the cheat-penalised setting—simply because the constraints on "valid move" get weakened as one moves farther away from the origin. Indeed, this was noted by Mochon [Moc07] and he gave a heuristic calculation for how it might apply to his Dip-Dip-Boom protocol. This protocol for (non-cheat-penalised) weak coin-flipping has bias $1/6$. Mochon's informal calculations suggest that in the cheat-penalised setting, its bias vanishes as the cheating penalty grows.

2.3.1 Analytic TIPGs

To find analytic TIPGS, we start by looking at the Spekkens-Rudolph protocol [SR02] (the simplest instance of Dip-Dip-Boom) in the cheat-penalised setting in Section 9.1. We then look at the Dip-Dip-Boom protocol. We provide a careful derivation of the fact that the bias vanishes as the cheating penalty grows in Section 9.2. While Mochon's analysis established vanishing bias to order $O(1/\Lambda)$, we extend this result by deriving the higher order terms as well. Besides Mochon's analysis, we also look at cheat-penalised versions of other protocols in the literature.

While a promising start, it was unclear how or whether one can build on these to have efficient protocols with low bias and small penalty. For instance, Mochon's family of TIPGs that approach zero bias—of which Dip-Dip-Boom is the simplest instance—are insensitive to the penalty parameter. More concretely, the polynomial-based technique that Mochon uses to assign weights to the points in his TIPGs, depends only on the relative positions of the points under consideration—and therefore remain unchanged under translations (i.e., the cheating penalty parameter). On the other hand, doing a brute-force numerical search quickly becomes untenable as we explain below. Numerical searches were attempted already for (non-cheat-penalised) weak coin-flipping [NST14] with no luck until the question was resolved via other techniques [Moc07, ARW19, ARV21].

2.3.2 Numerical Algorithm for finding TIPGs

To overcome these challenges, we introduce a numerical algorithm that harnesses the structure underlying point games and works generically to solve **(time-independent) point game problems**—given a start-configuration s and an end-configuration e, find horizontally valid and vertically valid moves (h, v) such that

$$h + v = e - s.$$

We begin with the concept of a bivariate *profile function* from [Mil20]. For any two-dimensional move p, we define an associated profile function $\hat{p}\colon \mathbb{R}^2 \to \mathbb{R}$. Deferring a formal definition to Section 7.1, informally, for $\lambda, \gamma > 0$, we have

$$\hat{p}(\lambda, \gamma) \quad := \quad \sum_{x,y \geq 0} \left(\frac{\lambda x}{\lambda + x}\right) \left(\frac{\gamma y}{\gamma + y}\right) p(x, y). \tag{13}$$

The definition is essentially a restatement of the conditions used to define valid moves (see Section 2.2). Any horizontally or vertically valid function must have a non-negative profile function. As a consequence, a point game problem (s, e) can only be solved if $\hat{e} \geq \hat{s}$.

In [Mil20], the profile function was used to prove the impossibility of constructing weak coin-flipping protocols with low communication complexity. In Section 7, we show how the same concept can be used to *find* good point games. Below, we provide a four-step description of the algorithm, together with the intuition behind it.

1. Discretisation Recall that we want to find a pair (h, v) that solves the *point game problem* for a given s and e. A brute-force search will have to look through the set of all possible pairs (h, v) which is an infinite-dimensional space that is described by an infinite number of linear constraints. Concretely, h and v must satisfy the validity conditions from Equations 7 and 8. Instead, we can choose a discretisation of the plane for the coordinates x and y, together with a maximum and a minimum coordinate. This defines a finite set $S \subseteq \mathbb{R}_{\geq 0}$ which includes the coordinates appearing in the start-configuration s and end-configuration e. Since our search will require the use of the profile functions of h and v, we also discretise the parameter space for the profile functions, obtaining a finite set $T \subseteq \mathbb{R}_{\geq 0}$. Finally, we fix a threshold parameter $\delta \in \mathbb{R}_{>0}$ that will be used in the next step. In summary, the appropriate choice of the sets S and T together with the parameter δ is the first step of the algorithm.

2. Profile Matching We search for a move h that is horizontally valid (when restricted to T), such that the Euclidean norm

$$\left\| (\hat{h} + \hat{v}) - (\hat{e} - \hat{s}) \right\|$$

is small and $v = h^\top$. Since T is finite, the profile functions $\hat{h}, \hat{v}, \hat{e}$, and $\hat{s}$ can all be represented as finite-dimensional vectors.

We restrict the search for (h, v) to a smaller subspace V defined as follows. Consider the matrix M mapping moves with coordinates in S to profiles with coordinates in T. Let V be the span of right singular vectors of M with singular values greater than the threshold parameter δ (recall δ was fixed in the previous step). Restricting the search to V ensures that we only retain those components of a move that contribute the most to their profile function. This not only reduces the complexity of the search but also, crucially, keeps the norm of (h, v) small.

3. Configuration Matching Denote by (h, v) the pair of moves produced by the previous step. This is such that the profile $\hat{h} + \hat{v}$ is close to $\hat{e} - \hat{s}$. That, in turn, means that $(e - s) - (h + v) = t$ has a small profile. It turns out that this fact helps with constructing approximately valid moves.

Definition 5. A function $g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is *η-approximately valid* with $\eta > 0$, if

$$\sum_x g(x) + \eta \geq 0, \tag{14}$$

$$\sum_x g(x) \left(\frac{\lambda x}{x + \lambda} \right) + \eta \geq 0. \tag{15}$$

The definition extends naturally to approximately horizontally and vertically valid moves.

Our goal in this step, is to find moves p and q such that $(h + p) + (v + q) = e - s$ and p and q are approximately horizontally and vertically valid, respectively. To find such a pair (p, q), we decompose t into two parts where one part is approximately horizontally valid while the other is approximately vertically valid. It is not too hard to see that such a decomposition is guaranteed because t has a small profile. In fact, given that $h = v^\top$, one can also ensure that $p = q^\top$. Finally, while adding p and q increases the norm of the moves, this is unavoidable if one wants to match $e - s$ exactly.

4. Validity Enforcement So far, we have approximately valid functions (h', v') where $h' = h + p$ and $v' = v + q$ such that $h' + v' = e - s$. In this final step, we want the opposite trade-off: we want $h' + v'$ to be approximately equal to $e - s$ while ensuring h' and v' are exactly valid. To achieve this, we project h' onto the closest (exactly) valid vector h_*. This turns out to be a quadratic program that can be solved efficiently. Since h_* was close to h' (and $v' = h'^\top$), it follows that $h_* + v_*$ is close to $e - s$ as desired.

The algorithm above results in approximate penTIPGs and in the previous section we saw how to map precisely such penTIPGs into cheat-penalised WCF protocols. By running this algorithm numerically, we find several TIPGS with good parameters. We describe two of these in Theorem 6 below. These penTIPGs were used to construct the protocols described in the Main Result box. See Section 8.1 for an explicit description of these TIPGs and our GitHub page [AMMS25] for a Mathematica implementation of our algorithm.

Theorem 6 (Existence of $(\Lambda, \varepsilon_{\mathsf{approx}})$-penTIPG (from Section 8.1))**.** *There exist* $(\Lambda, \varepsilon_{\mathsf{approx}})$*-penTIPGs with starting point* $s = \frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!]$*, final point* $e = [\![\Lambda + \frac{1}{2} + \epsilon, \Lambda + \frac{1}{2} + \epsilon]\!]$ *and a grid of* 64 *points, with the following parameters:*

- $\Lambda = 1$, $\varepsilon_{\mathsf{approx}} = 10^{-18}$ *and* $\epsilon = 5 \times 10^{-3}$.
- $\Lambda = 0.01$, $\varepsilon_{\mathsf{approx}} = 10^{-18}$ *and* $\epsilon = 10^{-14}$.

3 The Primal and Dual Formulation of penWCF

We start by defining what it means to be a Λ-penalty weak coin flipping protocol formally.

Definition 7 (Λ-penWCF protocol with bias ϵ)**.** For n even, an n-message Λ-penalty weak coin-flipping protocol (Λ-penWCF protocol) between two parties, Alice and Bob, is described by:

- three Hilbert spaces $\mathcal{A}, \mathcal{B}$ corresponding to Alice's and Bob's private workspaces and a message space $\mathcal{M}$;
- an initial product state $|\psi_0\rangle := |\psi_{A,0}\rangle \otimes |\psi_{M,0}\rangle \otimes |\psi_{B,0}\rangle \in \mathcal{A} \otimes \mathcal{M} \otimes \mathcal{B}$;
- a set of n unitaries $\{U_1, \ldots, U_n\}$ acting on $\mathcal{A} \otimes \mathcal{B} \otimes \mathcal{C}$ with $U_i = U_{A,i} \otimes \mathbb{I}_{\mathcal{B}}$ for odd i and $U_i = \mathbb{I}_{\mathcal{A}} \otimes U_{B,i}$ for even i;
- a set of honest states $\{|\psi_0\rangle, \ldots, |\psi_n\rangle\}$ defined by $|\psi_i\rangle := U_i U_{i-1} \cdots U_1 |\psi_0\rangle$;
- a set of n projectors $\{E_1, \ldots, E_n\}$ acting on $\mathcal{A} \otimes \mathcal{M} \otimes \mathcal{B}$ with $E_i = E_{A,i} \otimes \mathbb{I}_{\mathcal{B}}$ for i odd, and $E_i = \mathbb{I}_{\mathcal{A}} \otimes E_{B,i}$ for i even, such that $E_i |\psi_i\rangle = |\psi_i\rangle$;
- two final POVMs $\left\{\Pi_A^{(0)}, \Pi_A^{(1)}, \Pi_A^{(\perp)}\right\}$ acting on $\mathcal{A}$ and $\{\Pi_B^{(0)}, \Pi_B^{(1)}, \Pi_B^{(\perp)}\}$ acting on $\mathcal{B}$ where the superscripts $(0), (1)$ denote outcome $0, 1$, respectively, while the superscript $(\perp)$ denotes the outcome $\perp$ which denotes "cheat detection".

The penWCF protocol proceeds as follows:

- In the beginning, Alice holds $|\psi_{A,0}\rangle \otimes |\psi_{M,0}\rangle$ and Bob holds $|\psi_{B,0}\rangle$.
- For $i = 1$ to n:
  - If i is odd, Alice measures the incoming state with the POVM $\{E_{i-1}, \mathbb{I} - E_{i-1}\}$. On the first outcome, she applies U_i and sends the message qubits to Bob; on the second outcome, she ends the protocol by outputting $\perp$ (i.e., Alice declares Bob cheated).
  - If i is even, Bob measures the incoming state with the POVM $\{E_{i-1}, \mathbb{I} - E_{i-1}\}$. On the first outcome, he applies U_i and sends the message qubits to Alice; on the second outcome, he ends the protocol by outputting $\perp$ (i.e., Bob declares himself to be the winner).
  - Alice and Bob measure their part of the state with the final POVM and output the outcome of their measurements.
- Points allocation:
  - If the output is 0, Alice gets $\Lambda + 1$ points, Bob gets Λ points.
  - If the output is 1, Bob gets $\Lambda + 1$ points, Alice gets Λ points.
  - If Alice (resp. Bob) outputs $\perp$, then Bob (resp. Alice) gets 0 points while Alice (resp. Bob) gets Λ points.

We require that a Λ-penWCF protocol satisfy the following properties:

- Correctness: When both players are honest, Alice and Bob never output $\perp$ and their outcomes are identical, i.e. $\Pi_{\mathcal{A}}^{(0)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(1)} |\psi_n\rangle = \Pi_{\mathcal{A}}^{(1)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(1)} |\psi_n\rangle = 0$ and for all $x \in \{0, 1\}$, $\Pi_{\mathcal{A}}^{(\perp)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(x)} |\psi_n\rangle = \Pi_{\mathcal{A}}^{(x)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(\perp)} |\psi_n\rangle = 0$.

- Balanced: When both players are honest, they each win with probability $1/2$:

$$P_A = \left\| \Pi_{\mathcal{A}}^{(0)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(0)} \left| \psi_n \right\rangle \right\|^2 = \frac{1}{2}$$

  and

$$P_B = \left\| \Pi_{A}^{(1)} \otimes \mathbb{I}_{\mathcal{M}} \otimes \Pi_{\mathcal{B}}^{(1)} \left| \psi_n \right\rangle \right\|^2 = \frac{1}{2}.$$

- ϵ bias:
  - Notation:
    - Denote by R_A^* (resp. R_B^*) the expected number of points Alice (resp. Bob) obtains, maximised over all cheating strategies for Alice (resp. Bob) and call it "cheating rewards" for Alice (resp. Bob).
    - Define $P_A^* := R_A^* - \Lambda$ and similarly $P_B^* := R_B^* - \Lambda$ and call them "cheating probabilities" for Alice and Bob respectively.
  - We say the protocol has bias ϵ if $\max\{P_A^*, P_B^*\} \leq \frac{1}{2} + \epsilon$.

Remark 8 (P_A^*, P_B^* can be viewed as probabilities). From correctness, it follows that $P_A^*, P_B^* \geq 0$ are both non-negative. Specifically, this is because correctness guarantees that on an honest execution, $\perp$ never occurs which means that an honest strategy already yields Λ points and that, together with the definitions of $R_A^*, P_A^*, R_B^*, P_B^*$, establishes $P_A^*, P_B^* \geq 0$. From the definition of the problem, it is also clear that a malicious party can gain at most $\Lambda + 1$ points, thus, $P_A^*, P_B^* \leq 1$. Thus, we can view formally P_A^*, P_B^* as "cheating probabilities".

Remark 9 (Λ-penWCF without intermediate projectors). One can defer the intermediate measurements in Definition 7 above to the very end, without changing the security of the protocol. We add projectors for convenience later.

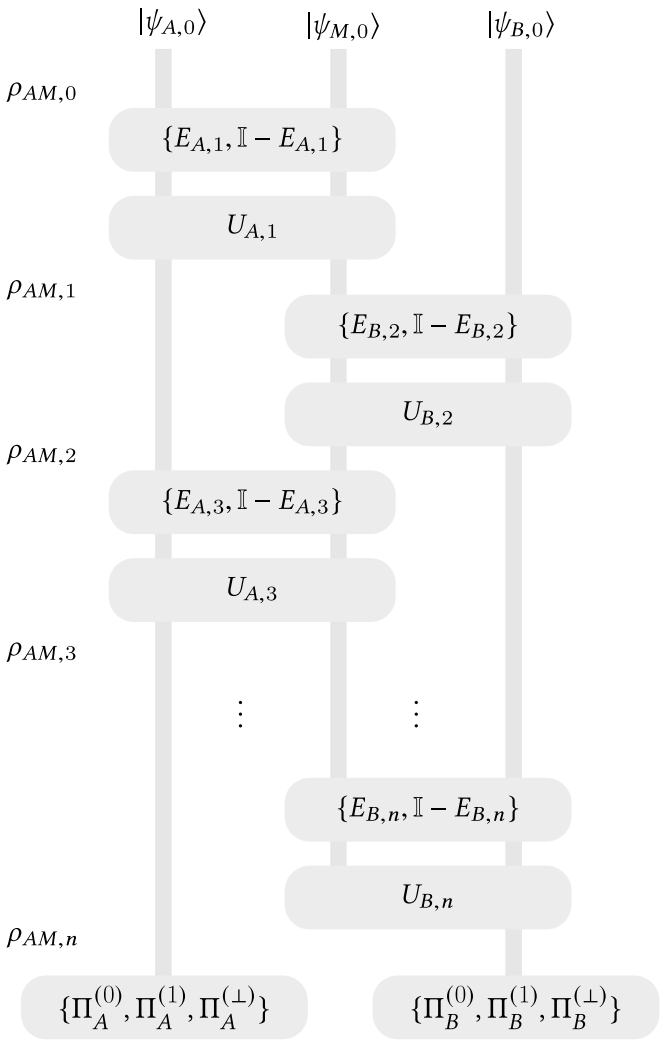


Figure 5: Schematic of a cheat-penalised weak coin-flipping protocol.

We now cast R_A^* and R_B^* as SDPs. This formulation enables us to relate point games to protocols for Λ-penWCF.

Theorem 10 (Primal SDP)**.** *It holds that*

$$R_B^* = \max \Big[(\Lambda + 1) \cdot \mathsf{tr}(\Pi_A^{(1)} \otimes \mathbb{I}_M \rho_{AM,n}) + \Lambda \cdot \mathsf{tr}(\Pi_A^{(0)} \otimes \mathbb{I}_M \rho_{AM,n}) \\ \cancel{+0 \cdot \mathsf{tr}(\Pi_A^{(\perp)} \otimes \mathbb{I}_M \rho_{AM,n})} \Big]$$

over all $\rho_{AM,i}$ satisfying the constraints

- $\mathsf{tr}_M(\rho_{AM,0}) = \mathsf{tr}_{MB}(|\psi_0\rangle\langle\psi_0|) = |\psi_{A,0}\rangle\langle\psi_{A,0}|$
- *for i odd,* $\mathsf{tr}_M(\rho_{AM,i}) = \mathsf{tr}_M(U_i E_i \rho_{AM,i-1} E_i U_i^\dagger)$
- *for i even,* $\mathsf{tr}_M(\rho_{AM,i}) = \mathsf{tr}_M(\rho_{AM,i-1})$.

Similarly, it holds that

$$R_A^* = \max\Big[(\Lambda+1)\cdot \mathsf{tr}(\Pi_B^{(0)}\otimes \mathbb{I}_M \rho_{BM,n}) + \Lambda \cdot \mathsf{tr}(\Pi_B^{(1)}\otimes \mathbb{I}_M \rho_{BM,n}) \\ \cancel{+0\cdot \mathsf{tr}(\Pi_B^{(\perp)}\otimes \mathbb{I}_M \rho_{BM,n})}\Big]$$

over all $\rho_{BM,i}$ satisfying the following constraints

- $\mathsf{tr}_M(\rho_{MB,0}) = \mathsf{tr}_{AM}(|\psi_0\rangle\langle\psi_0|) = |\psi_{B,0}\rangle\langle\psi_{B,0}|$
- *for i even,* $\mathsf{tr}_M(\rho_{MB,i}) = \mathsf{tr}_M(U_i E_i \rho_{MB,i-1} E_i U_i^\dagger)$
- *for i odd,* $\mathsf{tr}_M(\rho_{MB,i}) = \mathsf{tr}_M(\rho_{MB,i-1})$.

Proof. This is essentially the same as WCF (as in [ACG+14]) except that we changed the objective function to reflect the points a player gains on average (and also, we apply projections before instead of after). □

Remark 11. Note here that using the translated game already helps simplify the notation. For instance, since the outcome corresponding to $\mathbb{I} - E_i$ results in 0 points being awarded to the malicious party, one need not include the support of $U_i \rho_{AM,i-1} U_i^\dagger$ on $\mathbb{I} - E_i$ when considering the primal SDP for Bob, for instance.

Next, we consider the dual SDP.

Theorem 12 (Dual SDP). *It holds that*
$R_B^* = \min[\mathsf{tr}(Z_{A,0}|\psi_{A,0}\rangle\langle\psi_{A,0}|)]$ *over all dual variables $Z_{A,i}$ under the following dual constraints*

1. $\forall i\ Z_{A,i} \geq 0$
2. *for i odd,* $Z_{A,i-1}\otimes \mathbb{I}_M \geq E_{A,i-1} U_{A,i}^\dagger (Z_{A,i}\otimes \mathbb{I}_M) U_{A,i} E_{A,i-1}$
3. *for i even,* $Z_{A,i-1} = Z_{A,i}$
4. $Z_{A,n} = \Lambda\cdot\Pi_A^{(0)} + (\Lambda+1)\cdot\Pi_A^{(1)}$.

$R_A^* = \min[\mathsf{tr}(Z_{B,0}|\psi_{B,0}\rangle\langle\psi_{B,0}|)]$ *over all dual variables $Z_{B,i}$ satisfying the following dual constraints*

1. $\forall i,\ Z_{B,i} \geq 0$
2. *for i even,* $\mathbb{I}_M \otimes Z_{B,i-1} \geq E_{B,i-1} U_{B,i}^\dagger (\mathbb{I}_M \otimes Z_{B,i}) U_{B,i} E_{B,i-1}$,
3. *for i odd,* $Z_{B,i-1} = Z_{B,i}$
4. $Z_{B,n} = \Lambda\cdot\Pi_B^{(1)} + (\Lambda+1)\cdot\Pi_B^{(0)}$.

The collection of matrices $(Z_{A,0},\dots Z_{A,n})$ and $(Z_{B,0},\dots Z_{B,n})$ satisfying the constraints above, and also the following constraint, are defined to be dual feasible points.

5. *$|\psi_{A,0}\rangle$ is an eigenvector of $Z_{A,0}$ with eigenvalue $\beta > 0$ and $|\psi_{B,0}\rangle$ is an eigenvector of $Z_{B,0}$ with eigenvalue $\alpha > 0$.*

It holds that

$$R_B^* = \inf\alpha \text{ and } R_A^* = \inf\beta \tag{16}$$

where the infimum is over all dual feasible points (with α,β are as in condition 5 above).

Proof. Using the standard weak duality argument, one can check that indeed the dual is as stated (see for instance Appendix B of Mochon's WCF paper)—except for condition 5.

We focus on proving Equation 16. We show $R_B^* = \inf \alpha$ (the proof for $R_A^* = \inf \beta$ is analogous). Suppose we are given an optimal solution $(Z_{A,i})_{i=1}^n$ to the dual SDP satisfying the first four constraint. This, in particular, entails that $R_B^* = \langle \psi_{A,0}| Z_{A,0} |\psi_{A,0}\rangle$. If $|\psi_{A,0}\rangle$ were an eigenvector of $Z_{A,0}$ then there is nothing left to prove. Let us assume the contrary. If we can construct another operator $Z'_{A,0}$ such that (1) $Z'_{A,0} \geq Z_{A,0}$ and (2) $Z'_{A,0} |\psi_{A,0}\rangle = (R_B^* + \epsilon) |\psi_{A,0}\rangle$ where $\epsilon > 0$ can be made arbitrarily small, then we would have proven Equation 16. This is because we could use the matrices $(Z'_{A,0}, Z_{A,1} \dots Z_{A,n})$ as a feasible point to the dual SDP whose objective $\mathsf{tr}(Z'_{A,0} |\psi_{A,0}\rangle \langle\psi_{A,0}|)$ equals R_B^* in the limit $\epsilon \to 0$.

We drop the subscript $A, 0$ henceforth for clarity, viz. we use $(Z, Z', |\psi\rangle)$ instead of $(Z_{A,0}, Z'_{A,0}, |\psi_{A,0}\rangle)$. We show that for any $\epsilon > 0$, there is a $\mathsf{val} \in \mathbb{R}_{\geq 0}$ such that

$$Z' := (\langle\psi| Z |\psi\rangle + \epsilon) |\psi\rangle \langle\psi| + \mathsf{val}(\mathbb{I} - |\psi\rangle \langle\psi|) \tag{17}$$

satisfies the aforementioned requirements (1) and (2). In fact, requirement (2) is satisfied by construction. We impose requirement (1) to obtain val as follows: we require that for all normalised vectors $|\phi\rangle \in \mathcal{A}$, it holds that

$$\langle\phi| \left(Z' - Z\right) |\phi\rangle = \left[(R_B^* + \epsilon) |\langle\phi|\psi\rangle|^2 + \mathsf{val} \cdot \left(1 - |\langle\phi|\psi\rangle|^2\right)\right] - \langle\phi| Z |\phi\rangle \geq 0.$$

One can write $|\phi\rangle = a |\psi\rangle + \tilde{a} \left|\psi^\perp\right\rangle$ where a' is real (the phase can be absorbed in $\left|\psi^\perp\right\rangle$) satisfying $|a|^2 + \tilde{a}^2 = 1$ and $\left\langle\psi|\psi^\perp\right\rangle = 0$. Using this, one has

$$\langle\phi| Z |\phi\rangle = |a|^2 \cancel{\langle\psi| Z |\psi\rangle} \xrightarrow{R_B^*} + \tilde{a}^2 \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle + \tilde{a} \left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right).$$

Substituting this in the previous equation, one obtains

$$\begin{aligned}
\langle\phi| \left(Z' - Z\right) |\phi\rangle &= (R_B^* + \epsilon) |a|^2 + \mathsf{val} \cdot \cancel{(1 - |a|^2)} \xrightarrow{\tilde{a}^2} \\
&\quad - |a|^2 R_B^* - \tilde{a}^2 \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle - \tilde{a} \left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right) \\
&= \tilde{a}^2 \left(\mathsf{val} - \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle\right) - \tilde{a} \left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right) + |a|^2 \epsilon \geq 0.
\end{aligned}$$

If $|\phi\rangle$ is such that $a = 0$, then one can simply pick any $\mathsf{val} \geq \|Z\|$ (where $\|Z\|$ is the operator norm of Z, i.e. the highest eigenvalue of Z in this case). But this may not be enough otherwise. When $a \neq 0$, one can view the expression above as a quadratic in $\tilde{a}$. We take $\mathsf{val} \geq \|Z\|$ so that the coefficient of $\tilde{a}^2$ is positive. Now, if we can ensure that the quadratic has no roots then the inequality is guaranteed to hold. To this end, we require that the discriminant is negative. This yields

$$\begin{aligned}
&\left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right)^2 - 4|a|^2 \epsilon \left(\mathsf{val} - \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle\right) \leq 0 \\
\iff \qquad &\frac{1}{4|a|^2\epsilon} \left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right)^2 + \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle \leq \mathsf{val}.
\end{aligned}$$

Using the fact that $z + \mathsf{h.c.} = 2\Re(z) \leq 2|z|$ for any complex number z, we have that

$$\frac{1}{4|a|^2\epsilon} \left(a \left\langle\psi^\perp\right| Z |\psi\rangle + \mathsf{h.c.}\right)^2 \leq \frac{\cancel{4|a|^2}}{\cancel{4|a|^2}\epsilon} \|Z\|$$

which means that it suffices to require that val satisfies

$$\frac{\|Z\|}{\epsilon} \leq \frac{\|Z\|}{\epsilon} + \left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle \leq \mathsf{val}$$

to ensure the determinant is negative. Using $\left\langle\psi^\perp\right| Z \left|\psi^\perp\right\rangle \leq \|Z\|$, we conclude that it suffices to set $\mathsf{val} = \max\left\{\left(\frac{1}{\epsilon} + 1\right) \|Z\|, 2\|Z\|\right\}$.

One can proceed analogously for the R_A^* case. □

4 Λ-Penalty EBM Point Games

Following [ACG+14], we define EBM point games as a step towards time-dependent point games.

Definition 13 (Prob)**.** Consider $Z \geq 0$ and let $\Pi^{[z]}$ represent the projector on the eigenspace of eigenvalue $z \in \mathsf{spectrum}(Z)$. We have $Z = \sum_z z\Pi^{[z]}$. Let $|\psi\rangle$ be a (not necessarily normalised) vector. We define the function with finite support $\mathsf{Prob}[Z, \psi] := [0, \infty) \to [0, \infty)$ as

$$\mathsf{Prob}[Z, \psi](z) = \begin{cases} \langle\psi| \Pi^{[z]} |\psi\rangle & \text{if } z \in \mathsf{spectrum}(Z) \\ 0 & \text{else.} \end{cases}$$

If $Z = Z_A \otimes \mathbb{I}_M \otimes Z_B$, using the same notation, we define the bivariate function with finite support $\mathsf{Prob}[Z_A, Z_B, \psi] : [0, \infty) \times [0, \infty) \to [0, \infty)$ as

$$\mathsf{Prob}[Z_A, Z_B, \psi](z_A, z_B) := \begin{cases} \langle\psi| \Pi^{[z_A]} \otimes \mathbb{I}_M \otimes \Pi^{[z_B]} |\psi\rangle & \text{if } (z_A, z_B) \in \mathsf{spectrum}(Z_A) \times \mathsf{spectrum}(Z_B) \\ 0 & \text{else.} \end{cases}$$

Definition 14 (EBM line transition)**.** Let $g, h : [0, \infty) \to [0, \infty)$ be two functions with finite supports. The line transition $g \to h$ is *expressible by matrices (EBM)* if there exist two matrices $0 \leq G \leq H$ and a (not necessarily normalised) vector $|\psi\rangle$ such that $g = \mathsf{Prob}[G, |\psi\rangle]$ and $h = \mathsf{Prob}[H, |\psi\rangle]$. We say $g \to h$ above is *EBM with spectrum in* $[a, b]$ if $\mathsf{spectrum}(G) \cup \mathsf{spectrum}(H) \subseteq [a, b]$.

Definition 15 (EBM transition)**.** Let $p, q : [0, \infty) \times [0, \infty) \to [0, \infty)$ be two functions with finite supports. The transition $p \to q$ is an:

- EBM horizontal transition if for all $y \in [0, \infty)$, $p(\cdot, y) \to q(\cdot, y)$ is an EBM line transition, and
- EBM vertical transition if for all $x \in [0, \infty)$, $p(x, \cdot) \to q(x, \cdot)$ is an EBM line transition.

The previous definitions were the same as in the standard WCF setting. Now, we introduce the cheat-penalised version of the EBM point games. The only change is in the starting and final frames.

Definition 16 (Λ-penEBM point game)**.** A Λ-penalty EBM point game is a sequence of functions $(p_0, \ldots, p_n)$ with finite support such that

- $p_0 = \frac{1}{2}[\![\Lambda, \Lambda + 1]\!] + \frac{1}{2}[\![\Lambda + 1, \Lambda]\!]$,
- for all even i, $p_i \to p_{i+1}$ is an EBM vertical transition,
- for all odd i, $p_i \to p_{i+1}$ is an EBM horizontal transition,
- $p_n = 1\,[\![\beta, \alpha]\!]$ for some $\alpha, \beta \in [\Lambda, \Lambda + 1]$. We call $[\![\beta, \alpha]\!]$ the final point of the EBM point game.

We sometimes refer to the functions p_i as frames or configurations. Next, we show the relation between Λ-penWCF protocols and Λ-penEBM point games.

4.1 Λ-penWCF protocol implies Λ-penEBM point game

Proposition 17 (Λ-penWCF $\implies$ Λ-penEBM point game)**.** *Given*

- *a Λ-penWCF protocol with cheating rewards R_A^* and R_B^* (see Definition 7), and*
- *any positive number $\delta > 0$,*

there exists a Λ-penEBM point game with final point $[\![R_A^* + \delta, R_B^* + \delta]\!]$.

Remark 18. The proof of this proposition closely follows that of the analogous statement for (standard) WCF, although we note and correct a minor error in the argument presented in [ACG+14]. The main difference from the standard case is in the boundary conditions.

Proof. Assume we are given a Λ-penWCF protocol (with no intermediate measurements as justified in Remark 9), together with the dual certificates $\{Z_{A,i}\}_{i=1}^{n}$ and $\{Z_{B,i}\}_{i=1}^{n}$ (see Theorem 12) witnessing cheating points R_A^* and R_B^*. Using the proof of the last part of Theorem 12, we can use $Z'_{A,0}$ (resp. $Z'_{B,0}$) as in Equation 17 instead of $Z_{A,0}$ (resp. $Z_{B,0}$) which admits $|\psi_{A,0}\rangle$ (resp. $|\psi_{B,0}\rangle$) as eigenvectors with eigenvalues $\beta = R_B^* + \delta$ (resp. $\alpha = R_A^* + \delta$).

We use the "reversed time" convention: $Z_A^{(i)} := Z_{A,n-i}$, $Z_B^{(i)} := Z_{B,n-i}$ and $\left|\psi^{(i)}\right\rangle := |\psi_{n-i}\rangle$ for $i \in \{0,\dots,n\}$ (note that the notion conflicts slightly; we already used $\Pi_A^{(0/1/\perp)}$ (resp. $\Pi_B^{(0/1/\perp)}$) to denote Alice's (resp. Bob's) final POVM measurements). We use this notation also for $U_{A,i}$ and $U_{B,i}$.

With the notation in place, we start the proof by defining the first and last frames. We continue writing the projectors $E_{A,i}, E_{B,i}$ for illustrating where they make a difference.

First frame. Define $p_0 := \mathsf{Prob}[Z_A^{(0)}, Z_B^{(0)}, \left|\psi^{(0)}\right\rangle]$. Consider $Z_A^{(0)} \otimes \mathbb{I}_M \otimes Z_B^{(0)}$ and the state $\left|\psi^{(0)}\right\rangle$. Recall (see Theorem 12) that $Z_A^{(0)} = \Lambda \cdot \Pi_A^{(0)} + (\Lambda+1)\cdot\Pi_A^{(1)}$ and $Z_B^{(0)} = \Lambda \cdot \Pi_B^{(1)} + (\Lambda+1)\cdot\Pi_B^{(0)}$. Recall also that by definition of a Λ-penWCF protocol (see Definition 7), it holds that

$$\mathsf{tr}\left(\Pi_A^{(1)} \otimes \mathbb{I}_M \otimes \Pi_B^{(1)} \left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|\right) = \mathsf{tr}\left(\Pi_A^{(0)} \otimes \mathbb{I}_M \otimes \Pi_B^{(0)} \left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|\right) = \frac{1}{2} \qquad \text{(Balanced)}$$

$$\mathsf{tr}\left(\Pi_A^{(a)} \otimes \mathbb{I}_M \otimes \Pi_B^{(b)} \left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|\right) = 0 \quad \forall (a,b) \in S_{\text{invalid}} \qquad \text{(Correctness)}$$

where $S_{\text{invalid}} := \{(0,1),(1,0),(\perp,0),(\perp,1),(0,\perp),(1,\perp),(\perp,\perp)\}$ captures events where the honest parties either output abort or disagree on the output. One can thus write

$$p_0 = \mathsf{Prob}\left[(\Lambda+1)\cdot\Pi_A^{(1)} + \Lambda\cdot\Pi_A^{(0)} + 0\cdot\Pi_A^{(\perp)}, (\Lambda+1)\cdot\Pi_B^{(0)} + \Lambda\cdot\Pi_B^{(1)} + 0\cdot\Pi_B^{(\perp)}, \left|\psi^{(0)}\right\rangle\right].$$

Note that one can also write

$$p_0 = \sum_{(x,y)\in\{\Lambda+1,\Lambda,0\}\times\{\Lambda+1,\Lambda,0\}} \mathsf{coeff}_{xy}\,[\![x,y]\!]$$

for some choice of $\mathsf{coeff}_{xy} \in \mathbb{R}$ for each x, y. Further, we have

$$\begin{aligned} p_0 &= \mathsf{tr}\left(\Pi_A^{(1)} \otimes \mathbb{I}_M \otimes \Pi_B^{(1)} \cdot \psi^{(0)}\right)[\![\Lambda+1,\Lambda]\!] + \\ &\quad \mathsf{tr}\left(\Pi_A^{(0)} \otimes \mathbb{I}_M \otimes \Pi_B^{(0)} \cdot \psi^{(0)}\right)[\![\Lambda,\Lambda+1]\!] && \text{(other } \mathsf{coeff}_{xy} = 0\text{; Correctness)} \\ &= \frac{1}{2}\cdot[\![\Lambda+1,\Lambda]\!] + \frac{1}{2}\cdot[\![\Lambda,\Lambda+1]\!] && \text{(Balanced)} \end{aligned}$$

where we use $\psi^{(0)}$ to denote $\left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|$.

Last frame. Now, define $p_n := \mathsf{Prob}\left[Z_A^{(n)}, Z_B^{(n)}, \left|\psi^{(n)}\right\rangle\right]$. Consider $Z_A^{(n)} \otimes \mathbb{I}_M \otimes Z_B^{(n)}$ and $\left|\psi^{(n)}\right\rangle$. Recall that $\left|\psi^{(n)}\right\rangle = |\psi_{A,0}\rangle \otimes |\psi_{M,0}\rangle \otimes |\psi_{B,0}\rangle$ (see Definition 7) and as stated at the beginning, we can take $Z_A^{(n)} = \beta\left|\psi_A^{(n)}\right\rangle\left\langle\psi_A^{(n)}\right| + \mathsf{val}_A\left(\mathbb{I} - \left|\psi_A^{(n)}\right\rangle\left\langle\psi_A^{(n)}\right|\right)$, $Z_B^{(n)} = \alpha\left|\psi_B^{(n)}\right\rangle\left\langle\psi_B^{(n)}\right| + \mathsf{val}_B\left(\mathbb{I} - \left|\psi_B^{(n)}\right\rangle\left\langle\psi_B^{(n)}\right|\right)$ (using Equation 17). Since $\left|\psi_A^{(n)}\right\rangle$ and $\left|\psi_B^{(n)}\right\rangle$ are normalised vectors, together these imply that $p_n = 1\cdot[\![\beta,\alpha]\!]$.

Intermediate frames. We now define the remaining frames p_i as

$$p_i := \mathsf{Prob}\left[Z_A^{(i)}, Z_B^{(i)}, \left|\psi^{(i)}\right\rangle\right] = \sum_{(x,y)\in\mathsf{spectrum}(Z_A^{(i)})\times\mathsf{spectrum}(Z_B^{(i)})} \mathsf{tr}\left[\mathsf{Proj}(Z_A^{(i)},x)\otimes\mathbb{I}_M\otimes\mathsf{Proj}(Z_B^{(i)},y)\cdot\psi^{(i)}\right][\![x,y]\!] \tag{18}$$

where $\mathsf{Proj}(Z_A^{(i)}, x)$ is a projector on the x-eigenvalue subspace of $Z_A^{(i)}$ ($\mathsf{Proj}(Z_B^{(i)}, y)$ is defined analogously) and $\psi^{(i)}$ denotes $\left|\psi^{(i)}\right\rangle\left\langle\psi^{(i)}\right|$. We now show that $p_i \to p_{i+1}$ is an EBM transition (either horizontal or vertical, depending on i). This part of the proof is exactly the same as in the standard WCF case. Suppose i is such that $Z_B^{(i+1)} = Z_B^{(i)}$ while

$$Z_A^{(i+1)} \otimes \mathbb{I}_M \geq U_A^{(i)\dagger}\left(Z_A^{(i)} \otimes \mathbb{I}_M\right) U_A^{(i)} \tag{19}$$

(this depends on whether $n-i$ is odd or even; in the other case, $Z_A^{(i+1)} = Z_A^{(i)}$ while $Z_B^{(i+1)}$ will satisfy an inequality). It also holds that $\left|\psi^{(i+1)}\right\rangle = U^{(i)\dagger}\left|\psi^{(i)}\right\rangle$. We note that

$$\begin{aligned}
p_i(\cdot, y) &= \sum_{x \in \mathsf{spectrum}(Z_A^{(i)})} \mathsf{tr}[\mathsf{Proj}(Z_A^{(i)}, x) \otimes \mathbb{I}_M \otimes \mathsf{Proj}(Z_B^{(i)}, y) \cdot \psi^{(i)}]\, [\![x]\!] \\
&= \mathsf{Prob}\left[Z_A^{(i)} \otimes \mathbb{I}_M \otimes \mathsf{Proj}(Z_B^{(i)}, y), \left|\psi^{(i)}\right\rangle\right] && \text{using the Definition of Prob} \\
&= \mathsf{Prob}\left[U_A^{(i)\dagger}\left(Z_A^{(i)} \otimes \mathbb{I}_M\right) U_A^{(i)} \otimes \mathsf{Proj}(Z_B^{(i)}, y), U^{(i)\dagger}\left|\psi^{(i)}\right\rangle\right] && \because \text{the unitaries cancel and} \\
& && \mathsf{spectrum}(Z) = \mathsf{spectrum}(U^\dagger Z U) \qquad (20) \\
&= \mathsf{Prob}[G_y, \left|\psi^{(i+1)}\right\rangle]
\end{aligned}$$

where $G_y := U_A^{(i)\dagger}\left(Z_A^{(i)} \otimes \mathbb{I}_M\right) U_A^{(i)} \otimes \mathsf{Proj}(Z_B^{(i+1)}, y)$ because $Z_A^{(i+1)} = Z_A^{(i)}$.[6] Similarly, one can write

$$\begin{aligned}
p_{i+1}(\cdot, y) &= \mathsf{Prob}\left[Z_A^{(i+1)} \otimes \mathbb{I}_M \otimes \mathsf{Proj}(Z_B^{(i+1)}, y), \left|\psi^{(i+1)}\right\rangle\right] \\
&= \mathsf{Prob}\left[H_y, \left|\psi^{(i+1)}\right\rangle\right]
\end{aligned}$$

where $H_y := Z_A^{(i+1)} \otimes \mathbb{I}_M \otimes \mathsf{Proj}(Z_B^{(i+1)}, y)$. Now clearly, for each y, using Equation 19, we have that $H_y \geq G_y$ and so $p_i(\cdot, y) \to p_{i+1}(\cdot, y)$ is an EBM transition for each y.

□

It turns out that, just as for standard WCF, the converse of also holds.

4.2 Λ-penEBM point game implies Λ-penWCF protocol

Theorem 19. *Given a Λ-penEBM point game with final point $[\![\beta, \alpha]\!]$, there exists a Λ-penWCF protocol with $R_A^* \leq \beta$ and $R_B^* \leq \alpha$.*

The proof of this is also very similar to the proof for the standard WCF case. Here, we improve the proof from [ARW19] by making the resulting protocol more round efficient. To this end, we use the following characterisation of EBM transitions.

Notation 20. We often consider (EBM line) transitions from $g \to h$, from $n_g > 0$ points to $n_h > 0$ points (here subscript g and h are not indices—but behaving more like the symbol $\prime$ behaves in g'). We write them as $g = \sum_{i=1}^{n_g} p_{g_i} [\![x_{g_i}]\!]$ and $h = \sum_{i=1}^{n_h} p_{h_i} [\![x_{h_i}]\!]$ where (again subscript g and h are not indices) $p_{g_i} > 0, p_{h_i} > 0$ are the weights, and $x_{g_i} \geq 0, x_{h_i} \geq 0$ are distinct coordinates (i.e. $x_{g_i} \neq x_{g_{i'}}$ for $i \neq i'$, and similarly $x_{h_i} \neq x_{h_{i'}}$ for $i \neq i'$).

[6] We point out that Equation 20 would not hold if there were a projector because $\mathsf{spectrum}(Z) \neq \mathsf{spectrum}(EZE)$ for an arbitrary projector E (even if it is constrained to be such that $E|\psi\rangle = |\psi\rangle$. Previous work [ACG+14] implicitly assume this in their proof making it slightly incorrect—as one can proceed like we did here by absorbing all projectors to the very end.

Lemma 21. *Let $g \to h$ be an EBM line transition with spectrum in $[a,b]$ and suppose g and h have disjoint support. Then, there exists a unitary U, and diagonal matrices X_h, X_g (with no multiplicities except possibly those of a and b) of size at most $m \times m$ for $m := n_g + n_h - 1$ such that*

$$\underbrace{\begin{bmatrix} x_{h_1} & & & & & \\ & \ddots & & & & \\ & & x_{h_{n_h}} & & & \\ & & & b & & \\ & & & & \ddots & \\ & & & & & b \end{bmatrix}}_{:=X_h} \geq U \underbrace{\begin{bmatrix} x_{g_1} & & & & & \\ & \ddots & & & & \\ & & x_{g_{n_g}} & & & \\ & & & a & & \\ & & & & \ddots & \\ & & & & & a \end{bmatrix}}_{:=X_g} U^\dagger$$

and

$$\begin{bmatrix} \sqrt{p_{h_1}} \\ \vdots \\ \sqrt{p_{h_{n_h}}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = U \begin{bmatrix} \sqrt{p_{g_1}} \\ \vdots \\ \sqrt{p_{g_{n_g}}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} =: |\psi\rangle$$

or more briefly, $X_h \geq U X_g U^\dagger$ and $\sum_{i=1}^{n_h} \sqrt{p_{h_i}} |i\rangle = U \left(\sum_{i=1}^{n_g} \sqrt{p_{g_i}} |i\rangle \right) =: |\psi\rangle$.

This is also true if g and h have common support, i.e., if $g = \sum_{i=1}^{n_k} p_{k_i} [\![x_{k_i}]\!] + \sum_{i=1}^{n_g} p_{g_i} [\![x_{g_i}]\!]$ and $h = \sum_{i=1}^{n_k} p_k [\![x_{k_i}]\!] + \sum_{i=1}^{n_h} p_{h_i} [\![x_{h_i}]\!]$.

Proof. Let us quickly make some elementary observations. Since $g \to h$ is EBM, there exist matrices $H \geq G$ and a vector $|\psi\rangle$ such that $\mathsf{Prob}[H, |\psi\rangle] = h$ and $\mathsf{Prob}[G, |\psi\rangle] = g$. One can write

$$\begin{aligned} & H \geq G & \\ \iff \quad & U_h X_h U_h^\dagger \geq U_g X_g U_g^\dagger & \text{diagonalising} \\ \iff \quad & X_h \geq U X_g U & U := U_h^\dagger U_g. \end{aligned}$$

It takes some work to argue about the dimension and multiplicities, but once that is done, it is straightforward to conclude that $|\psi\rangle$ can be chosen to have the form claimed. □

4.3 Proof of Theorem 19

The rest of this section is devoted to proving Theorem 19. Suppose a Λ-penEBM point game $(p_0, p_1, \ldots, p_n)$ is given. To each "frame", i.e., to each p_i, we associate a "canonical form".

Definition 22 (Canonical Form)**.** Consider the tuple $(|\psi\rangle_{ABM}, Z^A, Z^B)$ where $\mathcal{A}, \mathcal{B}, \mathcal{M}$ are (finite dimensional) Hilbert spaces and $|\psi\rangle_{ABM} \in \mathcal{A} \otimes \mathcal{B} \otimes \mathcal{M}$, $Z^A \in \mathcal{A}$, $Z^B \in \mathcal{B}$. We say this tuple is in the *Canonical Form* with respect to a frame $p = \sum_i P_i [\![x_i, y_i]\!]$ of a TDPG if (see Figure 6) $|\psi\rangle = \sum_i \sqrt{P_i} |ii\rangle_{AB} \otimes |\varphi\rangle_M$, $Z^A = \sum_i x_i |i\rangle \langle i|_A$ and $Z^B = \sum_i y_i |i\rangle \langle i|_B$ where $|\varphi\rangle_M$ is an arbitrary state (representing the state of extra uncoupled registers that might be present).

Our approach is to assign a canonical form to each frame of the Λ-penEBM point game point game and construct unitaries (and projectors) that define a Λ-penWCF which admits the Z variables in the canonical form as dual feasible solutions. This ensures that the resulting Λ-penWCF protocol has the same cheating reward as the final point of the Λ-penEBM point game.

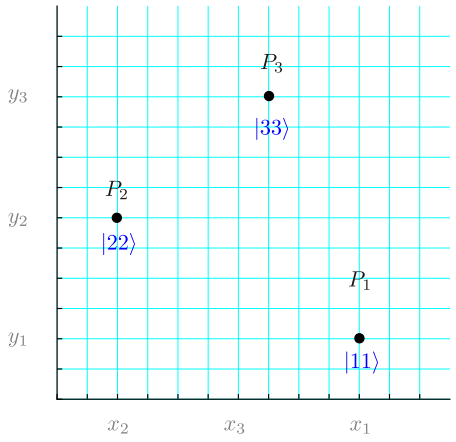


Figure 6: A frame in the Canonical Form.

We use the reverse time convention (as we did in the proof of Proposition 17): The variables $(|\psi_{(i)}\rangle, Z^A_{(i)}, Z^B_{(i)})$ describing the ith frame p_i of Λ-EBPM point game, and the unitary $U_{(i)}$ transforming p_i to p_{i+1}(which we define shortly), are going to be related to those of the Λ-WCF protocol by

$$|\psi_i\rangle := |\psi_{(n-i)}\rangle, Z_i = Z_{(n-i)}, \text{ and } U_i = U^\dagger_{(n-i)}. \tag{21}$$

We emphasise that the proof is based on that for the standard WCF case (as in [ARW19])—and this is by design because we chose the scoring convention carefully when defining Λ-penWCF protocols in Definition 7. We now proceed to give an informal description of the unitaries to provide some intuition and afterwards a formal description of them.

Informal Description of the Unitaries

Consider frames $p_j =: g$ and $p_{j+1} =: h$ of a Λ-penEBM point game $(p_0, \ldots, p_n)$. For simplicity, suppose $g \to h$ involves changing points along a horizontal line (i.e., corresponding to Alice's move) as illustrated in Figure 7a. We use $\{g_i\}$ to index the initial points and $\{h_i\}$ to index the final points and $\{k_i\}$ to index points that do not change. More precisely, we write $g = \sum_i p_{g_i} [\![x_{g_i}, y_{g_i}]\!] + \sum_i p_{k_i} [\![x_{k_i}, y_{k_i}]\!]$ and $h = \sum_i p_{h_i} [\![x_{h_i}, y_{h_i}]\!] + \sum_i p_{k_i} [\![x_{k_i}, y_{k_i}]\!]$ where $y_{g_i} = y_{h_i} = y$ for all i. Let $\{\{|g_i\rangle\}_i, \{|h_i\rangle\}_i, \{|k_i\rangle\}_i, |m\rangle\}$ denote orthonormal vectors (for instance, $\langle h_i | g_{i'} \rangle = 0$). We assume that a canonical form is given for g using $|g_i\rangle$ instead $|i\rangle$ and that for h is given using $|h_i\rangle$ instead of $|i\rangle$. Using Lemma 21, it is not hard to see that there is a unitary U such that $\sum_i x_{h_i} |h_i\rangle \langle h_i| \geq E_h U \left(\sum_i x_{g_i} |g_i\rangle \langle g_i|\right) U^\dagger E_h$ where $E_h = \sum_i |h_i\rangle \langle h_i|$.

Consider the following state which is in the canonical form for g,

$$|\psi_{(1)}\rangle = \left(\sum_i \sqrt{p_{g_i}} |g_i g_i\rangle_{AB} + \sum_i \sqrt{p_{k_i}} |k_i k_i\rangle_{AB} \right) \otimes |m\rangle_M .$$

We want Bob to send his part of $|g_i\rangle$ states to Alice via the message register. One way of doing this is to have Bob conditionally swap to obtain the following

$$|\psi_{(2)}\rangle = \sum_i \sqrt{p_{g_i}} |g_i g_i\rangle_{AM} \otimes |m\rangle_B + \sum_i \sqrt{p_{k_i}} |k_i k_i\rangle_{AB} \otimes |m\rangle_M .$$

The intuition here is that since $y_{g_i} = y_{h_i} = y$, and since Bob performs this step, it will not correspond to a non-trivial constraint in the dual SDP. Alice can now update the probabilities locally by applying U'—which is the

same as U but uses $|gg_i\rangle$ and $|hh_i\rangle$ instead of $|g_i\rangle$ and $|h_i\rangle$ while leaving the remaining space unchanged—to obtain

$$|\psi_{(3)}\rangle = \sum_i \sqrt{p_{h_i}}\,|h_i h_i\rangle_{AM} \otimes |m\rangle_B + \sum_i \sqrt{p_{k_i}}\,|k_i k_i\rangle \otimes |m\rangle_M .$$

This is the step that must satisfy a non-trivial constraint in the dual SDP. Finally, Bob now accepts the new state by "unswapping" to obtain

$$|\psi_{(4)}\rangle = \left(\sum_i \sqrt{p_{h_i}}\,|h_i h_i\rangle_{AB} + \sum_i \sqrt{p_{k_i}}\,|k_i k_i\rangle_{AB}\right) \otimes |m\rangle_M .$$

We will see that this step also satisfies the constraint in the dual SDP.

A few brief remarks before we move on to the formal proofs. (1) Note the the actual protocol runs backwards relative to how we described the operations here (see Equation 21). (2) Even though we are going from p_i to p_{i+1}, instead of using a single unitary, we are using three unitaries (of which only one is actually non-trivial). This basically allows us to decouple the message register after each p_i to p_{i+1} transition. (3) In our description above, we assumed that points along a single horizontal line are affected. However, we can handle multiple horizontal lines at once.

Formal Description of the unitaries

Consider frames $p_j =: g$ and $p_{j+1} =: h$ of a Λ-EBM point game $(p_0, \ldots, p_n)$ as illustrated in Figure 7b where multiple horizontal line transitions take place. We write

$$g =: \sum_\ell \left(\sum_i p_{g_{\ell i}}\,[\![x_{g_{\ell i}}, y_{g_\ell}]\!]\right) + \sum_i p_{k_i}\,[\![x_{k_i}, y_{k_i}]\!]$$

and

$$h =: \sum_\ell \left(\sum_i p_{h_{\ell i}}\,[\![x_{h_{\ell i}}, y_{h_\ell}]\!]\right) + \sum_i p_{k_i}\,[\![x_{k_i}, y_{k_i}]\!]$$

where we use labels $g_{\ell i}$ (resp. $h_{\ell i}$) to denote the properties of the ith point, involved in the ℓth horizontal transition, in frame g (resp. frame h) and since $y_{g_{\ell i}} = y_{g_{\ell i'}}$ for all i, i', we simply use y_{g_ℓ} (similarly for y_{h_ℓ}). We further simplify the notation and write

$$y_{g_\ell} = y_{h_\ell} =: y_\ell$$

as ℓ indexes horizontal transitions which means the y coordinates for the corresponding points in g and h are the same. We are going to define a dual SDP Theorem 12 and also construct a dual feasible solution to it, iteratively. Let $\{\{|g_{\ell i}\rangle\}_{\ell i}, \{|h_{\ell i}\rangle\}_{\ell i}, \{|k_i\rangle\}_i, |m\rangle\}$ denote orthonormal vectors (as in the previous subsection, except for the change in indexing). Consider the following definitions.

1. **The jth frame.**

   Consider the following definitions:

$$\begin{aligned}
|\psi_{(j,1)}\rangle &:= \left(\sum_{\ell i} \sqrt{p_{\ell i}}\,|g_{\ell i} g_{\ell i}\rangle_{AB} + \sum_i \sqrt{p_{k_i}}\,|k_i k_i\rangle_{AB}\right) \otimes |m\rangle_M , \\
Z^A_{(j,1)} &:= \sum_{\ell i} x_{g_{\ell i}}\,|g_{\ell i}\rangle\langle g_{\ell i}|_A + \sum_i x_{k_i}\,|k_i\rangle\langle k_i|_A , \\
Z^B_{(j,1)} &:= \sum_{\ell i} y_\ell\,|g_{\ell i}\rangle\langle g_{\ell i}|_B + \sum_i y_{k_i}\,|k_i\rangle\langle k_i|_B .
\end{aligned}$$

Claim 23. The tuple $\left(\left|\psi_{(j)}\right\rangle, Z^A_{(j)}, Z^B_{(j)}\right)$ is in the canonical form for the frame $p_j =: g$.

2. **Bob sends to Alice.**

Consider the following definitions:

$$\begin{aligned}
\left|\psi_{(j,2)}\right\rangle &:= \sum_{\ell i} \sqrt{p_{g_{\ell i}}} \left|g_{\ell i} g_{\ell i}\right\rangle_{AM} \otimes \left|m\right\rangle_B + \sum_i \sqrt{p_{k_i}} \left|k_i k_i\right\rangle_{AB} \otimes \left|m\right\rangle_M \\
U_{(j,1)} &:= \sum_{\ell i} \left|g_{\ell i} m\right\rangle \left\langle m g_{\ell i}\right|_{BM} + \mathsf{h.c.} + \underbrace{\left[\mathbb{I}_{BM} - \left(\sum_{\ell i} \left|g_{\ell i} m\right\rangle \left\langle g_{\ell i} m\right| + \sum_{\ell i} \left|m g_{\ell i}\right\rangle \left\langle m g_{\ell i}\right|\right)_{BM}\right]}_{=\mathbb{I}_{\text{rest}}} \\
E_{(j,2)} &:= \sum_{\ell i} \left|g_\ell m\right\rangle \left\langle g_\ell m\right|_{BM} + \sum_i \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M \\
Z^A_{(j,2)} &= Z^A_{(j,1)} \\
Z^B_{(j,2)} &= \sum_\ell y_\ell \mathbb{I}_B^{\{\{g_{\ell i}\}_i, m\}} + \sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B
\end{aligned}$$

where $U_{(j,1)}$ is basically permuting $\left|g_{\ell i} m\right\rangle_{BM}$ with $\left|m g_{\ell i}\right\rangle_{BM}$.

Claim 24. It holds that (a) $\left|\psi_{(j,2)}\right\rangle = E_{(j,2)} U_{(j,1)} \left|\psi_{(j,1)}\right\rangle$ and (b) $Z_{(j,2)} \otimes \mathbb{I}_M \geq E_{(j,2)} U_{(j,1)} \left(Z^B_{(j,1)} \otimes \mathbb{I}_M\right) U^\dagger_{(j,1)} E_{(j,2)}$.

Proof. Item (a) holds trivially by definition of $U_{(j,1)}$ and $E_{(j,2)}$. In this proof, let U denote $U_{(j,1)}$, let E denote $E_{(j,2)}$. Note that $U^\dagger = U$. To establish item (b), we write

$$\begin{aligned}
& EU \left(Z^B_{(j,1)} \otimes \mathbb{I}_M\right) UE \\
=& EU \left(\sum_{\ell i} y_\ell \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right|_B \otimes \mathbb{I}_M + \sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M\right) UE \\
=& EU \Bigg(\sum_{\ell i} y_\ell \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right|_B \otimes \left|m\right\rangle \left\langle m\right|_M + \underbrace{\sum_{\ell i} y_\ell \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right| \otimes \left(\mathbb{I}_M - \left|m\right\rangle \left\langle m\right|_M\right)}_{} \\
& \qquad + \underbrace{\sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M}_{} \Bigg) UE \qquad \text{highlighted terms are unaffected by } U \\
=& E \Bigg(\sum_{\ell i} y_\ell \left|m g_{\ell i}\right\rangle \left\langle m g_{\ell i}\right|_{BM} + \sum_{\ell i} y_\ell \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right| \otimes \left(\mathbb{I}_M - \left|m\right\rangle \left\langle m\right|_M\right) \\
& \qquad + \sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M \Bigg) E \\
=& \left(\sum_{\ell i} y_\ell \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right| \otimes \left(\mathbb{I}_M - \left|m\right\rangle \left\langle m\right|_M\right) + \sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M\right) \\
\leq & \sum_\ell y_\ell \mathbb{I}_B^{\{\{g_{\ell i}\}, m\}} \otimes \mathbb{I}_M + \sum_i y_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_B \otimes \mathbb{I}_M = Z^B_{(j,2)} \otimes \mathbb{I}_M.
\end{aligned}$$

This means $EU \left(Z^B_{(j,1)} \otimes \mathbb{I}_M\right) UE \leq Z^B_{(j,2)} \otimes \mathbb{I}_M$. □

3. **Alice applies the non-trivial unitary.**

Consider the following definitions:

$$\begin{aligned}
\left|\psi_{(j,3)}\right\rangle &:= \sum_{\ell i} \sqrt{p_{h_{\ell i}}} \left|h_{\ell i} h_{\ell i}\right\rangle_{AM} \otimes \left|m\right\rangle_B + \sum_i \sqrt{p_{k_i}} \left|k_i k_i\right\rangle_{AB} \otimes \left|m\right\rangle_M \\
U_{(j,2)} &:= \left|w\right\rangle \left\langle v\right| + \text{other terms acting non-trivially on } \mathcal{H} := \mathsf{span}\{\{\left|h_{\ell i} h_{\ell i}\right\rangle\}_{\ell i}, \{\left|g_{\ell i} g_{\ell i}\right\rangle\}_{\ell i}\} \\
E_{(j,3)} &:= \left(\sum_{\ell i} \left|h_{\ell i}\right\rangle \left\langle h_{\ell i}\right| + \sum_i \left|k_i\right\rangle \left\langle k_i\right|\right)_A \otimes \mathbb{I}_M \\
Z^A_{(j,3)} &= \sum_{\ell i} x_{h_{\ell i}} \left|h_{\ell i}\right\rangle \left\langle h_{\ell i}\right|_A + \sum_i x_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_A \\
Z^B_{(j,3)} &= Z^B_{(j,2)}
\end{aligned}$$

where

$$\left|v\right\rangle = \frac{\sum_{\ell i} \sqrt{p_{g_{\ell i}}} \left|g_{\ell i} g_{\ell i}\right\rangle}{\sqrt{\sum_{\ell i} p_{g_{\ell i}}}}, \quad \left|w\right\rangle = \frac{\sum_{\ell i} \sqrt{p_{h_{\ell i}}} \left|h_{\ell i} h_{\ell i}\right\rangle}{\sqrt{\sum_{\ell i} p_{h_{\ell i}}}},$$

and $U_{(j,2)}$ is satisfies

$$\sum_{\ell i} x_{h_{\ell i}} \left|h_{\ell i} h_{\ell i}\right\rangle \left\langle h_{\ell i} h_{\ell i}\right| \geq E_{(j,2)} U_{(j,2)} \left(\sum_{\ell i} x_{g_{\ell i}} \left|g_{\ell i} g_{\ell i}\right\rangle \left\langle g_{\ell i} g_{\ell i}\right|\right) U^\dagger_{(j,2)} E_{(j,2)}. \tag{22}$$

Claim 25. Let $g \to h$ be an EBM (horizontal) transition. Then, there is a unitary $U_{(j,2)}$ of the form described above that satisfies Equation 22.

Proof. It follows from Lemma 21 by an appropriate change of basis. □

Claim 26. It holds that (a) $\left|\psi_{(j,3)}\right\rangle = E_{(j,3)} U_{(j,2)} \left|\psi_{(j,2)}\right\rangle$ and (b) $Z^A_{(j,3)} \otimes \mathbb{I}_M \geq E_{(j,3)} U_{(j,2)} \left(Z^A_{(j,2)} \otimes \mathbb{I}_M\right) U^\dagger_{(j,2)} E_{(j,3)}$.

Proof. Again, item (a) follows directly by definitions of $U_{(j,2)}$ and $E_{(j,3)}$. We focus on establishing item (b) and we use U and E to denote $U_{(j,2)}$ and $E_{(j,3)}$ respectively. We write

$$\begin{aligned}
EU\left(Z^A_{(j,2)} \otimes \mathbb{I}_M\right) U^\dagger E = EU \Bigg[&\sum_{\ell i} x_{g_{\ell i}} \left|g_{\ell i} g_{\ell i}\right\rangle \left\langle g_{\ell i} g_{\ell i}\right|_{AM} + \sum x_{g_{\ell i}} \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right|_A \otimes \left(\mathbb{I} - \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right|\right)_M \\
&+ \sum_i x_{k_i} \left|k_i\right\rangle \left\langle k_i\right|_A \otimes \mathbb{I}_M \Bigg] U^\dagger E
\end{aligned}$$

and observe that $Z^A_{(j,2)} \otimes \mathbb{I}_M$ is block diagonal in $\mathcal{H}$ and $\mathcal{H}^\perp$. In fact, $Z^A_{(j,3)} \otimes \mathbb{I}_M$ and U are also block diagonal in $\mathcal{H}$ and $\mathcal{H}^\perp$. In the expression above, terms in I belong to the $\mathcal{H} \times \mathcal{H}$ block and U acts non-trivially only on this block. Restricting to this block, it is immediate from Equation 22 that in the $\mathcal{H} \times \mathcal{H}$ block, the following holds:

$$\sum x_{h_{\ell i}} \left|h_{\ell i} h_{\ell i}\right\rangle \left\langle h_{\ell i} h_{\ell i}\right| \geq EU \left(\sum x_{g_{\ell i}} \left|g_{\ell i}\right\rangle \left\langle g_{\ell i}\right|\right) U^\dagger E.$$

In the other block, U acts as identity. On this block, $Z_{(j,3)} - EU\left(Z_{(j,2)} \otimes \mathbb{I}\right)U^\dagger E$ is

$$\begin{aligned}
&\left(\sum_{\ell i} x_{h_{\ell i}} |h_{\ell i}\rangle \langle h_{\ell i}|_A + \sum_i x_{k_i} |k_i\rangle \langle k_i|\right) \otimes \mathbb{I}_M \\
&\quad - \underbrace{E\left(\sum x_{g_{\ell i}} |g_{\ell i}\rangle \langle g_{\ell i}|_A \otimes (\mathbb{I} - |g_{\ell i}\rangle \langle g_{\ell i}|)_M\right) E}_{=0} - \cancel{E}\left(\sum_i x_{k_i} |k_i\rangle \langle k_i|_A \otimes \mathbb{I}_M\right)\cancel{E} \\
&= \sum_{\ell i} x_{h_{\ell i}} |h_{\ell i}\rangle \langle h_{\ell i}|_A \geq 0
\end{aligned}$$

because $x_{h_{\ell i}} \geq 0$ for all ℓ, i. □

4. **Bob accepts Alice's change.**

Consider the following definitions:

$$\begin{aligned}
|\psi_{(j,4)}\rangle &:= \left(\sum_{\ell i} \sqrt{p_{h_{\ell i}}} |h_{\ell i} h_{\ell i}\rangle_{AB} + \sum_i \sqrt{p_{k_i}} |k_i k_i\rangle_{AB}\right) \otimes |m\rangle_M \\
U_{(j,3)} &:= U_{(j,1)} = \sum_{\ell i} |h_{\ell i} m\rangle \langle m h_{\ell i}|_{BM} + \mathsf{h.c.} + \underbrace{\left[\mathbb{I}_{BM} - \left(\sum_{\ell i} |h_{\ell i} m\rangle \langle h_{\ell i} m| + \sum_{\ell i} |m h_{\ell i}\rangle \langle m h_{\ell i}|\right)_{BM}\right]}_{=\mathbb{I}_{\text{rest}}} \\
E_{(j,3)} &:= \left(\sum_{\ell i} |h_{\ell i}\rangle \langle h_{\ell i}| + \sum_i |k_i\rangle \langle k_i|\right)_B \otimes \mathbb{I}_M \\
Z^A_{(j,4)} &= Z^A_{(j,3)} \\
Z^B_{(j,4)} &= \sum_{\ell i} y_{\ell i} |h_{\ell i}\rangle \langle h_{\ell i}|_B + \sum_i y_{k_i} |k_i\rangle \langle k_i|_B .
\end{aligned}$$

Claim 27. The tuple $\left(|\psi_{(j,4)}\rangle, Z^A_{(j,4)}, Z^B_{(j,4)}\right)$ is in canonical form for the frame p_{j+1}.

Proof. Follows by inspection of the appropriate definitions. □

Claim 28. It holds that (a) $|\psi_{(j,4)}\rangle = E_{(j,4)} U_{(j,3)} |\psi_{(j,3)}\rangle$ and (b)

$$Z^B_{(j,4)} \otimes \mathbb{I}_M \geq E_{(j,3)} U_{(j,3)} \left(Z^B_{(j,3)} \otimes \mathbb{I}_M\right) U^\dagger_{(j,3)} E_{(j,3)}.$$

Proof. As before, item (a) follows from the definitions of $|\psi_{(j,4)}\rangle, |\psi_{(j,3)}\rangle$ and $E_{(j,4)}, U_{(j,3)}$. For item (b), denote by E, U the objects $E_{(j,4)}, U_{(j,3)}$. Recall $Z^B_{(j,3)} = Z^B_{(j,2)} = \sum_\ell y_\ell \mathbb{I}_B^{\{\{g_{\ell i}\},m\}} + \sum_i y_{k_i} |k_i\rangle \langle k_i|$.

We write

$$\begin{aligned}
EU\left(Z^B_{(j,3)}\otimes\mathbb{I}_M\right)U^\dagger E &= EU\Bigg(\sum_\ell y_\ell \mathbb{I}_B^{\{\{g_{\ell i}\},m\}}\otimes\mathbb{I}_M \\
&\qquad + \sum_i y_{k_i}\left|k_i\right\rangle\left\langle k_i\right|_B\otimes\mathbb{I}_M\Bigg)U^\dagger E \\
&= \underbrace{E\left(\sum_\ell y_\ell \mathbb{I}_B^{\{g_{\ell i}\}}\otimes\mathbb{I}_M\right)E}_{=0} + && \because U \text{ acts trivially on this} \\
&\quad EU\left(\sum_\ell y_\ell \mathbb{I}_B^{\{m\}}\otimes\mathbb{I}_M\right)U^\dagger E + \\
&\quad \sum_i y_{k_i}\left|k_i\right\rangle\left\langle k_i\right|_B\otimes\mathbb{I}_M \\
&= \sum_\ell y_\ell\left|h_{\ell i}m\right\rangle\left\langle h_{\ell i}m\right|_B + \sum_i y_{k_i}\left|k_i\right\rangle\left\langle k_i\right|_B\otimes\mathbb{I}_M \\
&\leq \left(\sum_{\ell i} y_{\ell i}\left|h_{\ell i}\right\rangle\left\langle h_{\ell i}\right|_B + \sum_i y_{k_i}\left|k_i\right\rangle\left\langle k_i\right|_B\right)\otimes\mathbb{I}_M \\
&= Z^B_{(j,4)}\otimes\mathbb{I}_M.
\end{aligned}$$

This establishes item (b) as well, completing the proof. □

Theorem 29 (Strengthening of Theorem 19; Λ-EBM point game $\implies$ Λ-penWCF protocol)**.** *Let $(p_0\dots p_n)$ be a Λ-EBM point game with final point $[\![\beta,\alpha]\!]$. Then, there exists a Λ-penWCF protocol with cheating rewards $R^*_A \leq \alpha$ and $R^*_B \leq \beta$ which uses $2n$ rounds of communication and $\log\left[3\cdot\left(\max_j\left(\mathsf{PointCount}(p_j)+\mathsf{PointCount}(p_{j+1})\right)+1\right)\right]$ qubits where $\mathsf{PointCount}(p_j) = \left|\{(x,y) : p_j(x,y)\neq 0\}\right|$ is the number of points with non-zero weight.*

Proof. We have done most of the work already. Consider the definitions of $\left|\psi_{(j,d)}\right\rangle, Z^A_{(j,d)}, Z^B_{(j,d)}, E_{(j,d)}, U_{(j,d)}$ for $j\in\{0,\dots n\}$ and $d\in\{1,2,3,4\}$ ($E_{j,1}:=\mathbb{I}$ and $U_{(j,4)}$ was not defined). Now, for the four steps of say frame j, we used vectors $\left\{\{\left|g_{\ell i}\right\rangle\}_{\ell i},\{\left|h_{\ell i}\right\rangle\}_{\ell i},\{\left|k_i\right\rangle\}_i,\left|m\right\rangle\right\}$ where gs were used for initial points, h for final points and k for points that do not change (relative to the transition $p_j\to p_{j+1}$). At the first step of the frame $j+1$, we re-use the vectors $\left\{\{\left|h_{\ell i}\right\rangle\},\{\left|k_i\right\rangle\},\left|m\right\rangle\right\}$ but we relabel them as to again be consistent with the same convention—gs are used for initial points, k for points that do not change (relative to the transition $p_{j+1}\to p_{j+2}$) while $\left|m\right\rangle$ stays unchanged. And the vector $\{\left|g_{\ell i}\right\rangle\}$ can be re-used or extended as needed, in the remaining steps.

Using this convention for the basis vectors of the Hilbert space, one can consider the following definition for the protocol

$$\begin{aligned}
(U_m,U_{m-1},\dots) &:= \left(U_{(j,1)},U_{(j,2)},U_{(j,3)},U_{(j+1,1)},U_{(j+1,2)},U_{(j+1,3)}\right)_{j=(0,2,\dots n)} \\
(E_m,E_{m-1},\dots) &:= \left(E_{(j,1)},E_{(j,2)},E_{(j,3)},E_{(j+1,1)},E_{(j+1,2)},E_{(j+1,3)}\right)_{j=(0,2\dots n)}
\end{aligned}$$

where $m=3n$. Crucially, communication is only required immediately after $U_{(j,1)}$ and $U_{(j,2)}$ but not after $U_{(j,3)}$ because the message register gets decoupled and can be discarded. Thus, even though there are $3n$ unitaries that are applied, only $2n$ messages are exchanged. The number of qubits can be seen to be as asserted by inspection.

Further, let

$$(Z^A_m, Z^A_{m-1}, \dots) := \left(Z^A_{(j,1)}, Z^A_{(j,2)}, Z^A_{(j,3)}, \underbrace{Z^A_{(j,4)} = Z^A_{(j+1,1)}}, Z^A_{(j+1,2)}, Z^A_{(j+1,3)} \right)_{j=(0,2\dots n)}$$

$$(Z^B_m, Z^B_{m-1}, \dots) := \left(Z^B_{(j,1)}, Z^B_{(j,2)}, Z^B_{(j,3)}, \overbrace{Z^B_{(j,4)} = Z^B_{(j+1,1)}}, Z^B_{(j+1,2)}, Z^B_{(j+1,3)} \right)_{j=(0,2\dots n)}$$

where our choice of basis ensures the highlighted terms are the same (which can be seen by their respective definitions and by Claim 23 and Claim 27). Further, one can define the measurements $\Pi_A^{(0/1/\perp)}$ and $\Pi_B^{(0/1/\perp)}$ implicitly from $Z_{A,m}$ and $Z_{B,m}$ as stated in point 4 of Theorem 12, and also $|\psi_{A,0}\rangle, |\psi_{B,0}\rangle$ is defined as the eigenvector of $Z_{A,0}$ and $Z_{B,0}$. With these, one can define a Λ-penWCF protocol and check that the corresponding dual SDP as stated in Theorem 12, using Claim 24, Claim 25, Claim 26, Claim 27 and Claim 28, admits a solution specified by the dual variables (i.e. the Zs) above. □

5 Λ-Penalty Time Dependent Point Games (Λ-pen TDPG)

In the previous section, we looked at Λ-pen EBM point games that were built using EBM transitions. Prior works [Moc07, ACG+14] observed that it is helpful to consider EBM functions instead of transitions. This is because the set of EBM functions form a convex cone and one can use conic duality to obtain a better characterisation. We, therefore, first define EBM functions, recall its alternate characterisation—called valid functions—using operator monotone functions. We end the section by defining Λ-(penalty) point games using valid functions instead of EBM functions, together with the analogue of Theorem 29 that uses valid functions instead of EBM transitions.

5.1 EBM and valid functions (results from conic duality)

Definition 30 (K, EBM functions)**.** A function $a : [0,\infty) \to \mathbb{R}$ with finite support is an *EBM function* if the line transition $a^- \to a^+$ is EBM (see Definition 14), where $a^+ : [0,\infty) \to [0,\infty)$ and $a^- : [0,\infty) \to [0,\infty)$ denote, respectively, the positive and negative part of a (i.e. $a = a^+ - a^-$ with $\mathsf{supp}(a^+) \cap \mathsf{supp}(a^-) = \emptyset$ and $a^\pm \geq 0$).

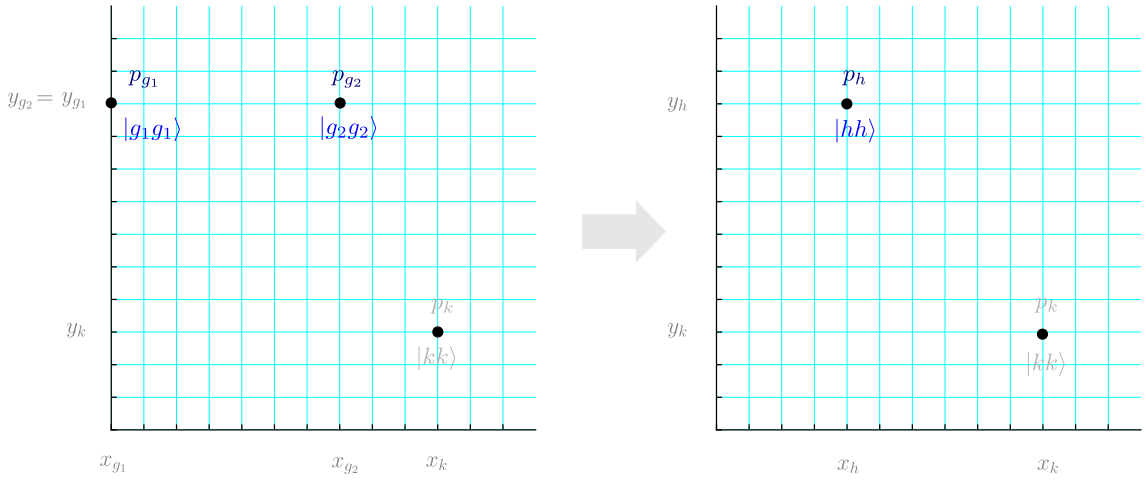
It turns out that the closure of EBM functions equals the set of valid functions, which are defined below. This follows from the fact that the bi-dual of a convex cone equals the closure of a cone. The dual of EBM functions is the set of operator monotone functions and, in turn, the dual of operator monotone functions is defined to be the set of valid functions. What helps here is that the set of operator monotone functions are known to have a very nice characterisation.

Definition 31 (Valid functions and transitions.)**.** We first define *valid functions* with one and two variables.

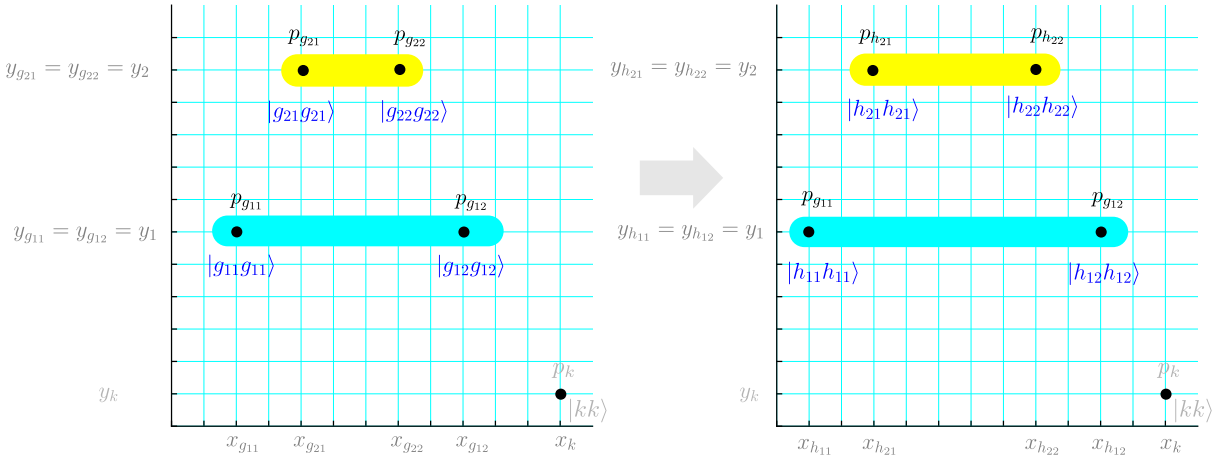
1. Let $a : [0,\infty) \to \mathbb{R}$ be a function with finite support such that $\sum_x a(x) = 0$. Then a is a *valid function* if for all $\lambda > 0$, $\sum_x \frac{-a(x)}{\lambda + x} \geq 0$ and $\sum_x x \cdot a(x) \geq 0$.
2. Let $t : [0,\infty) \times [0,\infty) \to \mathbb{R}$ is a
   (a) *horizontally valid function* if for all $y \geq 0$, $t(\cdot, y)$ is a valid function;
   (b) *vertically valid function* if for all $x \geq 0$, $t(x, \cdot)$ is a valid function.

Given finitely supported functions $g, h : [0,\infty) \times [0,\infty) \to [0,\infty)$ (resp. $g, h : [0,\infty) \to [0,\infty)$), we say $g \to h$ is a *horizontally/vertically valid transition (resp. valid line transition)* if $t := h - g$ is a horizontally/vertically valid function (resp. valid function).

When clear from the context, we drop *horizontally/vertically* and simply use the term valid function (even for bivariate functions).



(a) In [ARW19], they could only handle point game transitions along a single horizontal line at a time.



(b) Point game transition along multiple horizontal lines.

Figure 7: The parallelising step to improve round-efficiency. Comparison with prior work. Allowing multiple horizontal transitions to take place at once.

5.2 Λ-pen TDPG $\implies$ Λ-penWCF protocols

Just as we defined Λ-EBM point games that used the definition of an EBM transition, one can define Λ-point games with EBM functions. The basic idea being that instead of requiring the point game $(p_0, \dots, p_n)$ to be such that for each odd i (resp. even i), $p_i \to p_{i+1}$ is horizontally (resp. vertically) valid, we require that $p_{i+1} - p_i$ is a horizontal EBM function (resp. vertical EBM function)—where horizontal/vertical EBM functions are defined in the same spirit as we defined horizontal/vertical EBM transitions, i.e., by fixing one coordinate and requiring the remaining function to be an EBM function.

Since the closure of EBM functions turns out to be the same as the set valid functions, one can similarly define Λ-point games with valid functions. Interestingly, the analogue of Lemma 21 holds also for valid functions, once projectors are allowed as in Claim 25. Using this, one can use essentially the same proof that Λ-EBM point games implies Λ-pen WCF protocol to conclude that Λ-point games with valid functions also imply Λ-pen WCF protocols (with $2n$ rounds of communication).

Definition 32 (Λ-pen TDPG)**.** A Λ-penalty (time dependent) point game with valid functions is a sequence $(t_1, \dots t_n)$ of n functions such that $\{t_1, t_3 \dots\}$ (resp. $\{t_0, t_2 \dots\}$) are vertically valid (resp. horizontally valid) such that

- $\frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!] + \sum_{i=1}^{n} t_i = [\![\beta, \alpha]\!]$ (for some $\alpha, \beta \in [\Lambda, \Lambda+1]$)
- $\forall j \in \{1, \dots n\}$, $\frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!] + \sum_{i=1}^{n} t_i \geq 0$.

By the jth frame (of the point game), we mean $p_j := \frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!] + \sum_{i=1}^{n} t_i$ (for $j \in \{0, 1, \dots, n\}$ with $p_0 := \frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!]$). The point $[\![\beta, \alpha]\!]$ is referred to as the final point (of the point game).

The first condition above simply ensures that one reaches a single final point while the second condition ensures that each intermediate frame is valid.

Claim 33. Let t be a horizontally valid function. Then, there is a unitary $U_{(j,2)}$ of the form described in Step 3 "Alice applies the non-trivial unitary" (right above Claim 25) that satisfies Equation 22.

Proof sketch. Follows from prior works. See, for instance, Lemma 19 in [ARVW24].[7] □

We can now state the analogue of Theorem 29 for Λ-point games with valid functions.

Theorem 34 (Λ-penTDPG $\implies$ Λ-penWCF protocol)**.** *Let $(p_0, \dots, p_n)$ be the frames of a Λ-penalty (time dependent) point game, $(t_1, \dots t_n)$, with valid functions (see Definition 32) that has $[\![\beta, \alpha]\!]$ as the final point. Then, there exists a Λ-penWCF protocol with cheating rewards $R_A^* \leq \alpha$ and $R_B^* \leq \beta$ which uses $2n$ rounds of communication and $3 \cdot \log\left[(\max_j (\mathsf{PointCount}(p_j) + \mathsf{PointCount}(p_{j+1})) + 1)\right]$ qubits where $\mathsf{PointCount}(p_j) = |\{(x, y) : p_j(x, y) \neq 0\}|$ is the number of points with non-zero weight.*

Proof sketch. Proceed as in Section 4.2, and use Claim 33 instead of Claim 25. □

6 Λ-penalty Time Independent Point Games

We now define Λ-pen Time Independent Point Games (Λ-pen TDPGs).

Definition 35 (Λ-pen TIPG, ε-approx Λ-pen TIPG)**.** A Λ-penalty time independent point game (Λ-pen TIPG) is specified by two functions (h, v) where h is horizontally valid and v is vertically satisfying the constraint

$$h + v = 1 \cdot [\![\beta, \alpha]\!] - \frac{1}{2}[\![\Lambda, \Lambda+1]\!] - \frac{1}{2}[\![\Lambda+1, \Lambda]\!] \tag{23}$$

[7] [ARVW24] is a concise, self-contained version, describing results from [ARW19] and [ARV21] while [ARVW22] is the comprehensive self-contained version that includes all the details.

for some $\beta, \alpha \in [\Lambda, \Lambda+1]$. We call the point $[\![\beta, \alpha]\!]$ the final point of the game and call the set $\mathcal{S} = \mathsf{supp}(h) \cup \mathsf{supp}(v) \backslash \mathsf{supp}(h+v)$, the set of intermediate points.

An ε-approximate Λ-penalty time independent point game (ε-approx Λ-pen TIPG) is defined exactly as above except that instead of Equation 23, we require

$$\left\| h + v - \left(1 \cdot [\![\beta, \alpha]\!] - \frac{1}{2} [\![\Lambda, \Lambda+1]\!] - \frac{1}{2} [\![\Lambda+1, \Lambda]\!] \right) \right\|_1 \leq \varepsilon$$

where $\|f\|_1 := \sum_{x \in \mathsf{supp}(f)} |f(x)|$.

It is easy to see that a Λ-pen TDPG results in a Λ-pen TIPG with the same final point. Now we prove the converse. Note that in the theorem below we use slightly different notation to that in Theorem 36.

Theorem 36 (ε-approx Λ-pen TIPG $\implies$ Λ-pen TDPG)**.** *Given an ε-approximate Λ-penalty TIPG (h, v) with $h = v^T$, final point (β, α), with $m_1 = \min\{\mathsf{min\text{-}coordinate}(h), \mathsf{min\text{-}coordinate}(v)\}$, choose any $c_1 \in \left(0, \frac{m_1^2}{(\Lambda+1)\cdot\Lambda}\right)$. Then, for every $\delta \in (\delta_{\min}, 1)$ there is a Λ-penalty TDPG $(p_0, \ldots p_n)$ with final point $(\beta + \mathsf{err}, \alpha + \mathsf{err})$ where $\mathsf{err} = \sqrt{\delta \cdot (m_2 - \alpha)(m_2 - \beta)}$, $n = 10 + 2/\eta_2$. These, in turn are specified by*

- $\delta_{\min} := (1 - \varepsilon_2) \cdot \frac{c_3 \varepsilon_1}{1 + c_3 \varepsilon_1} + \varepsilon_2$, *with* $c_3 := c_1^{-1} - 1$,
- $m_2 := \max\{\mathsf{max\text{-}coordinate}(h), \tilde{m}_2\}$ *with*
- $\tilde{m}_2 := \min\left\{ (1 - w_1) \left(\frac{1}{(\Lambda+1)} - \frac{w_1}{m_1} \right)^{-1} : w_1 \in \{w_1^{\pm}\} \right\}$ *for*

$$w_1^{\pm} = \frac{\sqrt{8c_1\Lambda^2(\Lambda+1)^2 + m_1^2(8c_1\Lambda(\Lambda+1)+1) - 8c_1\Lambda\left(2\Lambda^2+3\Lambda+1\right)m_1} \pm m_1}{2(\Lambda+1)(m_1 - \Lambda)},$$

and finally,

- $\eta_2 := \frac{\delta_{\mathsf{clyst}}}{\|h^-\|} \cdot \frac{c_1(1-\varepsilon_1)+\varepsilon_1}{(1-\delta_{\mathsf{clyst}})}$ *with* $\delta_{\mathsf{clyst}} = 1 - ((1-\varepsilon_1) + \varepsilon_1/c_1) \cdot \frac{1-\delta}{1-\varepsilon_2}$.

Above, $\varepsilon_1, \varepsilon_2 \leq \varepsilon$ are such that $s = (1-\varepsilon_1)\, s_{\mathsf{ideal}} + \varepsilon_1 \cdot s_{\mathsf{error}}$ and $e = (1-\varepsilon_2) e_{\mathsf{ideal}} + \varepsilon_2 \cdot e_{\mathsf{error}}$ where $\|s_{\mathsf{error}}\|_1, \|e_{\mathsf{error}}\|_1 = 1$. The number of points in p_i is at most the number of points in (h, v), i.e. $|\mathsf{supp}(p_i)| \leq |\mathsf{supp}\{h\} \cup \mathsf{supp}\{v\}|$ for all $i \in \{0, \cdots, n\}$.

Compared to the analogous result for WCF, there are two key differences.

1. In WCF, the point games could not involve points with any coordinate below 0. And the original proofs relied on this fact. This fact itself translates, in the cheat penalty setting, to the constraint that the points cannot have coordinates lower than Λ. However, this is an unnecessary constraint and indeed, the point games we consider, do in fact involve points with coordinates lower than Λ. We thus generalise the proof slightly to allow points with coordinates below Λ.

2. The other difference is that our numerical algorithm produces approximate penalty TIPGs. For readers familiar with the proof for WCF, this involves the use of catalyst points and absorbing them in the end. This is very similar to having residual points from the approximation that need to be absorbed in the end into the final point. We therefore combine steps to obtain the final bound on error and communication complexity.

Before we start the proof, we note that combining Theorem 34 and Theorem 36, one immediately obtains a Λ-penalty WCF protocol starting from an ε-approximate Λ-penalty TDPG, together with bounds on the number of qubits and the number of rounds of communication.

6.1 Establishing that Λ-pen TIPG $\implies$ Λ-pen TDPGs assuming intermediate lemmas

We later consider sequences of valid transitions and to this end, it helps to introduce the notion of transitively valid functions.

Definition 37 (transitively valid)**.** Consider two functions $g, h : [0,\infty)\times[0,\infty) \to [0,\infty)$ with finite support. The transition $g \to h$ is *transitively valid (with n steps)* if there is a sequence of finitely supported functions $p_i : [0,\infty) \times [0,\infty) \to [0,\infty)$ such that $g \to p_1, p_1 \to p_2, \cdots, p_{n-1} \to p_n, p_n \to h$ are all valid transitions (see Definition 31).

To prove Theorem 36, we observe that the following statements are enough—proofs of which we get back to later. We use the following notation for the remainder of this section.

Notation 38. Let (h, v) be a ε-approx Λ-pen TIPG where $h = v^T$ (i.e., $h(x,y) = v(y,x)$ for all x, y). Let $h + v = e - s$ where e and s have disjoint support and both $e, s \geq 0$.

For any valid function, say h, we use $h = h^+ - h^-$ to denote the positive and negative parts of h, i.e., h^+ and h^- have disjoint support and $h^+, h^- \geq 0$ are both non-negative at all coordinates. Define $s_{\text{ideal}} := \frac{1}{2}[\![\Lambda, \Lambda+1]\!] + \frac{1}{2}[\![\Lambda+1, \Lambda]\!]$ and $e_{\text{ideal}} := [\![\beta, \alpha]\!]$ for some $\beta, \alpha \in [\Lambda, \Lambda+1]$. A "configuration" or a "frame" is any bivariate function $f : [0,\infty) \times [0,\infty) \to [0,\infty)$ with finite support. Finally, $\mathsf{max\text{-}coordinate}(f)$ of a frame is the highest x or y coordinate that any point with non-zero weight has, i.e. $\mathsf{max\text{-}coordinate}(f) := \max\{\{x : \exists y' \text{ s.t. } f(x,y') > 0\} \cup \{y : \exists x' \text{ s.t.} f(x',y) > 0\}\}$. Similarly, $\mathsf{min\text{-}coordinate}(f)$ is the lowest $\max\{x, y\}$ coordinate[8] with non-zero weight in f, i.e. $\mathsf{min\text{-}coordinate}(f) := \min\{\max\{x,y\} : f(x,y) > 0\}$.

The motivation for introducing max-coordinate and min-coordinate notation will become clear shortly. We start claiming that $s \to e$ becomes transitively valid if h^- is already present in the "background".

Lemma 39. *It holds that $s + h^- \to s + h^+$ and $s + h^+ \to e + h^-$ are both valid transitions.*

Obviously, one cannot simply assume the starting frame is $s + h^-$. The following basically allows one to ensure that $s + \gamma h^- \to e + \gamma h^-$ is also transitively valid, where the number of steps depends on the scalar $0 < \gamma < 1$.

Lemma 40. *Let $0 < \gamma < 1$ and ξ be any frame. If $s + \xi \to e + \xi$ is transitively valid with n_1 steps then $s + \gamma\xi \to \delta_1 s + (1-\delta_1)e + \gamma\xi$ is transitively valid with $n_2 := n_1 \cdot \left\lfloor \frac{1}{\gamma} \right\rfloor$ steps where $\delta_1 = 1 - \gamma \cdot \lfloor 1/\gamma \rfloor$. When $1/\gamma$ is some natural number, we get $s + \gamma\xi \to e + \gamma\xi$ is transitively valid with $n_2 = n_1/\gamma$ steps.*

That handles a big part of showing that a Λ-pen TIPG can be converted into a Λ-pen TDPG. We need to address a few things: (1) how to produce and absorb the γh^- frame and (2) how to handle the approximation ε.

First, we rewrite the approximate starting and ending frames.

Claim 41. If $h = v^T$ and $\|h + v - (e_{\text{ideal}} - s_{\text{ideal}})\|_1 \leq \varepsilon < 1$, then $s = (1-\varepsilon_1)\, s_{\text{ideal}} + \varepsilon_1 \cdot s_{\text{error}}$ and $e = (1-\varepsilon_2)e_{\text{ideal}} + \varepsilon_2 \cdot e_{\text{error}}$ where $\|s_{\text{error}}\|_1, \|e_{\text{error}}\|_1 = 1$ and ε_1 and ε_2 are both at most ε.

We now show that one can go from

$$s_{\text{ideal}} \to (1-\delta)e_{\text{ideal}} + \delta \cdot [\![m_2, m_2]\!] \tag{24}$$

but with some constraints. First, a parameter c_1 must be selected which must be in some range specified by the minimum coordinate of interest m_1 and the cheat penalty Λ. Now, this c_1 parameter specifies $\delta_{\min}$ which is the smallest δ can be and also m_2 which specifies a large coordinate (explained shortly). The proof that Equation 24 is transitively valid essentially starts by using a small weight from s_{ideal} and producing (a) s_{error}

[8] Consider the $\Lambda = 0$ case. Without the max, the answer for h or v would be 0 because they involve points on the axes.

together with (b) a small amount of h^-. In this process, a residual point at $[\![m_2, m_2]\!]$ with a small weight is also produced. Then using lemma 39 and Lemma 40, the proof allows one to convert $s + \gamma h^- \to e + \gamma h^-$. Finally, with some raises, one obtains the desired final configuration/frame, i.e. $(1-\delta)e_{\mathsf{ideal}} + \delta \cdot [\![m_2, m_2]\!]$. The subtlety in all this is, of course, making everything quantitative and keeping track of the number of intermediate transitions.

Lemma 42. *Suppose the premise of Claim 41 holds and let*

$$m_1 = \min\{\mathsf{min\text{-}coordinate}(s_{\mathsf{error}}), \mathsf{min\text{-}coordinate}(h^-)\}, \quad \textit{and} \quad c_1 \in \left[0, \frac{m_1^2}{(\Lambda+1)\cdot\Lambda}\right).$$

It holds that for every $\delta \in (\delta_{\min}, 1)$, the transition

$$s_{\mathsf{ideal}} \to (1-\delta)e_{\mathsf{ideal}} + \delta \cdot [\![m_2, m_2]\!]$$

is transitively valid with $6 + 2/\eta_2$ steps (assuming $1/\eta_2$ is an integer), where we define

- $\delta_{\min} := (1-\varepsilon_2) \cdot \frac{c_3\varepsilon_1}{1+c_3\varepsilon_1} + \varepsilon_2$, *with* $c_3 := c_1^{-1} - 1$,
- $m_2 := \max\{\mathsf{max\text{-}coordinate}(h), \tilde{m}_2\}$ *with*
- $\tilde{m}_2 := \min\left\{(1-w_1)\left(\frac{1}{(\Lambda+1)} - \frac{w_1}{m_1}\right)^{-1} : w_1 \in \{w_1^{\pm}\}\right\}$ *for*

$$w_1^{\pm} = \frac{\sqrt{8c_1\Lambda^2(\Lambda+1)^2 + m_1^2(8c_1\Lambda(\Lambda+1)+1) - 8c_1\Lambda\left(2\Lambda^2+3\Lambda+1\right)m_1} \pm m_1}{2(\Lambda+1)(m_1-\Lambda)},$$

and finally,

- $\eta_2 := \frac{\delta_{\mathsf{clyst}}}{\|h^-\|} \cdot \frac{c_1(1-\varepsilon_1)+\varepsilon_1}{(1-\delta_{\mathsf{clyst}})}$ *with* $\delta_{\mathsf{clyst}} = 1 - ((1-\varepsilon_1) + \varepsilon_1/c_1) \cdot \frac{1-\delta}{1-\varepsilon_2}$.

It remains to show that the point $[\![m_2, m_2]\!]$ can be absorbed. The following lemma says that for any desired deviation $\epsilon > 0$ from the ideal final frame $[\![\beta, \alpha]\!]$, there is a $\delta_{\max}$ below which for any δ, $(1-\delta)e_{\mathsf{ideal}} + \delta \cdot [\![m_2, m_2]\!] \to [\![\beta+\epsilon, \alpha+\epsilon]\!]$ is transitively valid. In the case where $\delta_{\min} = 0$ (i.e. we are in the exact case where $\varepsilon_1 = \varepsilon_2 = 0$), it means that one can always find a $\delta < \delta_{\max}$ and thus obtain protocols with end points getting arbitrarily close to $[\![\beta, \alpha]\!]$—at the expense of increased communication (scaling roughly as $1/\delta$).

Lemma 43. *Recall that $e_{\mathsf{ideal}} = [\![\beta, \alpha]\!]$ and fix any $[\![m_2, m_2]\!]$ (such that $m_2 \geq \max\{\beta, \alpha\}$). For every $\epsilon > 0$, there is a $\delta_{\max} := \frac{\epsilon^2}{(m_2-\alpha)(m_2-\beta)}$ such that for all $\delta < \delta_{\max}$, the transition*

$$(1-\delta)e_{\mathsf{ideal}} + \delta \cdot [\![m_2, m_2]\!] \to [\![\beta+\epsilon, \alpha+\epsilon]\!]$$

is transitively valid with 4 steps.

However, when $\delta_{\min} > 0$, one cannot get arbitrarily close to $[\![\beta, \alpha]\!]$. In that case, for every c_1, one can compute $\delta_{\min}(c_1)$ and $m_2(c_1)$. Given m_2, and ϵ, using the lemma above, one can also compute $\delta_{\max}(\epsilon, c_1)$. The smallest ϵ for which $\delta_{\min}(c_1) \leq \delta_{\max}(\epsilon, c_1)$ allows one to choose $\delta_{\min}(c_1) \leq \delta \leq \delta_{\max}(\epsilon, c_1)$ which specifies a protocol with $6 + 2/\eta_2(c_1, \delta)$ rounds of interaction and $\epsilon = \delta_{\max} \cdot (m_2 - \beta)$ bias.

Proof of Theorem 36. The proof of Lemma 42 uses Lemma 39, Lemma 40 and Claim 41 as we alluded to above and prove later. The proof of Lemma 43 is not too involved and is also deferred (follows from prior works). Assuming Lemma 42 and Lemma 43, the proof of Theorem 36 is immediate. □

6.2 Proving the intermediate lemmas

Before we get to the more interesting proofs, let us establish Claim 41 which basically says that $\|h + v - (e_{\mathsf{ideal}} - s_{\mathsf{ideal}})\|_1 \leq \varepsilon$ implies that one can write $s = (1-\varepsilon_1)s_{\mathsf{ideal}} + \varepsilon_1 \cdot s_{\mathsf{error}}$ and $e = (1-\varepsilon_2)e_{\mathsf{ideal}} + \varepsilon_2 \cdot e_{\mathsf{error}}$ where the functions on the right are non-negative and normalised, and ε_1 and ε_2 are at most ε.

Proof of Claim 41. We start by recalling that $\mathsf{supp}(e) \cap \mathsf{supp}(s) = \emptyset$. Further, it also holds that $\mathsf{supp}(e) \supseteq \mathsf{supp}(e_{\mathsf{ideal}})$ because otherwise, $\|e - s - (e_{\mathsf{ideal}} - s_{\mathsf{ideal}})\|_1 \geq \|e_{\mathsf{ideal}}\| = 1$ which violates our premise that $\varepsilon < 1$. Similarly, one can argue that $\mathsf{supp}(s) \supseteq \mathsf{supp}(s_{\mathsf{ideal}})$. Thus, one can write $\|e - s - (e_{\mathsf{error}} - s_{\mathsf{ideal}})\| = \|e - e_{\mathsf{ideal}}\|_1 + \|s - s_{\mathsf{ideal}}\|_1 \leq \varepsilon$. This means that both $\|e - e_{\mathsf{ideal}}\|_1 \leq \varepsilon$ and $\|s - s_{\mathsf{ideal}}\|_1 \leq \varepsilon$. Finally, using these inequalities and since $h = v^T$, we can write $s = (1-\varepsilon_1)s_{\mathsf{ideal}} + \varepsilon_1 \cdot s_{\mathsf{error}}$ where $\varepsilon_1 \leq \varepsilon$ and $\mathsf{supp}(s_{\mathsf{ideal}}) \cap \mathsf{supp}(s_{\mathsf{error}}) = \emptyset$ (if no such ε_1 existed, then the inequality could not hold). Since $\|s\|_1 = 1$, and $\|s_{\mathsf{ideal}}\| = 1$, we have that $\|s_{\mathsf{error}}\| = 1$. One can similarly establish that $e = (1-\varepsilon_2)e_{\mathsf{ideal}} + \varepsilon_2 e_{\mathsf{error}}$ where $\mathsf{supp}(e_{\mathsf{ideal}}) \cap \mathsf{supp}(e_{\mathsf{error}}) = \emptyset$ and $\varepsilon_2 \leq \varepsilon$. □

As a warm-up, we state and prove a simple observation that will be useful later.

Lemma 44. *If the transition $g' \to h'$ is transitively valid with n steps and $\zeta : [0,\infty) \times [0,\infty) \to [0,\infty)$ is a non-negative function with finite support, then the transition $\delta \cdot g' + \zeta \to \delta \cdot h' + \zeta$ is also transitively valid with n steps for all $\delta > 0$.*

Proof. It suffices to show that the statement holds for valid line transitions (because a valid transition is a valid line transition along each x (or y) coordinate). Given that $\ell \to r$ is a valid line transition, it is immediate that $\delta \cdot \ell + \xi \to \delta \cdot r + \xi$ for any $\xi : [0,\infty) \to [0,\infty)$ finitely supported function. This is because $\delta \cdot (r - \ell)$ is a valid function if $(r - \ell)$ is a valid function, which is the case because $\ell \to r$ is a valid transition. The number of steps are preserved because one can apply this reasoning to each step of $g' \to p_1, p_1 \to p_2 \ldots p_n \to h'$ to obtain $\delta \cdot g' + \zeta \to \delta \cdot p_1 + \zeta, \ldots \delta \cdot p_n + \zeta \to \delta \cdot h' + \zeta$. □

We now establish Lemma 39 which says that $s + h^- \to e + h^-$ is transitively valid with 2 steps.

Proof of Lemma 39. It is clear that $s + h^- \to s + h^+$ is valid because h is a valid function and we can add s to the valid transition $h^- \to h^+$ while preserving validity as in Lemma 44. Showing $s + h^+ \to e + h^+$ requires some work. One can rewrite $h + v = s - e$ as $h^+ - h^- + v^+ - v^- = e - s$ which in turn can be rewritten as

$$h^+ + s = \underbrace{e + h^- - v^+}_{\xi} + v^-.$$

We prove shortly that $\xi \geq 0$ but assume this is the case for the moment. Then, since $v^- \to v^+$ is a valid transition, using Lemma 44, we have that $\xi + v^- \to \xi + v^+ = e + h^- - \cancel{v^+ + v^+}$ is also a valid transition. Thes establishes that $s + h^+ \to e + h^-$ is a valid transition, assuming $\xi \geq 0$ which we now prove. Observe that ξ can only be negative at $\mathsf{supp}(v^+)$. Since $\mathsf{supp}(v^-) \cap \mathsf{supp}(v^+) = \emptyset$, it follows that $\xi \geq 0 \iff \xi + v^- \geq 0$. But recall that $\xi + v^- = h^+ + s$ which is manifestly non-negative. □

Next, we establish Lemma 40 which says, in particular, that if $s + h^- \to e + h^-$ is transitively valid with n_1 steps, then $s + \gamma h^- \to e + \gamma h^-$ is also transitively valid with $n_2 := n_1/\gamma$ steps.

Proof of Lemma 40. We proceed as follows

$$\begin{aligned} s+\gamma\xi &= (1-\gamma)s+\gamma(s+\xi) \\ &\to (1-\gamma)s+ \qquad\quad +\gamma(e+\xi) \\ &= (1-2\gamma)s+\gamma(s+\xi)+\gamma e \\ &\to (1-2\gamma)s+\gamma(s+\xi)+2\gamma e \\ &\to (1-3\gamma)s+\gamma(s+\xi)+3\gamma e \\ &\vdots \\ &\to (1-m\gamma)s+ \qquad\quad +m\gamma e+\gamma\xi \end{aligned}$$

where $m=\lfloor 1/\gamma \rfloor$ so that $0\le 1-m\gamma\le\gamma$ and $m\gamma\ge 1-\gamma$. This corresponds to $n_1\cdot m$ valid transitions thus $s+\gamma\xi\to\delta_1 s+(1-\delta_1)e+\gamma\xi$ is transitively valid. □

The statements so far, did not really depend on the actual definitions of h,v (that also specify s,e). To prove the subsequent statements, we assume we are starting with s_{ideal} and want to end up with a single point. Let us start with a simple claim.

Claim 45. For every coordinate $m_1\in(0,\Lambda]$, and any weight $c_1\in\left[0,\frac{m_1^2}{(\Lambda+1)\cdot\Lambda}\right)$, the transition

$$s_{\mathsf{ideal}}\to(1-\delta_1)s_{\mathsf{ideal}}+\delta_1\Big(c_1\left[\!\left[m_1,m_1\right]\!\right]+\underbrace{(1-c_1)}_{=:c_2}\left[\!\left[m_2,m_2\right]\!\right]\Big)$$

is transitively valid in 2 steps and $m_2:=\min\left\{(1-w_1)\left(\frac{1}{(\Lambda+1)}-\frac{w_1}{m_1}\right)^{-1}:w_1\in\{w_1^{\pm}\}\right\}$ where $w_1^{\pm}=\frac{\sqrt{8c_1\Lambda^2(\Lambda+1)^2+m_1^2(8c_1\Lambda(\Lambda+1)+1)-8c_1\Lambda(2\Lambda^2+3\Lambda+1)m_1}\pm m_1}{2(\Lambda+1)(m_1-\Lambda)}$.

Proof of Claim 45. The idea is elementary. Simply perform a horizontal split and then a vertical split starting from, say, $\left[\!\left[\Lambda+1,\Lambda\right]\!\right]$. The horizontal split

$$\begin{aligned} s_{\mathsf{ideal}} &= \frac{1}{2}\left[\!\left[\Lambda+1,\Lambda\right]\!\right]+\frac{1}{2}\left[\!\left[\Lambda,\Lambda+1\right]\!\right] \\ &\to \frac{1}{2}w_1\left[\!\left[m_1,\Lambda\right]\!\right]+\frac{1}{2}(1-w_1)\left[\!\left[m_2',\Lambda\right]\!\right]+\frac{1}{2}\left[\!\left[\Lambda,\Lambda+1\right]\!\right] \end{aligned}$$

gives the following constraint

$$\frac{w_1}{m_1}+\frac{1-w_1}{m_2'}\le\frac{1}{(\Lambda+1)}.$$

Here we treat $0\le w_1\le 1$ as a free variable but the constraint above requires, $w_1\le\frac{m_1}{(\Lambda+1)}$ and we have

$$m_2':=(1-w_1)\left(\frac{1}{(\Lambda+1)}-\frac{w_1}{m_1}\right)^{-1}.$$

(Note that as w_1 approaches $m_1/(\Lambda+1)$, m_2' approaches infinty).

Now, we split vertically

$$\begin{aligned} &\frac{1}{2}w_1\left[\!\left[m_1,\Lambda\right]\!\right]+\frac{1}{2}(1-w_1)\left[\!\left[m_2',\Lambda\right]\!\right]+\frac{1}{2}\left[\!\left[\Lambda,\Lambda+1\right]\!\right] \\ \to&\frac{1}{2}w_1\left(w_2\left[\!\left[m_1,m_1\right]\!\right]+(1-w_2)\left[\!\left[m_1,m_2''\right]\!\right]\right)+\frac{1}{2}(1-w_1)\left[\!\left[m_2',\Lambda\right]\!\right]+\frac{1}{2}\left[\!\left[\Lambda,\Lambda+1\right]\!\right] \end{aligned}$$

which gives the following constraint

$$\frac{w_2}{m_1} + \frac{(1-w_2)}{m_2''} \leq \frac{1}{\Lambda}.$$

Treating $0 \leq w_2 \leq 1$ also as a free variable, we get $w_2 \leq \frac{m_1}{\Lambda}$ with

$$m_2'' := (1-w_2)\left(\frac{1}{\Lambda} - \frac{w_2}{m_1}\right)^{-1}.$$

We demand that $m_2'' = m_2' =: m_2$ to obtain a relation between w_1 and w_2, i.e.

$$\begin{aligned}
&\frac{(1-w_2)}{(1-w_1)} = \frac{\left(\frac{1}{\Lambda} - \frac{w_2}{m_1}\right)}{\left(\frac{1}{\Lambda+1} - \frac{w_1}{m_1}\right)} \\
\iff &\frac{(1-w_2)}{(1-w_1)} = \frac{\left(\frac{m_1 - \Lambda w_2}{\Lambda}\right)}{\left(\frac{m_1 - (\Lambda+1)w_1}{(\Lambda+1)}\right)} \\
\iff &\frac{\Lambda(1-w_2)}{(m_1 - \Lambda w_2)} = \frac{(\Lambda+1)(1-w_1)}{(m_1 - (\Lambda+1)w_1)}.
\end{aligned}$$

We also have another relation between w_1 and w_2 which is $c_1 = \frac{1}{2} w_1 w_2$. Substituting $w_2 = \frac{2c_1}{w_1}$ in the expression above, we can solve for $w_1$in terms of m_1, Λ and c_1

$$\begin{aligned}
&\frac{\Lambda(1 - 2c_1/w_1)}{(m_1 - 2\Lambda c_1/w_1)} = \frac{(\Lambda+1)(1-w_1)}{(m_1 - (\Lambda+1)w_1)} \\
\iff &\Lambda(1 - 2c_1/w_1)(m_1 - (\Lambda+1)w_1) = (\Lambda+1)(1-w_1)(m_1 - 2\Lambda c_1/w_1)
\end{aligned}$$

which yields a quadratic that can be solved to obtain

$$w_1^{\pm} = \frac{\sqrt{8c_1\Lambda^2(\Lambda+1)^2 + m_1^2(8c_1\Lambda(\Lambda+1)+1) - 8c_1\Lambda\left(2\Lambda^2+3\Lambda+1\right)m_1} \pm m_1}{2(\Lambda+1)(m_1-\Lambda)}$$

and that in turn yields $m_2 = \min\left\{(1-w_1)\left(\frac{1}{(\Lambda+1)} - \frac{w_1}{m_1}\right)^{-1} : w_1 \in \{w_1^{\pm}\}\right\}$. □

We now establish Lemma 42 which shows that one can start from s_{ideal} and get to $(1-\delta)e_{\mathsf{ideal}} + \delta \cdot [\![m_2, m_2]\!]$ for $\delta \geq \delta_{\min}$ where $\delta_{\min}$ vanishes as $\varepsilon_1, \varepsilon_2$ vanish (i.e. if $s = s_{\mathsf{ideal}}$ and $e = e_{\mathsf{ideal}}$).

Proof of Lemma 42. Our goal is to establish that, for every $1 > \delta > 0$, $s_{\mathsf{ideal}} \to (1-\delta)e_{\mathsf{ideal}} + \delta \cdot e_{\mathsf{rest}}$ is transitively valid and to give a bound on the number steps it takes.

Let $1 > \delta_{\mathsf{clyst}} > 0$ be a free parameter. Let $\delta_{\mathsf{s\text{-}fix}} > 0$ be another parameter which is fixed by δ_{clyst} (and other parameters). We can now easily see the following are transitively valid transitions.

$$\begin{aligned}
s_{\mathsf{ideal}} \to \quad & (1 - \delta_{\mathsf{s\text{-}fix}} - \delta_{\mathsf{clyst}}) s_{\mathsf{ideal}} + \delta_{\mathsf{s\text{-}fix}} c_1 [\![m_1, m_1]\!] + \delta_{\mathsf{clyst}} c_1 [\![m_1, m_1]\!] \\
& + (\delta_{\mathsf{s\text{-}fix}} + \delta_{\mathsf{clyst}}) c_2 [\![m_2, m_2]\!] \qquad \text{follows from Claim 45} \\
\to \quad & (1 - \delta_{\mathsf{s\text{-}fix}} - \delta_{\mathsf{clyst}}) s_{\mathsf{ideal}} + \delta_{\mathsf{s\text{-}fix}} c_1 s_{\mathsf{error}} + \delta_{\mathsf{clyst}} c_1 \frac{h^-}{\|h^-\|} \\
& + (\delta_{\mathsf{s\text{-}fix}} + \delta_{\mathsf{clyst}}) c_2 [\![m_2, m_2]\!] \qquad (25)
\end{aligned}$$

where the second transition follows from the fact that m_1 is the minimum coordinate appearing in the point game. We want the coefficient of s_{ideal} and s_{error} to have the ratio $(1-\delta_{\mathsf{s\text{-}fix}}-\delta_{\mathsf{clyst}})/(\delta_{\mathsf{s\text{-}fix}}c_1) = (1-\varepsilon_1)/\varepsilon_1$ which yields

$$\varepsilon_1 - \varepsilon_1\delta_{\mathsf{s\text{-}fix}} - \varepsilon_1\delta_{\mathsf{clyst}} = \delta_{\mathsf{s\text{-}fix}}c_1(1-\varepsilon_1) \iff \delta_{\mathsf{s\text{-}fix}} = \frac{\varepsilon_1(1-\delta_{\mathsf{clyst}})}{c_1(1-\varepsilon_1)+\varepsilon_1}$$

we therefore have

$$\delta_{\mathsf{s\text{-}fix}} = \begin{cases} \frac{(1-\delta_{\mathsf{clyst}})}{c_1\left(\frac{1}{\varepsilon_1}-1\right)+1} & \varepsilon_1 > 0 \\ 0 & \varepsilon_1 = 0. \end{cases}$$

To proceed, it would be easier to rewrite Equation 25 as follows, where we solve for η_1, η_2, η_3 below. We therefore impose that

$$\text{Equation 25} = \quad (1-\eta_1)\left((1-\varepsilon_1)s_{\mathsf{ideal}} + \varepsilon_1 s_{\mathsf{error}} + \eta_2 h^-\right) + \eta_3 \left[\!\left[m_2, m_2\right]\!\right]$$

where $(1-\eta_1)\varepsilon_1 = \delta_{\mathsf{s\text{-}fix}}c_1$ which yields

$$\eta_1 := 1 - \frac{\delta_{\mathsf{s\text{-}fix}}c_1}{\varepsilon_1} = 1 - \frac{c_1(1-\delta_{\mathsf{clyst}})}{c_1(1-\varepsilon_1)+\varepsilon_1}$$

and $(1-\eta_1)\eta_2 = \frac{\delta_{\mathsf{clyst}}c_1}{\|h^-\|}$ which yields

$$\eta_2 := \frac{\delta_{\mathsf{clyst}}\cancel{c_1}}{\|h^-\|} \cdot \frac{\varepsilon_1}{\delta_{\mathsf{s\text{-}fix}}\cancel{c_1}} = \frac{\delta_{\mathsf{clyst}}}{\|h^-\|} \cdot \frac{c_1(1-\varepsilon_1)+\varepsilon_1}{(1-\delta_{\mathsf{clyst}})} \tag{26}$$

and finally $\eta_3 := 1 - (1-\eta_1)\left(1+\eta_2\left\|h^-\right\|\right)$ from normalisation. Note that all of these are fixed by δ_{clyst}.

Proceeding with this change of variables, we have that the following transitions are transitively valid (we count the steps at the end).

$$\begin{aligned}
s_{\mathsf{ideal}} &\to (1-\eta_1)\left((1-\varepsilon_1)s_{\mathsf{ideal}} + \varepsilon_1 s_{\mathsf{error}} + \eta_2 h^-\right) + \eta_3\left[\!\left[m_2, m_2\right]\!\right] \\
&= (1-\eta_1)\left(s + \eta_2 h^-\right) + \eta_3\left[\!\left[m_2, m_2\right]\!\right] \\
&\to (1-\eta_1)\left(e + \eta_2 h^-\right) + \eta_3\left[\!\left[m_2, m_2\right]\!\right] && \text{in } 1/\eta_2 \text{ steps} \\
&= (1-\eta_1)\left((1-\varepsilon_2)e_{\mathsf{ideal}} + \varepsilon_2 e_{\mathsf{error}} + \eta_2 h^-\right) + \eta_3\left[\!\left[m_2, m_2\right]\!\right] \\
&\to (1-\eta_1)\left((1-\varepsilon_2)e_{\mathsf{ideal}} + (\varepsilon_2 + \eta_2\left\|h^-\right\|)\left[\!\left[m_2, m_2\right]\!\right]\right) + \eta_3\left[\!\left[m_2, m_2\right]\!\right] \\
&= (1-\delta)e_{\mathsf{ideal}} + \delta\left[\!\left[m_2, m_2\right]\!\right]
\end{aligned}$$

where $\delta = \eta_1 + \varepsilon_2 - \varepsilon_2\eta_1 = (1-\varepsilon_2)\eta_1 + \varepsilon_2$. Now,

$$\eta_1 \geq 1 - \frac{1}{(1-\varepsilon_1)+\varepsilon_1/c_1} = \frac{(1-\varepsilon_1)+\varepsilon_1/c_1 - 1}{(1-\varepsilon_1)+\varepsilon_1/c_1} = \frac{\left(\frac{1}{c_1}-1\right)\varepsilon_1}{1+\left(\frac{1}{c_1}-1\right)\varepsilon_1} = \frac{c_3\varepsilon_1}{1+c_3\varepsilon_1}$$

(where $c_3 := c_1^{-1} - 1$) which is saturated as δ_{clyst} goes to zero, and so

$$\delta \geq (1-\varepsilon_2)\cdot\frac{c_3\varepsilon_1}{1+c_3\varepsilon_1} + \varepsilon_2.$$

As a sanity check, note that when $\varepsilon_1 = \varepsilon_2 = 0$, the bound just requires $\delta \geq 0$ which basically means that we can choose the weight of the catalyst to be arbitrarily small to ensure δ is arbitrarily small, in the exact case. For approximate penalty TIPGs, we cannot go lower than the bound above. For small enough ε_1 and ε_2,

however, one should still be able to obtain low bias. We end by writing an expression for δ_{clyst} in terms of δ as

$$\begin{aligned}
\delta =& (1-\varepsilon_2)\left(1 - \frac{(1-\delta_{\mathsf{clyst}})}{(1-\varepsilon_1)+\varepsilon_1/c_1}\right) + \varepsilon_2 \\
\iff \quad \frac{(1-\delta_{\mathsf{clyst}})}{(1-\varepsilon_1)+\varepsilon_1/c_1} =& 1 - \frac{\delta-\varepsilon_2}{1-\varepsilon_2} \\
\iff \quad \frac{(1-\delta_{\mathsf{clyst}})}{(1-\varepsilon_1)+\varepsilon_1/c_1} =& \frac{1\cancel{-\varepsilon_2} - \delta \cancel{+\varepsilon_2}}{1-\varepsilon_2} \\
\iff \quad (1-\delta_{\mathsf{clyst}}) =& ((1-\varepsilon_1)+\varepsilon_1/c_1)\cdot\frac{1-\delta}{1-\varepsilon_2} \\
\iff \quad \delta_{\mathsf{clyst}} =& 1 - ((1-\varepsilon_1)+\varepsilon_1/c_1)\cdot\frac{1-\delta}{1-\varepsilon_2}.
\end{aligned}$$

Finally, η_2 can be computed using δ_{clyst} using Equation 26 and $1/\eta_2$ gives (roughly) the number of steps needed in the catalyst step.

It remains to count the total number of steps.

- 4 steps: 2 for for setting up $[\![m_1, m_1]\!]$, and 2 for turning them into s_{error} and h^-.
- $2/\eta_2$ steps: the catalyst step (for simplicity, we assume $1/\eta$ is an integer)
- 2 steps: final two raises to have all remaining points be at $[\![m_2, m_2]\!]$.

  Thus, the total number of steps turns out to be $6 + 2/\eta_2$.

□

It remains to establish Lemma 43 and its proof is essentially the same as that in the standard WCF setting. We restate it here for convenience.

Proof of Lemma 43. Define δ and δ' to be such that the following two merges are valid (we use m instead of m_2 for simplicity)

$$\begin{aligned}
\delta' [\![m, \alpha]\!] + \delta [\![m, m]\!] &\to (\delta+\delta') [\![m, \alpha+\epsilon]\!] \\
(1-\delta-\delta') [\![\beta, \alpha+\epsilon]\!] + (\delta+\delta') [\![m, \alpha+\epsilon]\!] &\to [\![\beta+\epsilon, \alpha+\epsilon]\!] .
\end{aligned}$$

The merge conditions are

$$\begin{aligned}
\delta'\alpha + \delta m &= (\delta+\delta')(\alpha+\epsilon) \\
(1-\delta-\delta')\beta + (\delta+\delta')m &= \beta+\epsilon
\end{aligned}$$

using which one can solve for δ and δ' as

$$\begin{aligned}
\delta &= \frac{\epsilon^2}{(m-\beta)(m-\alpha)} \\
\delta' &= \frac{\epsilon}{(m-\beta)}\left(1 - \frac{\epsilon}{m-\alpha}\right).
\end{aligned}$$

These can now be combined to see that the following transitions are valid

$$\begin{aligned}
(1-\delta)[\![\beta, \alpha]\!] + \delta [\![m, m]\!] &\to (1-\delta-\delta')[\![\beta, \alpha]\!] + \delta' [\![m, \alpha]\!] + \delta [\![m, m]\!] && \text{raise} \\
&\to (1-\delta-\delta')[\![\beta, \alpha]\!] + (\delta'+\delta) [\![m, \alpha+\epsilon]\!] && \text{using the first merge} \\
&\to (1-\delta-\delta')[\![\beta, \alpha+\epsilon]\!] + (\delta'+\delta) [\![m, \alpha+\epsilon]\!] && \text{raise} \\
&\to [\![\beta+\epsilon, \alpha+\epsilon]\!] && \text{using the second merge.}
\end{aligned}$$

Note that this takes four valid transitions. Now, note that one can set $\delta_{\max} = \frac{\epsilon^2}{(m-\beta)(m-\alpha)}$ and run the argument above for any $\delta \leq \delta_{\max}$ by first raising to ensure the starting frame is $(1-\delta_{\max}) [\![\beta, \alpha]\!] + \delta_{\max} [\![m, m]\!]$ and one can transitively go to $[\![\beta + \epsilon, \alpha + \epsilon]\!]$ in four steps. □

7 An Algorithm for Finding Approximate TIPGs

Despite the crucial role of point games for solving the coin-flipping problem, there have been notably few examples of point games constructed in the literature. Indeed, the family of TIPGs in [Moc07] and [HP13] are the only known family of TIPGs for the weak coin-flipping problem in which the bias tends to zero. To the best of our knowledge there are no general algorithms for constructing point games.

Theorem 36 showed that to solve the point game problems arising from the cheat-penalised weak coin-flipping problem, it suffices to find a time-independent point game which *approximately* computes the move on the right-hand side of equation (23). We therefore concern ourselves in this section with the following general problem.

Given $\epsilon > 0$, a start-configuration s and an end-configuration e, find a horizontally valid move h and a vertically valid move v such that

$$\|h + v - (e - s)\|_1 \quad \leq \quad \epsilon. \tag{27}$$

The difficulty in solving this problem is that it involves optimising a function over a high-dimensional space (namely, the set of all TIPGs) defined by an infinite number of linear constraints (conditions (7)–(8)). The insight behind our algorithm is that we can break this search problem into parts, where the first is a convex optimisation over a low-dimensional space, and the second is merely linear algebra. The algorithm is based on the two-dimensional profile function, a concept introduced in [Mil20].

We describe the algorithm in this section. Mathematica code for the algorithm is available at [AMMS25]. We provide an example calculation in Section 8. We leave a formal analysis of the performance of the algorithm to future work.

7.1 Setup

Assume that configurations $s, e \colon \mathbb{R}^2_{\geq 0} \to \mathbb{R}$ are given, with

$$\sum_{x,y} s(x,y) \quad = \quad \sum_{x,y} e(x,y). \tag{28}$$

We build towards the definition of a two-dimensional profile function of a move. Our definition is analogous that in [Mil20], although based on a different (simpler) mathematical formula for the one-dimensional profile function.

Definition 46 (Profile function)**.** For any $x > 0$, let $P_x : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be defined as

$$P_x(\lambda) = \begin{cases} \lambda x/(\lambda + x) & \text{if } \lambda > 0 \\ \\ 1 & \text{if } \lambda \leq 0. \end{cases} \tag{29}$$

Let

$$P_0(\lambda) = \begin{cases} 0 & \text{if } \lambda \geq 0 \\ \\ 1 & \text{if } \lambda < 0. \end{cases} \tag{30}$$

For any one-dimensional move f, let the profile function $\hat{f}\colon \mathbb{R} \to \mathbb{R}$ of f be defined by

$$\hat{f}(\lambda) \quad = \quad \sum_x f(x) P_x(\lambda). \tag{31}$$

By definition, f is valid if and only if $\hat{f}$ is nonnegative and $\hat{f}(-1) = 0$.

For any two-dimensional move r, define the profile function $\hat{r}\colon \mathbb{R}^2 \to \mathbb{R}$ by

$$\hat{r}(\alpha, \beta) = \sum_{x,y \in \mathbb{R}} r(x,y) P_x(\alpha) P_y(\beta). \tag{32}$$

It is easy to see that both vertically valid and horizontally valid moves must have non-negative profiles. Therefore, the end-configuration e can be reached from the start-configuration s only if $\hat{s} \leq \hat{e}$.

If W is a set, then we write $\mathbb{R}^W$ to mean the set of all functions from W to $\mathbb{R}$. If $U \subseteq W$ is a finite subset of W, then we can define an inner product on $\mathbb{R}^W$ by

$$\langle p, q \rangle_U \quad = \quad \sum_{\lambda \in U} p(\lambda) q(\lambda) \tag{33}$$

$$\|p\|_U \quad = \quad \sqrt{\sum_{x \in U} p(x)^2} \tag{34}$$

for any $p, q\colon W \to \mathbb{R}$. If W is itself a finite set, then we may simply write $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ (without any subscript) to denote $\langle \cdot, \cdot \rangle_W$ and $\|\cdot\|_W$.

7.2 Step 1: Choose threshhold constants and search sets

We first choose a positive real number δ. Roughly speaking, δ determines the precision of one of the initial steps in our search for a point game solution, and M places a limit on the size of the weights in the point games included in the search. These choices require striking a balance: smaller choices of δ and larger choices of M will yield a search that takes longer to execute, but which is more likely to yield a point game solution with smaller bias and less communication complexity.

Choose a finite set $S \subseteq \mathbb{R}_{\geq 0}$ and a finite set $T \subseteq \mathbb{R}$ such that $-1 \in T$. These sets are used to simplify our search for a solution to the point game problem (s, e): we only consider two-dimensional moves that are supported in the set $S \times S$, and when we are assessing whether two moves have a similar profile, we look only at the profile values on the set $T \times T$.

7.3 Step 2: Search for (approximately) valid moves with (approximately) correct profiles

The next step is to search for moves h and v such that $h + v$ has nearly the same profile as $e - s$, and such that h and v are (at least approximately) horizontally valid and vertically valid, respectively. We do this search in a way that is intended to minimise the complexity of the search space.

Let $S = \{s_1, s_2, \ldots, s_\ell\}$, where $s_i < s_{i+1}$ for all $i \in \{1, 2, \ldots, \ell\}$. Let $D\colon \mathbb{R}^{S \setminus \{s_\ell\}} \to \mathbb{R}^S$. For $f \in \mathbb{R}^{S \setminus \{s_\ell\}}$, we write $D(f) = D_f \in \mathbb{R}^S$. We define D as

$$D_f(s_1) \quad = \quad f(s_1) \tag{35}$$

$$D_f(s_j) \quad = \quad f(s_j) - f(s_{j-1}) \qquad \text{for } 2 \leq j \leq \ell - 1 \tag{36}$$

$$D_f(s_\ell) \quad = \quad -f(s_{\ell-1}). \tag{37}$$

The image of D is exactly the set of functions in $\mathbb{R}^S$ that sum to zero. Define also the linear map $H : \mathbb{R}^S \to \mathbb{R}^T$ by

$$H_f(t) \quad = \quad \hat{f}(t) \tag{38}$$

where $\hat{f}$ denotes the profile associated with f. In other words, the operator H maps a function f defined on S to its corresponding profile, restricted to the set T.

Let $H' = H \circ D$, and let the image of H' (which is a set of functions from T to $\mathbb{R}$) be denoted by $\operatorname{Im} H'$. The image $\operatorname{Im} H'$ has an inner product given by $\langle \cdot, \cdot \rangle_T$, and the domain space $\mathbb{R}^{S \setminus \{s_\ell\}}$ has a native inner product. Compute the singular values

$$c_1 \geq c_2 \geq \ldots \geq c_{\ell-1} \tag{39}$$

for H' and corresponding (unit-length, right) singular vectors

$$v_1, v_2, \ldots, v_{\ell-1} \in \mathbb{R}^{S \setminus \{s_\ell\}}. \tag{40}$$

Choose $k \in \{1, 2, \ldots, \ell - 1\}$ such that $c_j \geq \delta$ for all $j \leq k$ and $c_j < \delta$ for all $j > k$. Let $V \subseteq \mathbb{R}^{S \setminus \{s_\ell\}}$ be the span of $\{v_1, v_2, \ldots, v_k\}$. (Informally, the moves in $D(V^\perp)$ tend to have profiles that are close to zero, and therefore we will essentially ignore the space $D(V^\perp)$ during this step of the algorithm, and focus our search within $D(V)$.)

Let us say that a one-dimensional move $f \colon S \to \mathbb{R}$ is T**-valid** if $\sum_{x \in S} f(x) = 0$ and $\hat{f}(t) \geq 0$ for all $t \in T$. Compute an $h \colon S \times S \to \mathbb{R}$ which minimises

$$\left\| (\hat{h} + \hat{v}) - (\hat{e} - \hat{s}) \right\|_{T \times T} \tag{41}$$

where $v := h^\top$, and h is subject to the following constraint: each row of the move h is contained in $D(V)$ and is T-valid.

7.4 Step 3: Compute approximately valid moves with the correct start- and end-configuration

At this point we expect to have moves h and v that are approximately horizontally valid and approximately vertically valid, respectively, and are such that the profile of $h + v$ is close to that of $e - s$. However, this does not mean that $h + v$ is close to $e - s$. To remedy this, we now introduce additional moves p and q (correction factors) such that p is approximately horizontally valid, q is approximately vertically valid, and $(h + p) + (v + q))$ is equal to $e - s$.

Again we use singular value analysis, although we focus directly on the linear transformation H rather than on H'. Let

$$d_1 \geq d_2 \geq \ldots \geq d_\ell \tag{42}$$

be the singular values of H, and let

$$w_1, w_2, \ldots, w_\ell \in \mathbb{R}^S \tag{43}$$

be the corresponding singular vectors in the domain H. The vectors $\{w_i\}$ form an othornomal basis for $\mathbb{R}^S$, and $\hat{w}_1, \ldots, \hat{w}_\ell$ form an orthogonal set (under $\langle \cdot, \cdot \rangle_T$) such that

$$\|\hat{w}_j\|_T \quad = \quad d_j. \tag{44}$$

For any $j, k \in \{1, 2, \ldots, \ell\}$, define $w_j \otimes w_k \colon S \times S \to \mathbb{R}$ by

$$(w_j \otimes w_k)(x, y) \quad = \quad w_j(x) w_k(y). \tag{45}$$

The functions $\{w_j \otimes w_k\}$ likewise form an orthonormal basis for $\mathbb{R}^{S \times S}$, and the profiles of $\{w_j \otimes w_k\}$ are all orthogonal to one another under $\langle \cdot, \cdot \rangle_T$. The T-norm of the profile of $w_j \otimes w_k$ is $d_j d_k$.

We can therefore compute an orthogonal decomposition of the difference function $(h+v)-(e-s)$. Find real values $\{t_{jk}\}_{1\leq j,k\leq \ell}$ such that

$$(e-s)-(h+v) \quad = \quad \sum_{j,k} t_{jk}(w_j \otimes w_k). \tag{46}$$

Let

$$p \quad = \quad \sum_{j>k} t_{jk}(w_j \otimes w_k) + \frac{1}{2}\sum_{k} t_{kk}(w_k \otimes w_k) \tag{47}$$

$$q \quad = \quad \sum_{j<k} t_{jk}(w_j \otimes w_k) + \frac{1}{2}\sum_{k} t_{kk}(w_k \otimes w_k), \tag{48}$$

and let $h' = h + p$ and $v' = v + q$. Note that $p = q^\top$ and given that $v = h^\top$, we have $h' = v'^\top$. Finally,

$$h' + v' = e - s. \tag{49}$$

The intuition behind this construction is as follows. Since the profile of $(e-s)-(h+v)$ has small T-norm, the T-norms of the profiles of $t_{jk}(w_j \otimes w_k)$ (which are equal to $t_{jk}d_j d_k$, respectively) must likewise be small. Thus, for any $j, k \in \{1, 2, \ldots, \ell\}$, either t_{jk} is small, the T-norm of $\hat{w}_j$ is small, or the T-norm of $\hat{w}_k$ is small. In the case of the terms $t_{jk}(w_j \otimes w_k)$ that appear in the expression for p (see Equation 47), we always have $d_j \leq d_k$, and so either t_{jk} or d_j must be small, which makes p approximately horizontally valid. A similar heuristic suggests that q should be approximately vertically valid. Therefore we expect this step to yield a pair (h', v') such that h' is approximately horizontally valid and v' is approximately vertically valid.

7.5 Step 4: Project approximately valid moves to valid moves

The final step is to replace the approximately valid moves h' and v' with valid moves h_* and v_* such that $\|h_* + v_* - e + s\|_{S\times S}$ is minimised. To achieve this, we project each approximate move onto the corresponding set of valid moves. The projection procedure, outlined below, ensures that the resulting h_* and v_* remain as close as possible to h' and v' while satisfying the validity constraints.

To implement the projection, we consider the following constrained minimization problem:

$$\begin{aligned} \min_{v_* \in \mathbb{R}^{S\times S}} \quad & \left\|v' - v_*\right\|^2_{S\times S} \\ \text{s.t.} \quad & H(v_*) \geq 0 \end{aligned} \tag{50}$$

This problem seeks the valid move v_* closest to the approximate move v'. It can be reformulated as a quadratic program of the form

$$\begin{aligned} \min_{x} \quad & \tfrac{1}{2}x^\top Q x + c^\top x \\ \text{s.t.} \quad & Ax \leq b, \end{aligned} \tag{51}$$

where Q is the identity matrix, $c = -v'$, $A = -H$, and $b = 0$. We assume that the move v_* and v' have been mapped to a vector and that H has been mapped to a matrix. In this formulation, the quadratic objective captures the squared distance to v', while the linear inequality encodes the validity constraint.

Since $Q = I \geq 0$, the objective function is strictly convex. Combined with the linear inequality constraint, this makes the problem a convex quadratic program. Convexity guarantees both the existence of a global optimum and the availability of efficient numerical algorithms to compute it. In practice, this ensures that the projection step can be carried out robustly, yielding the closest valid move v_* to the given approximation v'.

Note that the constraint ensures that $\hat{v}_*(\lambda)$ is nonnegative on the set T. Consequently, T should be chosen to cover as broad a range of values as possible in order to guarantee the validity of the move. Recall that we

also required $-1 \in T$, but this constraint is automatically satisfied by imposing that the sum of the weights of a move is 0. In practice, however, we find that a relatively small choice of T is already sufficient to enforce the validity condition. We observe that, in general, applying this projection procedure to obtain v_* and h_* also ensures that $v_* = h_*^\top$. This symmetry arises because the corresponding condition is already enforced when constructing the approximate moves v' and h'.

8 penTIPGs for Cheat-Penalised WCF

We now present several of the penTIPGs obtained numerically, together with the protocols derived from them in terms of bias, number of messages, and qubits used. The penTIPG data are provided as matrices corresponding to the valid moves h_* and v_*. In addition, we include the matrix representation of the profile function H, along with the number of non-truncated elements, as described in Section 7.3. The sets S and T are also made available. All of these values are provided in a supplementary text file, which can be accessed at [AMMS25]. In this text file we present the matrices for the moves as h' and v' together with the matrix corresponding to H to project to valid moves. For completeness, we also present several representative penTIPGs explicitly as matrices within this section.

The bias and number of messages in the protocols derived from the penTIPGs can be determined using Theorem 36, while the number of qubits required follows from Theorem 34. The qubit cost depends on the size of the set S chosen in the construction. Notably, the number of elements in S does not need to be large in order to achieve solutions with low bias. Even more strikingly, our numerical results suggest that arbitrarily low bias can be attained with a constant number of qubits, in sharp contrast to Mochon's family of protocols, where vanishing bias requires taking the limit of infinitely many qubits.

As stated in Theorem 36, the number of points in the penTDPG—and therefore the number of messages—depends on the parameter η_2. This parameter in turn depends on δ and c_1. Numerically, we set c_1 as close as possible to $\frac{m_1^2}{(\Lambda+1)\Lambda}$ and then find that the closer we set δ to $\delta_{\min}$, the lower the bias. There is of course a trade-off between the number of messages and the bias obtained in this way.

8.1 penTIPGs

In this subsection we list some of the penTIPGs we found numerically. We provide the matrix encoding $v*$. The move h_* can be found by noting $h_* = v_*^\top$. We also provide the sets S and T together with a truncation parameter which defines how many singular values we keep for the optimisation. Note that we don't add -1 to the set T as that constraint is automatically satisfied in the numerics. For penTIPG 1 we present some detail on the bias and round complexity.

8.1.1 penTIPG 1 with $\Lambda = 1$

The parameters are defined as:

$$S = \{0.3, 0.7, 1, 1.25, 1.500005, 1.75, 2, 3\},$$

$$\text{truncation} = 6,$$

$$T = \{0.1, 0.3, 0.5, 1, 1.5, 2, 3, 4, 10, 1000\},$$

Matrix v_*:

$$\begin{bmatrix}
0.000000 & -0.012549 & 0.000867 & 0.002990 & -0.005655 & 0.000098 & 0.001934 & 0.002993 \\
0.012549 & 0.000000 & -0.027156 & -0.021163 & 0.007004 & 0.005266 & 0.025788 & -0.007110 \\
-0.000867 & 0.027156 & 0.000000 & -0.035587 & -0.014383 & -0.147784 & -0.304895 & -0.028092 \\
-0.002990 & 0.021163 & 0.035587 & 0.000000 & -0.196252 & 0.006301 & 0.165097 & -0.031862 \\
0.005655 & -0.007004 & 0.014383 & 0.196252 & 0.500000 & 0.227074 & 0.092193 & -0.028695 \\
-0.000098 & -0.005266 & 0.147784 & -0.006301 & -0.227074 & 0.000000 & 0.077654 & 0.016646 \\
-0.001934 & -0.025788 & -0.195105 & -0.165097 & -0.092193 & -0.077654 & 0.000000 & 0.064828 \\
-0.002993 & 0.007110 & 0.028092 & 0.031862 & 0.028695 & -0.016646 & -0.064828 & 0.000000
\end{bmatrix}$$

Trade-off between error and rounds. The error err and rounds of communication obtained in the protocol are controlled by the choice of δ and c_1 in Theorem 36. For example, when choosing $\delta = \delta_{\min} + 10^{-5}$ and c_1 close to $\frac{m_1^2}{(\Lambda+1)\Lambda}$, we can obtain $\mathrm{err} = 0.004$ and 8×10^6 rounds of communication as depicted in Figure 8. Note that the final bias of this protocol is dominated by err and therefore to decrease the bias we need to decrease δ. By choosing $\delta = \delta_{\min} + 10^{-7}$, we find that $=0.0004$ and the number of rounds is 7×10^9 as shown in Figure 9. Even though the number of rounds has increased by two orders of magnitude, the number of qubits used is the same. Therefore, reducing the error is just a matter of choosing δ as close as possible to $\delta_{\min}$.

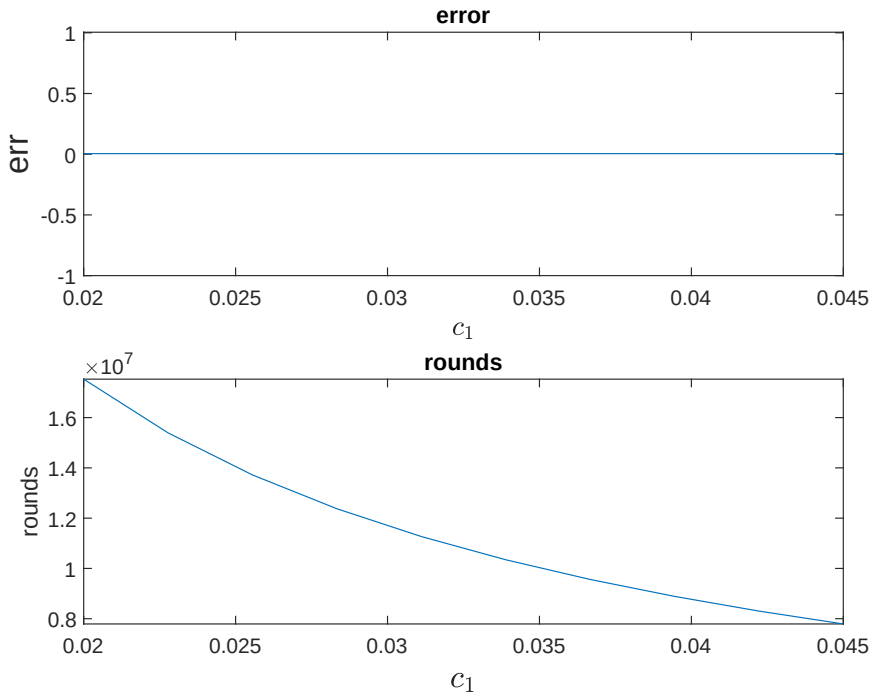


Figure 8: The error and rounds of communication for penTIPG 1 when choosing $\delta = \delta_{\min} + 10^{-5}$.

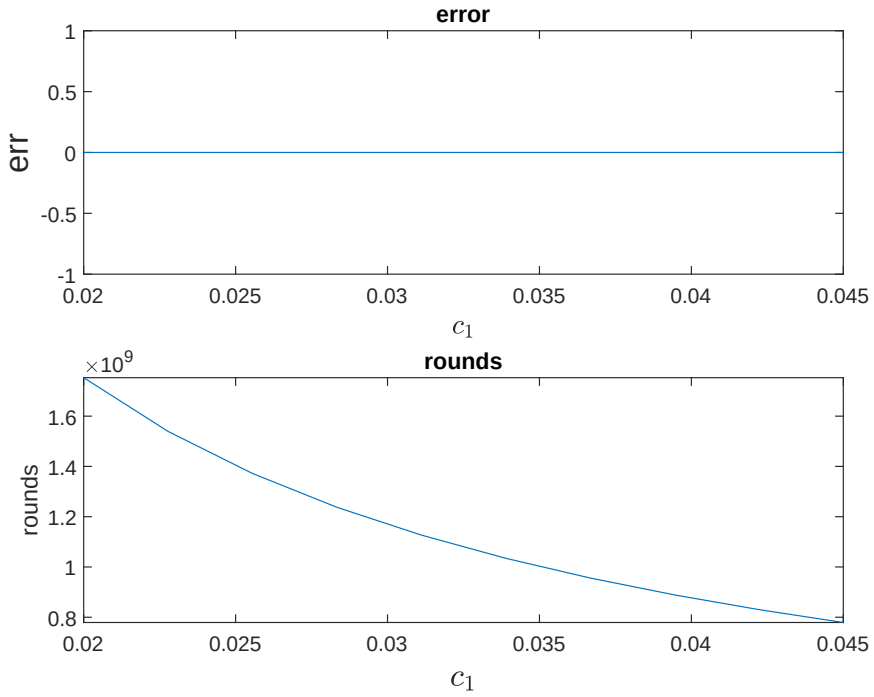


Figure 9: The error and rounds of communication for penTIPG 1 when choosing $\delta = \delta_{\min} + 10^{-7}$.

8.1.2 penTIPG 2 with $\Lambda = 1$

$$S = \{0.6, 0.8, 1, 1.225, 1.505, 1.75, 2, 2.5\},$$

$$\text{truncation} = 6,$$

$$T = \{0.1, 0.3, 0.5, 1, 1.5, 2, 3, 4, 10, 1000\},$$

Matrix v_*:

$$\begin{bmatrix}
0.000000 & -0.022240 & -0.010330 & 0.001572 & 0.017053 & 0.012344 & -0.008076 & -0.002015 \\
0.022240 & 0.000000 & -0.032154 & -0.023705 & -0.041653 & 0.014959 & 0.091371 & -0.039714 \\
0.010330 & 0.032154 & 0.000000 & -0.059100 & -0.005204 & -0.150847 & -0.293117 & -0.041182 \\
-0.001572 & 0.023705 & 0.059100 & 0.000000 & -0.210070 & -0.008831 & 0.155849 & -0.023644 \\
-0.017053 & 0.041653 & 0.005204 & 0.210070 & 0.500000 & 0.212440 & 0.038713 & 0.006219 \\
-0.012344 & -0.014959 & 0.150847 & 0.008831 & -0.212440 & 0.000000 & 0.055970 & 0.025358 \\
0.008076 & -0.091371 & -0.206883 & -0.155849 & -0.038713 & -0.055970 & 0.000000 & 0.048148 \\
0.002015 & 0.039714 & 0.041182 & 0.023644 & -0.006219 & -0.025358 & -0.048148 & 0.000000
\end{bmatrix}$$

8.1.3 penTIPG 3 with $\Lambda = 0.01$

$$S = \{0.005, 0.007, 0.01, 0.125, 0.51000000000001, 0.75, 1.01, 1.02\},$$

$$\text{truncation} = 6,$$

$$T = \{0.1, 0.3, 0.5, 1, 1.5, 2, 3, 4, 10, 1000\},$$

Matrix v_*:

$$\begin{bmatrix} 0.000000 & -0.011971 & -0.006441 & 0.006342 & -0.062810 & 0.058173 & 0.025592 & -0.029527 \\ 0.011971 & 0.000000 & -0.010990 & -0.001980 & -0.012568 & -0.046571 & 0.105927 & -0.059991 \\ 0.006441 & 0.010990 & 0.000000 & -0.017410 & 0.036784 & -0.126314 & -0.342226 & -0.079367 \\ -0.006342 & 0.001980 & 0.017410 & 0.000000 & -0.104147 & 0.024187 & 0.066522 & -0.025626 \\ 0.062810 & 0.012568 & -0.036784 & 0.104147 & 0.500000 & 0.243379 & 0.068062 & 0.013066 \\ -0.058173 & 0.046571 & 0.126314 & -0.024187 & -0.243379 & 0.000000 & 0.145096 & -0.017206 \\ -0.025592 & -0.105927 & -0.157774 & -0.066522 & -0.068062 & -0.145096 & 0.000000 & 0.118873 \\ 0.029527 & 0.059991 & 0.079367 & 0.025626 & -0.013066 & 0.017206 & -0.118873 & 0.000000 \end{bmatrix}$$

In Figure 2, we give a graphical depiction of v_*.

9 Other Cheat-Penalised WCF Protocols

In this section, we give an overview of a few other cheat-penalised protocols for WCF with the purpose of comparing them to the protocols found through our method.

9.1 Spekkens-Rudolph

In [SR02] a cheat-sensitive protocol was proposed with a bias close to 0.207. The main idea is that Alice prepares a quantum state on two registers and sends one of them to Bob. Then, based on a measurement by Bob on the state, a party is chosen to verify the state on both registers. If the verification succeeds, the party who did not perform the verification is declared the winner. This construction can be naturally extended to the cheat-penalised setting. We summarise the protocol below.

- Alice prepares $|\psi\rangle = \sqrt{x}\,|00\rangle + \sqrt{1-x}\,|11\rangle \in D(\mathcal{AB})$ and sends $\mathcal{B}$ to Bob.
- Bob measures $\mathcal{B}$ with the measurement (E_0, E_1) to get b, where $E_0 = \frac{1}{2x}|0\rangle\langle 0|$, for $x \in (1/2, 1]$. He sends b to Alice.
- If $b = 0$, he sends $\mathcal{B}$ back to Alice. Alice measures the POVM $\{|\psi_0\rangle\langle\psi_0|, I - |\psi_0\rangle\langle\psi_0|\}$. If the result is $|\psi_0\rangle\langle\psi_0|$, Bob wins. If not, she declares Bob a cheater.
- If $b = 1$, Alice sends $\mathcal{A}$ to Bob. He measures the POVM $\{|\psi_1\rangle\langle\psi_1|, I - |\psi_1\rangle\langle\psi_1|\}$. If the result is $|\psi_1\rangle\langle\psi_1|$, Alice wins. If not, he declares Alice a cheater.

Where we define $|\psi_b\rangle = \frac{(I\otimes\sqrt{E_b})|\psi\rangle}{\sqrt{\langle\psi_b|I\otimes E_b|\psi\rangle}}$.

In the weak coin-flipping setting studied by [SR02], the optimal bias of $\frac{1}{\sqrt{2}} - \frac{1}{2} \approx 0.207$ is achieved by choosing $x = \frac{1}{\sqrt{2}}$. In the cheat-penalised setting, the semidefinite programs characterising the optimal strategies for cheating Alice and cheating Bob are given as follows.

SDP cheating Alice.

$$\max_{\rho_{AB}} \quad (\Lambda+1)\cdot \mathrm{Tr}\left[(I^A\otimes\sqrt{E_1})\rho_{AB}(I^A\otimes\sqrt{E_1})\,|\psi_1\rangle\langle\psi_1|\right] + \Lambda\cdot\mathrm{Tr}\left[\rho_{AB}(I^A\otimes E_0)\right]$$

subject to

$$Tr_{AB}\left[\rho_{AB}\right] = 1$$

$$\rho_{AB} \geq 0.$$

SDP cheating Bob

$$\max_{\rho_{ABC}} \quad (\Lambda+1)\cdot \mathrm{Tr}\left[(|\psi_0\rangle\langle\psi_0|\otimes|0\rangle\langle 0|_C)\rho_{ABC}\right] + \Lambda\cdot \mathrm{Tr}\left[(I^{AB}\otimes|1\rangle\langle 1|_C)\rho_{ABC}\right]$$

subject to

$$\mathrm{Tr}_{BC}\left[\rho_{ABC}\right] = \mathrm{Tr}_B\left[|\psi\rangle\langle\psi|\right]$$

$$\rho_{ABC} \geq 0.$$

9.1.1 Point game

We consider the Spekkens-Rudolph point game as in [Moc07] but with the starting point as defined for cheat-penalised WCF, i.e., with coordinates w and v such that $w > v > 0$.

$$\frac{1}{2}[\![v,w]\!] + \frac{1}{2}[\![w,v]\!] \quad \underset{\rightarrow}{\text{h. split}} \quad \frac{1}{2}[\![v,w]\!] + p\,[\![z_1,v]\!] + (\frac{1}{2}-p)\,[\![z_2,v]\!] \tag{52}$$

$$\underset{\rightarrow}{\text{v. raise}} \quad \frac{1}{2}[\![v,w]\!] + p\,[\![z_1,v]\!] + (\frac{1}{2}-p)\,[\![z_2,w]\!] \tag{53}$$

$$\underset{\rightarrow}{\text{h. merge}} \quad (1-p)\left[\!\!\left[\frac{\frac{v}{2}+(1/2-p)z_2}{1-p}\ ,\ w \right]\!\!\right] + p\,[\![z_1,v]\!] \tag{54}$$

$$\underset{\rightarrow}{\text{v. merge}} \quad [\![z_1,(1-p)w+pv]\!] \tag{55}$$

where due to the splits and merge we have the conditions

$$\frac{1}{2w} \geq \frac{p}{z_1} + \frac{\frac{1}{2}-p}{z_2}, \tag{56}$$

$$z_1 = \frac{\frac{v}{2}+(1/2-p)z_2}{1-p}, \tag{57}$$

$$0 < z_1 < 1, \tag{58}$$

$$z_2 > 1, \tag{59}$$

$$0 < p < 1. \tag{60}$$

If we solve for $z_1 = (1-p)w + pv$ under these constraints, we can find the cheating reward. By solving these equations using a non-linear optimisation routine we find that $x = 0.653$ gives the optimal measurement in the SDPs. This can be also checked by solving the SDPs and finding the x at which both Alice's and Bob's cheating bias are the same as shown in Figure 10. The value for optimal cheating bias is 0.152. The number of messages in this protocol is 8 and based on Theorem 34 the number of qubits is 6.

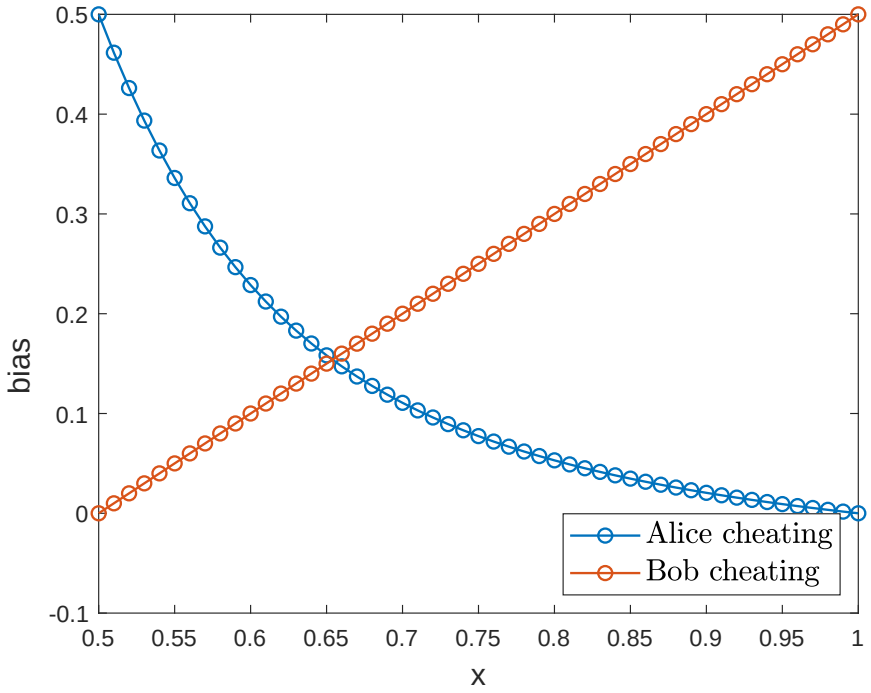


Figure 10: Bias for Spekkens-Rudolph protocol when $\Lambda = 6$. The intersection of the curves for the bias of Alice and Bob give the optimal x for the protocol.

9.2 Dip-Dip-Boom

The point game for the Spekkens-Rudolph protocol can be refined, as described in Section 3.2.2 of [Moc07], by increasing the number of points generated in the initial splits (see Figure 5 in [Moc07]). This point game characterises what is known as the Dip-Dip-Boom protocol. Mochon analyses the cheating reward in the cheat-penalised setting in the limit where the number of initial points tends to infinity. From his analysis, he derives a bound on the cheating reward. While we do not reproduce the full derivation here, we provide additional details relevant to our discussion.

Mochon works in the setting where winning the coin flip gives 1 point, loosing gives 0 points and if you get caught cheating you lose Λ points. From the starting frame $\frac{1}{2}[\![1,0]\!] + \frac{1}{2}[\![0,1]\!]$ we want to find the minimal R^* such that $[\![R^*, R^*]\!]$ can be reached by this protocol. The initial splits generate a continuous probability distribution of points of the form

$$\frac{1}{2}\int_{R^*}^{\infty} p(z)\,[\![z,0]\!]\,\mathrm{d}z + \frac{1}{2}\int_{R^*}^{\infty} p(z)\,[\![0,z]\!]\,\mathrm{d}z. \tag{61}$$

By conservation of probability, it is shown that $p(z) = \frac{2(R^*)^2}{z^3}$ to obtain a valid transition. Finally, to go from the initial frame to the frame in Equation 61, the following inequality must be satisfied (Equation 101 in [Moc07])

$$-\frac{1}{\Lambda+1} \leq -\left(\frac{\Lambda - 2R^*}{\Lambda^2} + \frac{2(R^*)^2}{\Lambda^3}\log\left(\frac{R^*+\Lambda}{R^*}\right)\right). \tag{62}$$

Mochon [Moc07] argues that this is the correct constraint corresponding to a split in the cheat-penalised setting, in accordance with his rewards convention.[9] Finally, we show the following claim giving the scaling

[9]Note that our convention, while equivalent in terms of constructing protocols, is different from Mochon's: we award $\Lambda + 1$ points on winning the coin flip, Λ on losing the coin flip and 0 on getting caught cheating.

of R^* when $\Lambda \to \infty$. In [Moc07], this was stated without proof.

Claim 47 (Section 3.3.2 in [Moc07]). As $\Lambda \to \infty$, the optimal cheating reward for the Dip-Dip-Boom protocol is $R^* = \frac{1}{2} + \frac{\log \Lambda}{4\Lambda} + O(\frac{1}{\Lambda})$.

Proof. Throughout the proof we assume the equality in Equation 62 holds. To see that equality is enough, note that we can write Equation 62 as

$$H_\Lambda(z) := \Lambda(\Lambda - 2z) + 2z^2 \log\left(1 + \frac{\Lambda}{z}\right) \;\leq\; \frac{\Lambda^3}{\Lambda + 1}. \tag{63}$$

For each fixed $\Lambda > 0$, the function $H_\Lambda(z)$ is continuous and strictly decreasing in $z > 0$, with

$$H_\Lambda(0^+) = \Lambda^2, \qquad H_\Lambda(z) \to 0 \quad (z \to \infty).$$

Thus, there exists a unique $R^*(\Lambda) > 0$ such that

$$H_\Lambda(R^*(\Lambda)) = \frac{\Lambda^3}{\Lambda + 1}.$$

At this value the inequality holds with equality, and strictly for $z > R^*(\Lambda)$. Therefore, we can consider Equation 62 as an equality for finding a lower bound to the cheating reward.

Consider the expansion

$$\frac{1}{\Lambda + 1} = \frac{1}{\Lambda} - \frac{1}{\Lambda^2} + O\left(\frac{1}{\Lambda^3}\right). \tag{64}$$

Then, multiplying both sides of Equation 62 by Λ^2 gives

$$\Lambda - 1 + O\left(\frac{1}{\Lambda}\right) = \Lambda - 2R^* + \frac{2(R^*)^2 \log\left(1 + \frac{\Lambda}{R^*}\right)}{\Lambda}. \tag{65}$$

We then have the equality

$$1 + O\left(\frac{1}{\Lambda}\right) = 2R^* - \frac{2(R^*)^2 \log\left(1 + \frac{\Lambda}{R^*}\right)}{\Lambda}. \tag{66}$$

As $\Lambda \to \infty$, we can consider $R^* = \frac{1}{2} + \delta$ with $\delta = o(1)$. Then, for large Λ we have $\log\left(1 + \frac{\Lambda}{R^*}\right) \approx \log\left(\frac{\Lambda}{R^*}\right) = \log(\Lambda) - \log(R^*)$ and moreover

$$\log R^* = \log\left(\frac{1}{2} + \delta\right) \tag{67}$$

$$= \log\left(\frac{1}{2}\right) + \log(1 + 2\delta) \tag{68}$$

$$= \log\left(\frac{1}{2}\right) + 2\delta + O(\delta^2). \tag{69}$$

Therefore, we have that $\log\left(1 + \frac{\Lambda}{R^*}\right) \approx \log(\Lambda) + \log(2) - 2\delta + O(\delta^2)$. Note that in this last expression any smaller orders in $1/\Lambda$ will scale as $O\left(\frac{1}{\Lambda}\right)$. Furthermore, $2(R^*)^2 = \frac{1}{2} + 2\delta + 2\delta^2$ and thus,

$$\frac{2(R^*)^2 \log\left(1 + \frac{\Lambda}{R^*}\right)}{\Lambda} = \frac{(\frac{1}{2} + 2\delta)(\log \Lambda + \log 2 - 2\delta)}{\Lambda} + O\left(\frac{\delta^2 \log \Lambda}{\Lambda}\right) \tag{70}$$

We shall neglect terms of order δ^2 as these terms are small when $\Lambda \to \infty$. From Equation 66, we have

$$1 + O\left(\frac{1}{\Lambda}\right) = 1 + 2\delta - \frac{2(R^*)^2 \log\left(1 + \frac{\Lambda}{R^*}\right)}{\Lambda}. \tag{71}$$

$$\approx 1 + 2\delta - \frac{\log \Lambda}{2\Lambda} - \frac{\log 2}{2\Lambda} + \frac{\delta}{\Lambda} - \frac{2\delta \log \Lambda}{\Lambda} - \frac{2\delta \log 2}{\Lambda} \tag{72}$$

In the last expression, the leading term is $\frac{\log \Lambda}{2\Lambda}$. Therefore, if we want this expression to match the order $O\left(\frac{1}{\Lambda}\right)$, we need to at least cancel this higher order term with δ. We must then have

$$2\delta = \frac{\log \Lambda}{2\Lambda} + O\left(\frac{1}{\Lambda}\right). \tag{73}$$

This shows that $R^* = \frac{1}{2} + \frac{\log \Lambda}{4\Lambda} + O(\frac{1}{\Lambda})$. □

In fact, we can show the higher-order scaling of R^*. We provide this result next.

Claim 48. As $\Lambda \to \infty$, the optimal cheating reward for the Dip-Dip-Boom protocol is $R^* = \frac{1}{2} + \frac{\frac{1}{4}\log(2\Lambda) - \frac{1}{2}}{\Lambda} + \frac{\frac{1}{4}\log^2(2\Lambda) - \frac{5}{8}\log(2\Lambda) + \frac{7}{8}}{\Lambda^2} + O\left(\frac{\log^3 \Lambda}{\Lambda^3}\right)$.

Proof. Consider the following expression for R^*,

$$R^* = \frac{1}{2} + \frac{A \log \Lambda + B}{\Lambda} + \frac{C \log^2 \Lambda + D \log \Lambda + E}{\Lambda^2} + O\left(\frac{\log^3 \Lambda}{\Lambda^3}\right). \tag{74}$$

We need to find A, B, C, D and E such that equality in Equation 62 is solved up to the relevant order. For simplicity, define $u_1 = A \log \Lambda + B$, $u_2 = C \log^2 \Lambda + D \log \Lambda + E$, $\delta = R^* - \frac{1}{2}$. Therefore, we have the following expansions.

$$\frac{1}{\Lambda + 1} = \frac{1}{\Lambda} - \frac{1}{\Lambda^2} + \frac{1}{\Lambda^3} - \frac{1}{\Lambda^4} + O(\Lambda^{-5}) \tag{75}$$

$$R^* = \frac{1}{2} + \frac{u_1}{\Lambda} + \frac{u_2}{\Lambda^2} + O(\Lambda^{-3}) \tag{76}$$

$$(R^*)^2 = \frac{1}{4} + \frac{1}{4} + \frac{u_1}{\Lambda} + \frac{u_2 + u_1^2}{\Lambda^2} + O(\Lambda^{-3}) \tag{77}$$

$$\log R^* = \log \frac{1}{2} + 2\delta - 2\delta^2 + O(\delta^3) \tag{78}$$

$$= \log \frac{1}{2} + \frac{2u_1}{\Lambda} + \frac{2u_2 - 2u_1^2}{\Lambda^2} + O(\Lambda^{-3}). \tag{79}$$

By replacing these expansions in Equation 62 and imposing equality of left and right hand side, we can match the orders and obtain coefficients A, B, C, D, E. The result is the expression

$$R^* = \frac{1}{2} + \frac{\frac{1}{4}\log(2\Lambda) - \frac{1}{2}}{\Lambda} + \frac{\frac{1}{4}\log^2(2\Lambda) - \frac{5}{8}\log(2\Lambda) + \frac{7}{8}}{\Lambda^2} + O\left(\frac{\log^3 \Lambda}{\Lambda^3}\right).$$

□

9.3 ABDR04 protocol

The Dip-Dip-Boom protocol analysed in Section 9.2 demonstrates that, when both the penalty and the number of rounds are taken to the infinite limit, the cheating reward converges to $\frac{1}{2}$. In contrast, the cheat-penalised protocol introduced in [ABDR04] establishes that it is sufficient to take only the penalty to the infinite limit in order to achieve this bound. This protocol, which we describe below, requires only 3 messages between Alice and Bob. Consider $\Lambda \geq 4$ and $\delta = \frac{2}{\sqrt{\Lambda}}$. Define the state $|\psi_a\rangle = \sqrt{\delta}\,|a\rangle\,|a\rangle + \sqrt{1-\delta}\,|2\rangle\,|2\rangle \in \mathbb{C}^3 \otimes \mathbb{C}^3$ with $a \in \{0,1\}$.

- Alice picks $a \in \{0,1\}$ uniformly at random and prepares $|\psi_a\rangle \in \mathcal{AB}$ and sends $\mathcal{B}$ to Bob.
- Bob picks $b \in \{0,1\}$ uniformly at random and sends b to Alice.
- Alice sends a and the register $\mathcal{A}$ to Bob. Then Bob measures $\mathcal{AB}$ with the POVM $\{|\psi_a\rangle\langle\psi_a|\,, I - |\psi_a\rangle\langle\psi_a|\}$. If Bob measures the state $|\psi_a\rangle$, the verification succeeds and the coin flip is $a \oplus b$, otherwise Bob declares that Alice cheated.

As demonstrated in [ABDR04], the cheating reward for both Alice and Bob is given by $\frac{1}{2} + \frac{1}{\sqrt{\Lambda}}$. We do not describe the point games in this case as they are not relevant here but they can be constructed using known techniques, if needed. As discussed previously, the protocol necessitates the use of two qutrits or, alternatively, four qubits.

10 Comparison of penWCF Protocols

We compare previously known protocols with those obtained in our work in terms of the bias achieved relative to the number of communication rounds and the number of qubits required. We use for the comparison penTIPG 2 which yields protocol (iii) and penTIPG 3 which yields protocols (i) and (ii) depending on the choice of parameters (more specifically on the choice of δ).

Round complexity. Figure 11 illustrates this comparison for $\Lambda \in \{1, 0.01\}$. Our protocols achieve a bias below the bound of the cheat-penalised version of Dip-Dip-Boom protocol, whose performance is derived under the assumption of infinitely many communication rounds between Alice and Bob. While the ABDR04 protocol can in principle reach arbitrarily low bias, it does so only in the limit of infinite penalty. By contrast, our results demonstrate that comparably low bias can already be achieved with finite penalty, albeit at the cost of a large number of communication rounds. Despite this large resource cost, the number of rounds in our protocol is still significantly smaller than in Mochon's TIPGs for WCF achieving bias $\frac{1}{2} + O(\epsilon)$. In particular, the lower bound from [Mil20] shows that the number of rounds required scales as $\exp(\Omega(1/\sqrt{\epsilon}))$. Previous estimations [AM25] for the constants in the lower bound indicate that to obtain a bias of $\epsilon = 10^{-8}$ one would require at least 10^{23} rounds. With protocol (i), we can reach a bias of 10^{-8} with 10^{16} rounds. Moreover, to reach a bias of 10^{-10} we only need 10^{18} rounds, which is still 10^5 times better in communication complexity compared to the lower bound with $\epsilon = 10^{-8}$.

Space complexity. Figure 1 compares bias against the number of qubits used in each protocol. Here, our numerics provide striking evidence that arbitrarily low bias can be obtained using only a constant number of qubits, even when the penalty is kept small. All the protocols in our work require 24 qubits, protocol (ii) based on penTIPG 3 can reach down to bias $\epsilon = 10^{-10}$. This reveals a sharp distinction from prior approaches: the Dip-Dip-Boom and ABDR04 protocol require infinite penalty. Note also that we are considering the Dip-Dip-Boom protocol in the "infinite limit" of points in the point game, i.e., infinite rounds. Mochon [Moc07] also constructs other families TIPGs for WCF that achieve a bias of $\frac{1}{2} + O(\epsilon)$ in the weak coin-flipping setting.

However, these TIPGs require an unbounded number of qubits as the bias approaches $\frac{1}{2}$. Our findings therefore highlight that our penTIPG-based constructions can achieve low bias with genuinely finite resources, suggesting a new route to practical protocols.

Given the previous considerations, a natural question is whether our algorithm can produce protocols achieving arbitrarily low bias when $\Lambda = 0$. Such a result would establish the existence of weak coin-flipping protocols that are more space-efficient than Dip-Dip-Boom and Mochon's TIPGs reaching a bias $\frac{1}{2} + (\epsilon)$. Our preliminary numerical evidence, however, suggests that this may not be the case. In particular, we were unable to find solutions close to the $\Lambda = 0$ regime when the chosen grid (i.e., the set S) excluded points below $[\![\Lambda, \Lambda]\!]$. This indicates that the ability to approach arbitrarily low bias appears to require access to such points in the grid.

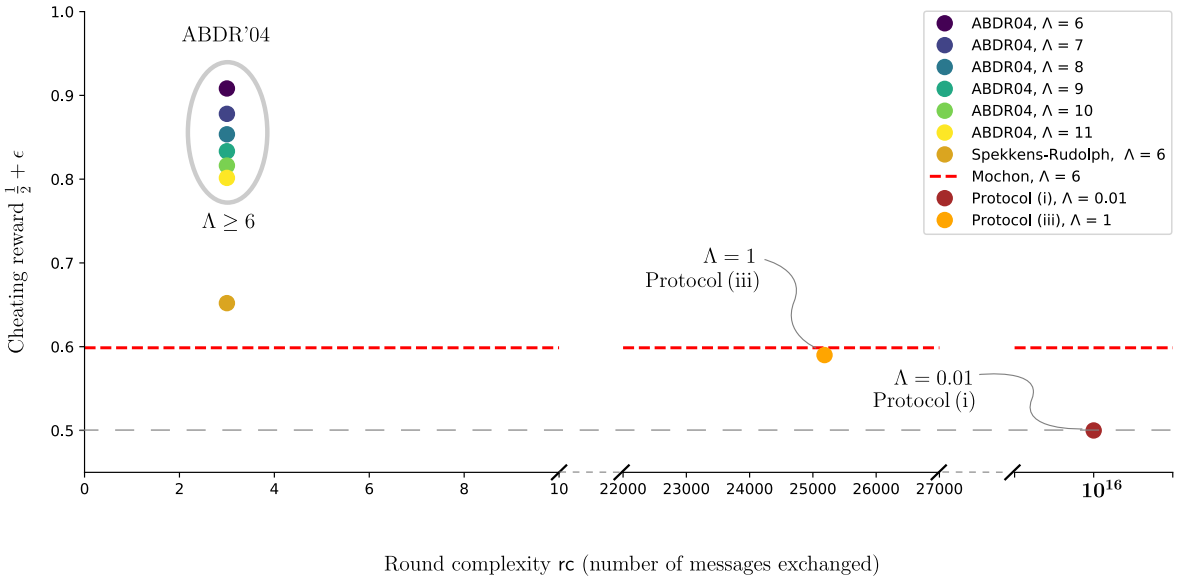


Figure 11: Comparison of cheat-penalised weak coin-flipping protocols in terms of the reward and the number of messages. We compare our protocols penTIPG 2 and penTIPG 3 to ABDR04 [ABDR04], Spekkens-Rudolph protocol [SR02] and Mochon's bound [Moc07]. bias and number of qubits. The protocol penTIPG 2 corresponds to Protocol (iii)—our potentially amenable to experiment protocol, with $\Lambda = 1$. The protocol penTIPG 3 corresponds to protocol (i).

Acknowledgements

ASA acknowledges support from the U.S. Department of Defense through a QuICS Hartree Fellowship and from IIIT Hyderabad. MESM acknowledges support from the U.S. Department of Defense through a QuICS Hartree Fellowship. JS is funded in part by the Commonwealth of Virginia's Commonwealth Cyber Initiative (CCI) under grant number 469351.

References

[ABDR04] Andris Ambainis, Harry Buhrman, Yevgeniy Dodis, and Hein Roehrig. Multiparty quantum coin flipping. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, pages 250–259, 2004.

[ACG+14] Dorit Aharonov, André Chailloux, Maor Ganz, Iordanis Kerenidis, and Loïck Magnin. A simpler proof of existence of quantum weak coin flipping with arbitrarily small bias. *SIAM Journal on Computing*, 45(3):633–679, 2014.

[AM25] Yusuf Alnawakhtha and Carl A Miller. A note on the impossibility of efficient quantum weak coin flipping. https://atulsingharora.github.io/penWCF/, 2025.

[Amb04] Andris Ambainis. A new protocol and lower bounds for quantum coin flipping. *Journal of Computer and System Sciences*, 68(2):398–416, 2004.

[AMMS25] Atul Singh Arora, Carl A Miller, Mauro E.S. Morales, and Jamie Sikora. GitHub page for "Cheat-penalised quantum weak coin flipping". https://atulsingharora.github.io/penWCF, 2025.

[ARV21] Atul Singh Arora, Jérémie Roland, and Chrysoula Vlachou. Analytic quantum weak coin flipping protocols with arbitrarily small bias. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pages 919–938, 2021.

[ARVW22] Atul Singh Arora, Jérémie Roland, Chrysoula Vlachou, and Stephan Weis. Solutions to quantum weak coin flipping. Cryptology ePrint Archive, Paper 2022/1101, 2022.

[ARVW24] Atul Singh Arora, Jérémie Roland, Chrysoula Vlachou, and Stephan Weis. Protocols for quantum weak coin flipping. arXiv preprint arXiv:2402.15855, 2024.

[ARW19] Atul Singh Arora, Jérémie Roland, and Stephan Weis. Quantum weak coin flipping. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 205–216, 2019.

[ATSVY00] Dorit Aharonov, Amnon Ta-Shma, Umesh V Vazirani, and Andrew C Yao. Quantum bit escrow. In *Proceedings of the thirty-second annual ACM Symposium on Theory of Computing*, pages 705–714, 2000.

[BB84] Charles H. Bennett and Gilles Brassard. Public-key distribution and coin tossing. In *Int. Conf. on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[BCWW24] Mathieu Bozzio, Claude Crépeau, Petros Wallden, and Philip Walther. Quantum cryptography beyond key distribution: theory and experiment. arXiv preprint arXiv:2411.08877, 2024.

[CGS16] André Chailloux, Gus Gutoski, and Jamie Sikora. Optimal bounds for semi-honest quantum oblivious transfer. *Chicago Journal of Theoretical Computer Science*, 13:1–17, 2016.

[CK09] André Chailloux and Iordanis Kerenidis. Optimal quantum strong coin flipping. In *the 50th annual IEEE Symposium on Foundations of Computer Science*, pages 527–533, 2009.

[CK11] André Chailloux and Iordanis Kerenidis. Optimal bounds for quantum bit commitment. In *the 52nd annual Symposium on Foundations of Computer Science*, pages 354–362, 2011.

[CK17] André Chailloux and Iordanis Kerenidis. Physical limitations of quantum cryptographic primitives or optimal bounds for quantum coin flipping and bit commitment. *SIAM Journal on Computing*, 46(5):1647–1677, 2017.

[Gan17] Maor Ganz. Quantum leader election. *Quantum Information Processing*, 16(3):73, 2017.

[HP13] Peter Høyer and Edouard Pelchat. Point Games in Quantum Weak Coin Flipping Protocols. Master's thesis, University of Calgary, 2013.

[LC97] Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410, 1997.

[LC98] Hoi-Kwong Lo and Hoi Fung Chau. Why quantum bit commitment and ideal quantum coin tossing are impossible. *Physica D: Nonlinear Phenomena*, 120(1-2):177–187, 1998.

[Lo97] Hoi-Kwong Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154, 1997.

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414, 1997.

[Mil20] Carl A Miller. The impossibility of efficient quantum weak coin flipping. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 916–929, 2020.

[Mil22] Carl A Miller. The mathematics of quantum coin-flipping. *Notices of the American Mathematical Society*, 69(11), 2022.

[MNS09] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In *Theory of Cryptography Conference*, pages 1–18, 2009.

[Moc04] Carlos Mochon. Quantum weak coin-flipping with bias of 0.192. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11, 2004.

[Moc05] Carlos Mochon. Large family of quantum weak coin-flipping protocols. *Phys. Rev. A*, 72:022341, 2005.

[Moc07] Carlos Mochon. Quantum weak coin flipping with arbitrarily small bias. *arXiv:0711.4114*, 2007.

[NST14] Ashwin Nayak, Jamie Sikora, and Levent Tunçel. A search for quantum coin-flipping protocols using optimization techniques. *Mathematical Programming*, 156(1-2):581–613, 2014.

[SR01] Robert W Spekkens and Terry Rudolph. Degrees of concealment and bindingness in quantum bit commitment protocols. *Physical Review A*, 65(1):012310, 2001.

[SR02] Robert W Spekkens and Terry Rudolph. Quantum protocol for cheat-sensitive weak coin flipping. *Physical Review Letters*, 89(22):227901, 2002.

[WEH18] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412):eaam9288, 2018.