Rational Arguments: Single Round Delegation with Sublinear Verification

Siyao Guo[*] Pavel Hubáček[†] Alon Rosen[‡] Margarita Vald[§]

September 30, 2013

Abstract

Rational proofs, recently introduced by Azar and Micali (STOC 2012), are a variant of interactive proofs in which the prover is neither honest nor malicious, but rather rational. The advantage of rational proofs over their classical counterparts is that they allow for extremely low communication and verification time. Azar and Micali demonstrated their potential by giving a one message rational proof for #SAT, in which the verifier runs in time $O(n)$, where n denotes the instance size. In a follow-up work (EC 2013), Azar and Micali proposed "super-efficient" and interactive versions of rational proofs and argued that they capture precisely the class TC0 of constant-depth, polynomial-size circuits with threshold gates.

In this paper, we show that by considering *rational arguments*, in which the prover is additionally restricted to be computationally bounded, the class NC1, of search problems computable by log-space uniform circuits of $O(\log n)$-depth, admits rational protocols that are simultaneously one-round and polylog(n) time verifiable. This demonstrates the potential of rational arguments as a way to extend the notion of "super-efficient" rational proofs beyond the class TC0.

The low interaction nature of our protocols, along with their sub-linear verification time, make them well suited for delegation of computation. While they provide a weaker (yet arguably meaningful) guarantee of soundness, they compare favorably with each of the known delegation schemes in at least one aspect. They are simple, rely on standard complexity hardness assumptions, provide a correctness guarantee for all instances, and do not require preprocessing.

Our rational arguments are constructed in two steps. We first design a multi-round rational proof and then collapse it into a single round argument. In doing so, we identify the gap between the reward given to a truthful prover in the proof and the one given to a dishonest one as a key parameter in the quality of the resulting argument. We leave it as an intriguing open problem to determine whether one could "scale up" the underlying proofs to classes beyond NC1, and potentially even beyond P (thus bypassing known impossibility results), and point out some limitations in doing so for non-trivial languages.

[*] Department of Computer Science and Engineering, Chinese University of Hong Kong. Email: syguo@cse.cuhk.edu.hk. Work supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307952.

[†] Department of Computer Science, Aarhus University, Denmark. Email: hubacek@cs.au.dk. The author acknowledges support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed. The author did part of this work while visiting IDC Herzliya supported by ISF grant no. 334/08.

[‡] Efi Arazi School of Computer Science, IDC Herzliya, Israel. Email: alon.rosen@idc.ac.il. Work supported by ISF grant no. 334/08 and by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307952.

[§] The Blavatnik School of Computer Science, Tel Aviv University, Israel. Email: margarita.vald@cs.tau.ac.il. Work supported by ISF grant no. 334/08.

1 Introduction

With the advent of cloud computing there has been increased interest in schemes that allow devices with limited computing power to verify the correctness of computations done on their behalf. Such procedures are referred to as *delegation schemes*. In their most basic incarnation, the weak device (the *verifier*) delegates a computation to the cloud (the *prover*) by sending it an input $x \in \{0,1\}^n$ and expecting the cloud to return the value $f(x)$, where f is a function computable by a predetermined circuit C. To make the computation of $f(x)$ verifiable, the prover is expected to send along a "certificate of correct computation", which is then checked by the verifier, preferably investing less resources than it would have invested had he computed $f(x)$ on its own (for this notion to be meaningful the circuit C should be given in some succinct way, e.g. it can be efficiently generated by a small program).

1.1 Efficient Proofs/Arguments

The first approaches for delegating computation (which in fact pre-dated the cloud days by about two decades) proposed to use Computationally Sound proofs/efficient arguments (cf. Kilian [Kil92], Micali [Mic00]). These delegation schemes allow the verifier to run in time $O(\text{poly}(n, \log S))$, where S denotes f's circuit size. However, they require heavy machinery such as PCP for general circuits, and hence introduce significant overhead to both the prover and the verifier. Moreover, efficient arguments require four messages of interaction between the prover and the verifier, inflicting undesirable burden on the parties.

While the interaction can be removed by using Random Oracles [Mic00], the question of whether there exist efficient non-interactive arguments under standard assumptions is still not well understood. One indication on the difficulty of resolving this problem is that no non-trivial language has a Succinct Non-interactive ARGument (SNARG) based on non-falsifiable assumptions (Gentry and Wichs [GW11]). Several works have been bypassing these limitations by relying on non-standard (and in particular not known to be falsifiable) assumptions (we survey these alternative proposals in Section 1.4), yet it is of interest to examine to what extent such assumptions are necessary for succinct delegation.

In any case, as observed by Goldwasser, Kalai and Rothblum [GKR08], the impossibility results do not pose any limitation in the context of real-life delegation. This is simply because in delegation we are only interested in efficiently computable problems. This brought them to introduce a weaker notion of delegation, in which the prover is only required to work for languages computable in polynomial time, resulting in more efficient delegation schemes, both in the standard model (i.e., with no set-up), or in a model allowing an off-line preprocessing stage (used to amortize the cost of verification across multiple evaluations of f). One-round versions of such schemes are computationally sound, and in particular based on intractability assumptions (e.g. [CKV10, GGP10, AIK10]), whereas interactive versions can be actually proved to be unconditionally sound [GKR08].

It should be noted that, while many of these more recent schemes do not rely on full blown PCP machinery, they still induce significant computational overhead on the prover. Even leaving prover's complexity aside, in all the delegation schemes mentioned above the verifier's running time is $O(\text{poly}(n, \log S))$ (or at best $n + \text{polylog}(S)$), suggesting that the least the verifier should do is to read the input in its entirety. On the other hand, recent work by Rothblum, Vadhan and Wigderson [RVW13], suggests that this apparent limitation can be actually bypassed by considering "Interactive Proofs of Proximity". For circuits of depth d, such proofs allow the verifier to work in time $(n/D + D)^{1+o(1)} \cdot \tilde{O}(d)$ (which is sub-linear for example when $D = \sqrt{n}$), but to reject only strings that are D-far from being the correct output of the computation.

1.2 Rational Proofs

Another recent line of work, initiated by Azar and Micali [AM12] proposes to relax the rigid soundness requirements of classical proofs. They do so by allowing the verifier to accept proofs of false statements, but make it *irrational* for the prover to provide them. While not good enough in case of a malicious prover (or say if one wants to detect faults), this still may fit very well a scenario in which a weak client delegates a complex computation to a more powerful server/cloud (think of delegation of some massive computation to Amazon, which would then be paid for its services). Thus, to the extent that such *Rational Proofs* are meaningful, one may consider them as an alternative to more traditional approaches to delegation.

Azar and Micali [AM12] give rational proofs for search problems in #P and more generally for the entire counting hierarchy. The main advantage of such proofs over their classical counterparts is that they are as simple as one could hope for. The basic idea is to let the prover simply give the verifier the result of the computation, and let the verifier randomly compute a "reward" based on the prover's answer. This reward reflects the quality of the prover's computation in that its expectation is maximized when the prover gives the correct answer. As a result there is almost no overhead both for the prover and for the verifier.

One natural question that comes to mind is whether rational proofs can be "scaled down" to lower complexity classes. This includes both classes such as NP, but also lower classes say within P (i.e., ones that are relevant to delegation). Towards this end, Azar and Micali [AM13] proposed "super-efficient" rational proofs, an extension of their notion, and showed that such super-efficient rational proofs capture precisely the class TC0 of *constant depth*, polynomial size circuits with threshold gates. The super-efficiency of their proofs, while introducing additional interaction, allows the reward to be computed by the verifier in sub-linear time (in the input size). This can be achieved in an extremely simple and natural manner, introducing hardly any overhead on the prover's work (beyond the computation of f).

Azar and Micali [AM13] also consider super-efficient one-round rational PCPs, in which the communication is poly(n) and the verifier has random access to the message from prover. Rational PCPs can capture languages beyond TC0. However, given the differences in the communication model and constraints, they are not directly comparable to super-efficient proofs. Moreover rational PCPs seem unsuitable for delegation, since in this model the prover sends the whole proof to the verifier that queries only a limited number bits of it to check the correctness of computation.

1.3 Our Results

The main objective of this paper is to attain a simple (and in particular low-overhead), one-round delegation scheme that captures as large class of problems as possible, and in which the verifier works sub-linearly in the input size. To this end we introduce a new concept, called *Rational Arguments*, which beyond postulating a rational prover additionally assumes that he is computationally bounded. We show how to realize rational arguments based on standard (falsifiable) cryptographic hardness assumptions, resulting in an arguably meaningful alternative to known delegation schemes.

Our initial observation is that for the class of search problems computable by threshold circuits of depth $d = o(n)$, the efficiency of rational proofs can be strengthened beyond what was originally shown by Azar and Micali [AM12], in the sense that the verification time can be performed in as little as $\tilde{O}(d)$ time. This comes at the cost of making the proof interactive, and specifically requires d rounds of communication. The resulting rational proof is very simple and induces hardly any overhead. All that is required from the prover is to perform the computation as prescribed, storing intermediate values along the way. A small subset of these values is subsequently used to convince the verifier in the correctness of the actual computation. Moreover, the verification procedure requires the verifier to only access a *single* bit of the input, and to perform a computation with complexity proportional to d, the depth of the circuit.

Our rational proofs resemble by structure the super-efficient rational proofs of Azar and Micali [AM13], and traverse the circuit in a similar manner. However, the computation of reward by the verifier differs in that it requires the prover to send a single bit per gate (the corresponding intermediate value), unlike in the case of [AM13] where the prover needs to count and report the number of ones among the children of each gate queried by the verifier. This yields to substantial improvement in terms of communication complexity.

Next, we go on and convert the above interactive procedure into a one with a *single round*. This is done by applying a variant of a previously known transformation by Kalai and Raz [KR09]. This transformation makes use of a single server Private Information Retrieval Scheme (PIR) and assumes that the prover is computationally bounded. The "database" held by the prover in this transformation consists of precisely the intermediate computation values mentioned above, and as a result the overhead is kept minimal (the Kalai Raz transformation was previously used to convert significantly "heavier" protocols, resulting in increased work for the prover, beyond the mere computation of f). Jumping ahead, we note that our analysis departs from the original Kalai Raz analysis in several aspects (we elaborate on the differences in Section 3.3).

Formalizing the above necessitates putting forward a completely new definition of *rational arguments*. This definition involves dealing with several subtleties and complications, arising from the combination of rational proofs with computational considerations and the presence of cost of computation. We view the formalization of this new concept as one of the main contributions of this paper. Our construction of rational arguments, along with the way it interacts with the definition, highlights the importance of the "gap" between the reward given in case of truthful behavior of the prover and the reward given in case of untruthful behavior. In our construction, the noticeable *reward gap* of our rational proofs for NC1 (in contrast to higher classes within NC) is exactly what allows us to successfully transform them into one-round rational arguments.

In addition to the above, we also establish strong limitations on the reward gap achievable by non-interactive public-coin rational proofs. Specifically, we show that non-trivial languages cannot have such rational proofs with both logarithmic communication complexity and noticeable reward gap. This suggests that the search for rational proofs with large reward gap should focus on private-coin and/or interactive protocols. Progress in this direction would be very interesting, as it might result in succinct non-interactive rational arguments, whose classical counterparts, as shown by Gentry and Wichs [GW11], cannot be achieved under falsifiable assumptions (at least not in a black box way).

To summarize, we view our main contributions to be the following:

- Downscaling rational proofs to efficiently handle a rich class of languages within P (specifically, the class of search problems computable by threshold circuits of depth $d = o(n)$). (Section 4.4)
- Defining a notion of rational arguments that bears meaningful semantics in the context of delegation of computation. (Section 3.1)
- Transforming any interactive rational proof with noticeable reward gap into highly efficient one-round rational arguments. (Section 3.2)
- Establishing limitations on the verification complexity (or more precisely on the complexity of computing the reward) in low communication rational proofs. (Section 4.5)
- Pointing out open questions related to the notions of rational proofs and arguments. (Section 5)

Our rational protocols, while providing a weaker (yet arguably meaningful) guarantee of soundness, compare favorably with each of the known delegation schemes in at least one aspect. They are simple, rely on standard complexity hardness assumptions, provide a correctness guarantee for all instances, and do not require preprocessing.

1.4 Comparison with Known Delegation Schemes

In classical model, the literature contains a variety of models for delegation. For problems within P the protocols can be roughly partitioned into *proofs* and *arguments*, where the latter has been considered both with and without *preprocessing*. For problems outside of P, the main notion considered is that of *succinct non-interactive arguments*. Arguments in preprocessing model and arguments for problems outside P are less relevant to this work. We survey some of them and note that the verification time is not sub-linear.

Before comparing with the classical setting, we present a comparison with other delegation schemes in the rational model. That is, we show a comparison between our rational arguments/rational proofs and rational proofs by Azar and Micali [AM13].

Rational proofs. Independently of our work, [AM13] show a rational proof for uniform constant depth threshold circuits and prove that constant round, super-efficient rational proofs capture TC0. Our rational arguments cover a potentially richer class of problems, while having only one round and only polylogarithmic overhead. We view it as an evidence that rational arguments have the potential to extend super-efficient rational proofs beyond class TC0.

The advantage of our rational proofs over rational proofs in [AM13] is minimality of communication, i.e., only one bit per round. For the restricted class of log-time constant depth threshold circuits, the verification time of our protocol as well as [AM13] is $O(\log n)$. For log-space uniform circuits, a comparison of parameters is in the following table:

Table 1: efficiency comparison of our rational proofs and arguments to [AM13]

	Complexity class	Rounds	Communication	Verification time
[AM13]	TC0	constant	$O(\log n)$	polylog(n)
Our rational proofs	TC0	constant	constant	polylog(n)
Our rational proofs	NC1	$O(\log n)$	$O(\log n)$	polylog(n)
Our rational arguments	NC1	1	polylog(n)	polylog(n)

Proofs. Goldwasser *et al.* [GKR08] give a simple interactive proof for any log-space uniform circuit in NC. Their solution is to traverse the circuit from the top down, run a sum-check protocol and apply standard arithmetizing techniques which take only sub-linear time for each level. This approach yields very good efficiency for all levels except the last one (the input level), where the verifier performs a heavy computation (verifying the evaluation of a polynomial on the whole input) which takes quasi-linear time in the input size.

Rothblum *et al.* [RVW13] study sub-linear verification in the model of interactive proofs of proximity, and gave a protocol based on parallel repetition of the Goldwasser *et al.* [GKR08] protocol. The main difference (and advantage) to Goldwasser *et al.* [GKR08] is that the new model allows to avoid the costly verification procedure for the input level. Efficiency comes at the cost of correctness, which is no longer guaranteed to hold for instances that are only close to being in the language. In addition, communication and verification complexity are increased due to parallel repetition (used for the sake of distance amplification).

Our construction is even simpler than the protocol of Goldwasser *et al.* [GKR08]. Not only our verifier runs in sub-linear time, but also our rational prover runs in linear time in size of the circuit (in previous constructions the prover runs in polynomial time in the circuit size). As in previous constructions, the verifier in our protocol traverses the circuit from the top down, however he merely asks the rational prover for one bit at each level (value of a single gate).[1] Finally, in our protocols the verifier only needs to access one bit of the input (i.e., one query to the input) at the last level. See Table 2 for the full comparison. One other interesting approach is to consider a model with multiple provers. Based on Goldwasser *et al.* [GKR08], Canetti, Riva and Rothblum [CRR13] and Kol and Raz [KR13] show how to obtain more efficient proofs for NC in the

[1] Unlike the previous works, we consider the class TC instead of NC. This is no limitation, since in general NC = TC. Moreover, we gain advantage in the case of constant depth, since $\mathrm{NC}^0 \subset \mathrm{TC}^0$. Note that TC^0 is a powerful model of bounded computation.

Table 2: efficiency comparison of our rational proofs to [GKR08, RVW13]

	Queries	Rounds	Communication	Verification time	Remarks
[GKR08]	n	$\tilde{O}(d)$	$\tilde{O}(d)$	$\tilde{O}(n)$	NC of depth d
[RVW13]	$\left(\frac{n}{D}\right)^{1+o(1)}$	$\tilde{O}(d)$	$D\cdot\left(\frac{n}{D}\right)^{o(1)}\cdot\tilde{O}(d)$	$\left(\frac{n}{D}+D\right)^{1+o(1)}\cdot\tilde{O}(d)$	D-proximity, NC of depth d
this work	1	d	d	$\tilde{O}(d)$	TC of depth d

competing prover model. Kalai, Raz and Rothblum [KRR13] consider delegation for bounded space (which may also contain languages outside NC) in the model of multi-prover interactive proofs. The running time of verifiers in their constructions are at least quasi-linear.

Arguments. Under cryptographic assumptions, it is in general possible to obtain more efficient protocols (though the soundness guarantee then only holds with respect to bounded cheating provers).

In order to minimize interaction, Kalai and Raz [KR09] give a transformation from specific type of interactive protocols into one-round arguments. Goldwasser *et al.* [GKR08] noted that this transformation can be used to achieve non-interactivity also for their delegation scheme. Subsequently, Cormode *et al.* [CMT12] optimized the efficiency of the prover into quasi-linear in circuit size, and Thaler [Tha13] achieved even better efficiency for a natural class of structured circuits. Essentially the same transformation can make the work of Kalai *et al.* [KRR13] one-round. Since the transformation preserves the complexity of the original interactive proof (with only a modest amount of overhead), the running time of the verifier in all the arguments obtained in this way is comparable to the original running time, but it still cannot get better than quasi-linear. In order to get a sub-linear one-round argument via this transformation, one would need to start with a sub-linear delegation scheme as in our case. The construction of Rothblum *et al.* [RVW13] achieves sub-linear verification, however it is not obvious if the same transformation can turn it into a one-round argument. The problem of constructing arguments that are one-round with sub-linear verification time is still open in the classical setting. We show how to simultaneously achieve both in the rational setting.

Preprocessing model and SNARGs. Gennaro, Gentry and Parno [GGP10], Chung, Kalai and Vadhan [CKV10] and Applebaum, Ishai and Kushilevitz [AIK10] use a computationally demanding off-line phase to achieve efficiency during the run of the delegation scheme. The work of Parno, Raykova and Vaikuntanathan [PRV12] further limits the amount of preprocessing needed to aim for *public-verifiability*. Other works consider delegation for NP languages based on non-falsifiable assumptions; the works of Bitansky *et al.* [BCI$^+$13], Damgård, Faust and Hazay [DFH12], Gennaro *et al.* [GGPR12], Groth [Gro10] or Lipmaa [Lip12] give succinct delegation schemes based on such assumptions.

2 Preliminaries and Definitions

Throughout the rest of the paper we use the following notation and definitions. For $n \in \mathbb{N}^+$, let $[n]$ denote the set of first n natural numbers, i.e., $[n] = \{1,\ldots,n\}$. A function $g : \mathbb{N} \to \mathbb{R}$ is negligible if it tends to 0 faster than any inverse polynomial, i.e., for every constant $c \in \mathbb{N}$ there exist constants $k_0 \in \mathbb{N}$ such that for every $k > k_0$ it holds that $g(k) < \frac{1}{n^c}$. We use $\mathrm{negl}(\cdot)$ to talk about negligible function if we do not need to specify its name.

In a rational proof, Arthur pays Merlin a randomized reward according to the transcript of the communication, and the communication constitutes a rational Merlin Arthur game if the correct evaluation $y = f(x)$ can be derived from transcript that maximizes the expected reward.

For a pair of interactive Turing machines, P and V, we denote by $(P,V)(x)$ the random variable representing the transcript between P and V when interacting on common input x. Let $\mathsf{reward}(\cdot)$ denote a randomized function computed by V that given a transcript calculates a reward for P, and by $\mathsf{output}((P,V)(x))$ the output of V after interacting with P on common input x. In this setting, the goal of a *rational* P is to maximize the expected value of $\mathsf{reward}(\cdot)$, while the goal of V is to learn (and output) the true evaluation of the desired function f on x.

We extend the definition of rational Merlin Arthur [AM12] to multiple rounds. Our definition can be seen as a functional analogue of their decisional rational MA definition. Similar to classical proofs, where the statement is fixed ahead of time (i.e., the statement is always proving x in the language), we consider the setting where a rational prover first declares his answer to $f(x)$, and only then tries to prove the correctness of the reported value.

Definition 1 (Functional Rational Merlin Arthur (r rounds)). *Let $C,T : \mathbb{N} \to \mathbb{R}$ be some functions. A function $f : \{0,1\}^* \to \{0,1\}^*$ is in* $\mathrm{FRMA}[r,C,T]$ *if there exists an r-round public-coin protocol (P,V), referred as rational proof, and a randomized reward function* $\mathsf{reward} : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ *such that for all inputs $x \in \{0,1\}^*$:*

(a) $\Pr[\mathsf{output}((P,V)(x)) = f(x)] = 1$

(b) Let $\mathcal{P}_i$ be the view of P up to his message at the ith-round. Then for every round i and for any prover $\tilde{P}$ that behaves as P up to round i and differs on round i holds: $\mathrm{E}[\mathsf{reward}((P,V)(x))] > \mathrm{E}[\mathsf{reward}((\tilde{P},V)(x))]$ *where the expectation is taken over the random coins of the verifier and the prover.*

(c) The communication complexity of P is $C(|x|)$.

(d) The running time of V is $T(|x|)$.

We associate a language L with a function $f : \{0,1\}^* \to \{0,1\}$ where $f(x) = 1$ if and only if $x \in L$. That is, deciding membership in L is equivalent to computing f. The class *Decisional Rational Merlin Arthur*, or DRMA in short, is defined by restricting Definition 1 to binary functions.

3 Rational Arguments

In this section we present the definition of rational arguments, and motivate the choices behind it. We then go on to construct a one-round rational argument, assuming a standard protocol for Private Information Retrieval. Our construction makes use of an interactive rational proof as a building block. The construction of the interactive rational proof is given in Section 4.4. The transformation we employ is similar to a transformation proposed by Kalai and Raz [KR09], with some significant differences in the analysis.

3.1 Definition

Rational arguments are designed to model what happens to the reward when restricting the prover to computationally bounded strategies. One difficulty in capturing the expected prover behavior lies in the fact that he always can attempt to solve the underlying hard problem, and in case of success can reap the highest payoff even while lying about the result. While these attempts will fail with overwhelming probability, one still has to account for those attempts that do succeed. Any definition of computationally bounded strategic behavior should thus take such strategies into consideration, and in particular accommodate negligible gains over the reward guaranteed by the prescribed behavior (but not more). This protects the verifier from giving out too high rewards on expectation, and at the same time ensures that the prover does not lose too much by honestly following the protocol (even though a purely rational prover might not follow the prescribed strategy, since it would try to solve the underlying hard problems).

However, such definition does not take into account the cost of computing $f(x)$. For example, it does not rule out a prover that always gives some default (possibly incorrect) output, without performing any computation, while getting just slightly less than the expectation of the prescribed behavior. Such strategy induces no cost to the prover and is certainly rational if the loss is smaller than the price of computing the function. To address this shortcoming we add another condition, whose objective is to "pin down" the profitability of deviating. Roughly speaking this condition requires that any strategy that lies about the value of $f(x)$ with noticeable probability does noticeably worse (in terms of expected reward) than the prescribed strategy (which always gives out the correct value of $f(x)$). This condition ensures that a rational prover will follow the prescribed strategy in order to increase his utility by a considerable amount. [2]

Definition 2 (Rational Argument). *A function $f : \{0,1\}^* \to \{0,1\}^*$ admits a* rational argument *with security parameter $\kappa : \mathbb{N} \to \mathbb{N}$ if there exists a protocol (P,V) and a randomized reward function* $\mathsf{reward} : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ *such that for any input $x \in \{0,1\}^*$ and any prover $\tilde{P}$ of size $\leq 2^{\kappa(|x|)}$ the following hold:*

(a) $\Pr[\mathsf{output}((P,V)(x)) = f(x)] = 1,$

(b) $\mathrm{E}[\mathsf{reward}((P,V)(x))] + \mathrm{negl}(\kappa(|x|)) \geq \mathrm{E}[\mathsf{reward}((\tilde{P},V)(x))]$*, and*

(c) if there exists a polynomial $p(\cdot)$ such that $\Pr[\mathsf{output}((\tilde{P},V)(x)) \neq f(x)] \geq p(|x|)^{-1}$ *then there exists a polynomial $q(\cdot)$ such that* $\mathrm{E}[\mathsf{reward}((\tilde{P},V)(x))] + q(|x|)^{-1} \leq \mathrm{E}[\mathsf{reward}((P,V)(x))]$

where expectations and the probabilities are taken over the random coins of the respective prover and verifier. We say that the rational argument is efficient *if the running time of V is $o(|x|)$ for every $x \in \{0,1\}^*$.*

The three conditions ensure that the definition is both meaningful and nontrivial. Property (a) corresponds to the notion of *completeness* in classical protocols, and guarantees that a prover following the prescribed protocol will indeed report the correct value of $f(x)$ to the verifier. Property (b) guarantees that the gain attained by deviating from the prescribed strategy in a computationally bounded way is at most negligible. This by itself already guarantees that having both parties follow the protocol is a computational Nash equilibrium. Property (c) guarantees that not reporting $f(x)$ with noticeable probability results in a noticeable utility loss.

We note that Definition 1 of rational proofs (as well as the definition of [AM12]), does not rule out the above discussed "lazy" behavior of the prover once the computation has cost. Having this in mind, we propose to measure how big is the loss of a prover that always reports $f(x)$ incorrectly. A noticeable gap in expectation between such a prover and the prescribed behavior then assures that it is beneficial for the prover to perform the computation to significantly increase its utility. Namely, the verifier could in principle scale the reward by a polynomial factor to make it profitable for the prover to compute the function, while not blowing up too much the budget necessary for paying out the reward. This gives rise to a notion of *reward gap* that allows us to argue for rationality of our proofs in presence of computational cost; it is formalized as follows:

Definition 3 (Reward Gap). *Let $f \in \mathrm{FRMA}[r,C,T]$ be some function and let (P,V) and $\mathsf{reward}(\cdot)$ be the guaranteed protocol and reward function. The* reward gap *of* $\mathsf{reward}(\cdot)$ *is a function $\Delta_{\mathsf{reward}} : \mathbb{N} \to \mathbb{R}$, such that for every $n \in \mathbb{N}$,*

$$\Delta_{\mathsf{reward}}(n) = \min_{x \in \{0,1\}^n} \min_{P^* \in S} \Big(\mathrm{E}[\mathsf{reward}((P,V)(x))] - \mathrm{E}[\mathsf{reward}((P^*,V)(x))] \Big),$$

where the expectation is taken over the random coins of the verifier and the prover, and S is the set of all P^* such that $\Pr[\mathsf{output}((P^*,V)(x)) \neq f(x)] = 1$.

[2] Similar to [AM13], we consider strategies that induce a noticeable utility loss to be irrational.

By simple calculation it follows that the utility loss of an arbitrary prover $\tilde{P}$ is at least $\Delta \cdot \Pr[\mathsf{output}((\tilde{P},V)(x)) \neq f(x)]$. Provers misreporting $f(x)$ with noticeable probability will face a noticeable utility loss, and hence rational proofs with noticeable reward gap offer similar guarantee to property (c) in our definition of rational arguments (Definition 2). This hints on the importance of the reward gap measure for constructing rational arguments.

The work of Azar and Micali [AM13] presented a new definition for rational MA. However, their definition seems to be extremely hard to satisfy due to the constant utility gap requirement. In particular, the constructions presented in their paper seem to fail with respect to their utility gap requirement. For example, consider a prover that is dishonest with only negligible probability. For such a prover, the expected reward is only negligibly far from the reward of the prescribed behavior and hence has only negligible utility gap. That is, any protocol would fail to satisfy that any deviation results in a noticeable utility-loss. Note that this is not an issue for our definition of rational arguments, since it only requires this condition to hold with respect to noticeable deviation. Such a prover also illustrates the choice of the reward gap definition that measures the gap only with respect to provers that always report $f(x)$ incorrectly.

3.2 Building Blocks and Construction

In this section we provide a general transformation that is later applied to rational proofs that we construct in Section 4.4. The main theorem, Theorem 5, establishes that any suitable rational proof (with the required properties described below) can be made one-round, while preserving both its efficiency and the incentive for a rational prover to be truthful. Namely, it guarantees to the verifier that after eliminating interaction, he will be (almost always) provided with a correct answer, although he might need to pay negligibly more than in the interactive version.

We note that the increase in expected reward achieved by the rational prover and his truthfulness directly relate to the user privacy of the underlying PIR scheme.This hints that trying to violate the PIR user privacy is the best strategy to increase one's reward (though this reasoning heavily relies on the assumption that trying to break the PIR incurs no computational cost whatsoever). We use the following definition of PIR scheme.

Definition 4 (Poly-logarithmic PIR [KR09]). *Let κ be the security parameter and N be the database size. Let Q^{PIR} and D^{PIR} be probabilistic circuits, and let R^{PIR} be a deterministic circuit. We say that $(Q^{PIR}, D^{PIR}, R^{PIR})$ is a* poly-logarithmic private information retrieval scheme *if the following conditions are satisfied:*

(a) (Size Restriction:) Q^{PIR} and R^{PIR} are of size $\leq \mathrm{poly}(\kappa, \log N)$, and D^{PIR} is of size $\leq \mathrm{poly}(\kappa, N)$. The output of Q^{PIR} and D^{PIR} is of size $\leq \mathrm{poly}(\kappa, \log N)$.

(b) (Correctness:) $\forall N, \forall \kappa, \forall$database $x = (x_1, \dots, x_N) \in \{0,1\}^N$, and $\forall i \in [N]$,

$$\Pr[R^{PIR}(\kappa, N, i, (q,s), a) = x_i | (q,s) \leftarrow Q^{PIR}(\kappa, N, i), a \leftarrow D^{PIR}(\kappa, x, q)] \geq 1 - 2^{-\kappa^3}.$$

(c) (User Privacy:) $\forall N, \forall \kappa, \forall i, j \in [N]$, and $\forall$ adversary $\mathcal{A}$ of size at most 2^{κ^3},

$$|\Pr[\mathcal{A}(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{PIR}(\kappa, N, i)] - \Pr[\mathcal{A}(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{PIR}(\kappa, N, j)]| \leq 2^{-\kappa^3}.$$

Such PIR schemes exist under different computational assumptions, cf. Cachin, Micali and Stadler [CMS99] who construct poly-logarithmic PIR for any $\kappa > \log N$ under the Φ-Hiding Assumption.

Admissible Protocols. As in Kalai and Raz [KR09] we need the rational proof (P, V) to have the following properties: (1) history-ignorance: each message of the prover P can depend only on the message sent by the verifier at the same round. (2) each message of the verifier V can depend only on its random coins.[3] (3) the running time of the verifier V is sub-linear in the input size. We call such protocols *admissible rational proof*.

We can now present the main theorem that enables us to construct one-round rational arguments from interactive rational proofs (full proof is in Section 3.3). For simplicity of exposition, we normalize the reward to be in $[0,1]$, and make the verifier output 1 with probability of the computed reward (in addition to the reported value of $f(x)$). Note that the probability of a verifier outputting 1 is then exactly the expected reward.

Theorem 5. *Let (P_I, V_I) be an r-round admissible rational proof for evaluating a polynomially bounded function $f : \{0,1\}^n \to \{0,1\}$ with communication complexity ℓ, and let $N = 2^\lambda$, where λ is the length of the longest message sent by V_I. Let $\mathsf{reward}(\cdot)$ and Δ be the reward function and the corresponding reward gap. Assume the existence of a poly-logarithmic PIR scheme with user-privacy $1-\delta$ (where $\delta = 2^{-\kappa^3}$), and let $\delta_0 = (r-1)\cdot\delta$.[4] Then for any security parameter $\kappa = \kappa(n) \geq \max\{\ell, \log n\}$, there exists a one-round protocol (P_A, V_A) with the following properties:*

(a) $\Pr[\mathsf{output}((P_A,V_A)(x)) = f(x)] = 1$

(b) $\mathrm{E}[\mathsf{reward}((P_A,V_A)(x))] \geq \mathrm{E}[\mathsf{reward}((P_I,V_I)(x))]\cdot(1-\delta_0)$.

(c) The communication complexity is $\ell \cdot \mathrm{poly}(\kappa, \log N)$.

(d) The verifier V_A runs in time $\ell\cdot\mathrm{poly}(\kappa) + O(T_{V_I})$, *where T_{V_I} is the running time of V_I.*

(e) The prover P_A runs in time $\mathrm{poly}(\kappa, N, T_{P_I})$, *where T_{P_I} is the running time of P_I.*

(f) For any prover $\tilde{P}$ of size $\leq 2^{\kappa^2}$ that achieves $\mathrm{E}[\mathsf{reward}((\tilde{P},V_A)(x))] = \mathrm{E}[\mathsf{reward}((P_A,V_A)(x))] + \delta^*$, *let* $\mu = \Pr[\mathsf{output}((\tilde{P},V_A)(x)) \neq f(x)]$. *It holds that*

  (i) (Utility gain) $\delta^* \leq 2\delta_0$, and

  (ii) (Utility loss) $(-\delta^*) \geq \mu\Delta - (2+\Delta)\delta_0$.

As pointed out in Section 3.1, the utility loss of an arbitrary prover is at least $\mu\Delta$; hence Theorem 5 presents a construction that basically preserves (up to a negligible factor) the utility gain and loss of the underlying rational proof. Moreover, it guarantees that if the rational proof has noticeable utility loss (in particular, noticeable reward gap) then the constructed protocol will have noticeable utility loss as well.

Theorem 6. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function in* $\mathrm{FRMA}[r,C,T]$. *Assume the existence of a PIR scheme with communication complexity* $\mathrm{poly}(\kappa)$ *and receiver work* $\mathrm{poly}(\kappa)$, *where $\kappa \geq \max\{C(n), \log n\}$ is the security parameter. If f has an admissible rational proof with noticeable reward gap Δ, then f admits rational argument which has the following properties:*

(a) The verifier runs in time $C(n)\cdot\mathrm{poly}(\kappa) + O(T(n))$.

(b) The communication complexity is $r\cdot\mathrm{poly}(\kappa,\log N)$ *with N defined as above.*

Proof. The running time of the verifier, the communication complexity, and property (a) of Definition 2 of rational argumetns are all explicitly provided by Theorem 5. It remains to show property (b) and property (c) of definition of rational arguments.

The utility gain is $\delta^* \leq 2\delta_0 \leq 2^{-\kappa^2+1} \leq 2^{-\log^2 n+1} = \mathrm{negl}(n)$. By the definition of δ^* we have,

$$\mathrm{negl}(n) + \mathrm{E}[\mathsf{reward}((P_A,V_A)(x))] \geq \delta^* + \mathrm{E}[\mathsf{reward}((P_A,V_A)(x))] = \mathrm{E}[\mathsf{reward}((\tilde{P},V_A)(x))],$$

[3] The verifier's messages are independent of the function and the input; however, they can be correlated among themselves.

[4] We assume the existence of poly-logarithmic PIR for any κ and any database size.

and hence property (b) of rational arguments follows.

To show property (c) of Definition 2, we assume that $\mu \geq p^{-1}(|x|)$ for some polynomial $p(\cdot)$. Due to the noticeable Δ, we know that $\mu\Delta \geq q_1^{-1}(|x|)$ for some polynomial $q_1(\cdot)$. From the utility loss bound we obtain that

$$(-\delta^*) \geq \mu\Delta - (2+\Delta)\delta_0 = \mu\Delta - \mathrm{negl}(n) \geq q_1^{-1}(|x|) - \mathrm{negl}(n) \geq q_1^{-1}(|x|)/2.$$

By defining polynomial $q(\cdot)$ to be $q(|x|) = 2q_1(|x|)$ we obtain that

$$\mathrm{E}[\mathsf{reward}((P_A,V_A)(x))] = \mathrm{E}[\mathsf{reward}((\tilde{P},V_A)(x))] - \delta^* \geq \mathrm{E}[\mathsf{reward}((\tilde{P},V_A)(x))] + q^{-1}(|x|)$$

as desired. □

3.3 Proof of Theorem 5

First we present the transformation that allows to squash interaction to a single round, and then show that it achieves the claimed efficiency parameters. The transformation we apply to obtain rational arguments from rational proofs is essentially that of Kalai and Raz [KR09], and it is described for completeness below. Note that even though it uses PIR in order to hide verifier's queries from the prover, it does not introduce the issue of *spooky interactions* discussed in the work of Dwork *et al.*[DLN$^+$04]. Indeed, Kalai and Raz [KR09] proved that the two-message argument obtained by their transformation from a suitable proof system is sound, which does not contradict the results of Dwork *et al.*[DLN$^+$04] stating that in some cases such transformation need not to be secure when applied on an efficient PCP.

From interactive rational proofs to rational arguments. Let (P_I,V_I) be an interactive rational proof for evaluating some function f, as in the statement of the Theorem 5. Recall that λ denotes length of the longest message sent by V_I in (P_I,V_I). For simplicity of exposition (and without loss of generality) we assume that this protocol consists of exactly ℓ rounds, where in the first round P_I sends $f(x)$, and in all other rounds V_I sends a message of size exactly λ, and P_I sends a single bit (i.e., $\ell = r$).

Fix any security parameter $\kappa \geq \max\{\ell, \log n\}$ and let $(Q^{\mathrm{PIR}}, D^{\mathrm{PIR}}, R^{\mathrm{PIR}})$ be a poly-logarithmic PIR scheme, with respect to security parameter κ and database size $N = 2^{\lambda}$. The one-round rational argument (P_A,V_A) is constructed as follows:

1. On common input $x \in \{0,1\}^n$, the verifier V_A proceeds as follows:
   (a) Emulate the verifier V_I and obtain messages $m_2,\ldots,m_\ell \in \{0,1\}^{\lambda}$ to be sent by V_I.[5]
   (b) Compute $(q_i,s_i) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_i)$ for $2 \leq i \leq \ell$ and send $(q_2,\ldots,q_\ell)$ to P_A.
2. Upon receiving a message $(q_2,\ldots,q_\ell)$ from V_A, the prover P_A operates as follows:
   (a) Emulate P_I to obtain $f(x)$.
   (b) For each $2 \leq i \leq \ell$, compute a database DB_i (of size N), which is a collection of all emulated replies of P_I in the i'th round upon receiving any possible message $m \in \{0,1\}^{\lambda}$.[6]
   (c) For each $2 \leq i \leq \ell$, compute $a_i \leftarrow D^{\mathrm{PIR}}(\kappa, DB_i, q_i)$ and send the message $(f(x),a_2,\ldots,a_\ell)$ to V_A.
3. Upon receiving the message $(f(x),a_2,\ldots,a_\ell)$ from P_A, the verifier V_A operates as follows:

[5] These messages can be computed in advance since in the protocol (P_I,V_I) all the messages sent by V_I depend only on V_I's random coin tosses (this follows from (P_I,V_I) being an admissible protocol).

[6] These databases can be computed due to the history-ignorant property of (P_I,V_I).

(a) For every $2 \leq i \leq \ell$, compute $b'_i \leftarrow R^{\mathrm{PIR}}(\kappa, N, m_i, (q_i, s_i), a_i)$.

(b) Emulate V_I on $(f(x), b'_2, \ldots, b'_\ell)$, as if each b'_i is P_I's response in the i'th round.

(c) Output whatever V_I outputs (i.e., $f(x)$ and '1' with probability of the computed reward).

Correctness. Let $x \in \{0,1\}^n$ be a common input. Let $m_2, \ldots, m_\ell \in \{0,1\}^\lambda$ be the messages generated by the underlying V_I. Let $b_2, \ldots, b_\ell$ be the responses to $m_2, \ldots, m_\ell$ of the underlying P_I. Let $b'_2, \ldots, b'_\ell$ be the responses obtained by V_A after applying the algorithm R^{PIR}. According to the correctness property of the poly-logarithmic PIR scheme, for every $2 \leq i \leq \ell$ it holds that $\Pr[b'_i \neq b_i] < 2^{-\kappa^3}$ and by the union bound it follows that

$$\Pr[\text{no error in } R^{\mathrm{PIR}}] = \Pr[(b_2, \ldots, b_\ell) = (b'_2, \ldots, b'_\ell)] \geq 1 - (\ell - 1) \cdot 2^{-\kappa^3} = 1 - \delta_0 \ .$$

Therefore,

$$\begin{aligned} \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] &\geq \mathrm{E}[\mathsf{reward}((P_A, V_A)(x)) | \text{no error in } R^{\mathrm{PIR}}] \cdot \Pr[\text{no error in } R^{\mathrm{PIR}}] \\ &\geq \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] \cdot (1 - \delta_0) \ . \end{aligned}$$

Item (a), which defines the output distribution of the constructed prover, simply follows from the definition of rational proofs and the transformation. More formally,

$$\Pr[\mathsf{output}((P_A, V_A)(x)) = f(x)] = \Pr[\mathsf{output}((P_I, V_I)(x)) = f(x)] = 1.$$

Complexity. The used PIR scheme is poly-logarithmic (see Definition 4), and hence the total communication complexity is

$$|f(x)| + \sum_{i=2}^{\ell} (|a_i| + |q_i|) \leq \ell \cdot \mathrm{poly}(\kappa, \log N) \ .$$

The resulting verifier V_A emulates the verifier of the interactive rational proof V_I and additionally executes the algorithms Q^{PIR} and R^{PIR} $(\ell - 1)$-times. Therefore, its running time can be bounded by

$$\ell \cdot \mathrm{poly}(\kappa, \log N) + O(T_{V_I}) \leq \ell \cdot \mathrm{poly}(\kappa) + O(T_{V_I}) \ .$$

Note that generating a single database requires $N \cdot T_{P_I}$, and each execution of the database algorithm D^{PIR} requires $\mathrm{poly}(\kappa, N)$. Therefore, the running time of P_A can be bounded by

$$\mathrm{poly}(\kappa, N, T_{P_I}) \ .$$

Before completing the proof of Theorem 5, we show how to convert any prover for the resulting (P_A, V_A) into a prover for the original (P_I, V_I). Our transformation preserves (up to a negligible amount) the expected reward and the probability of each answer.

Lemma 7. *Let (P_I, V_I) be an r-round interactive rational proof as in the statement of Theorem 5. Let (P_A, V_A) be the protocol after the transformation above, where we define κ, λ, ℓ, N also as above and assume the existence of poly-logarithmic PIR with user-privacy $1 - \delta$ (where $\delta = 2^{-\kappa^3}$), and let $\delta_0 = (r-1) \cdot \delta$. Given a prover $\tilde{P}_A$ of size at most 2^{κ^2} for (P_A, V_A), there exists a prover $\tilde{P}_I$ for (P_I, V_I) such that*

(a) $|\Pr[\mathsf{output}((\tilde{P}_A, V_A)(x)) = f(x)] - \Pr[\mathsf{output}((\tilde{P}_I, V_I)(x)) = f(x)]| \leq \delta_0$*, and*

(b) $\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_A)(x))] - \mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] \leq \delta_0$.

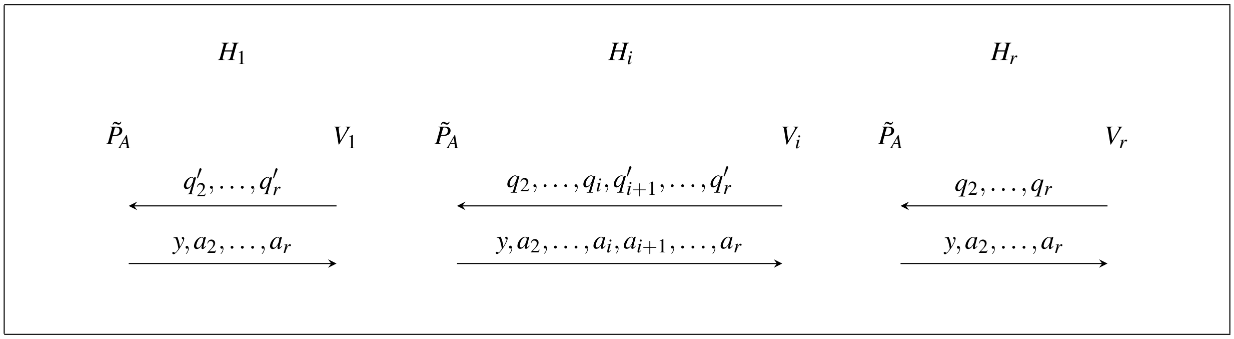

Figure 1: The sequence of hybrids H_i.

Proof. Let $\tilde{P}_A$ of size $T_{\tilde{P}_A} \le 2^{\kappa^2}$ be some prover for (P_A, V_A), and let $x \in \{0,1\}^n$ be some input. Consider $\tilde{P}_I$, a prover for (P_I, V_I), that behaves as follows:

1. Pick $m'_2,\dots,m'_r$ independently at random from $\{0,1\}^\lambda$.

2. Compute $(q'_j, s'_j) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_j)$ for $2 \le j \le r$.

3. Internally run $\tilde{P}_A$ on $(q'_2,\dots,q'_r)$, and obtain the message $(y, a_2,\dots,a_r)$, where y is $\tilde{P}_A$'s answer to $f(x)$.

4. Send y to V_I.

5. Continue in the interaction with V_I. For every $2 \le j \le r$, reply to m_j by a message b'_j that maximizes the expected reward over the randomness V_I conditioned on m_j and the history up to this round.

Assume by a way of contradiction that the statement of lemma does not hold with respect to the $\tilde{P}_I$ above. That is, either

$$|\Pr[\mathsf{output}((\tilde{P}_A, V_A)(x)) = f(x)] - \Pr[\mathsf{output}((\tilde{P}_I, V_I)(x)) = f(x)]| > \delta_0 \text{ , or} \tag{1}$$

$$\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_A)(x))] - \mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] > \delta_0 \text{ .} \tag{2}$$

Consider a sequence of hybrids $H_i = (\tilde{P}_A, V_i)$ (See Figure 1), for $1 \le i \le r$. The i'th hybrid is defined by the following V_i:

1. Emulate V_I to obtain $m_2,\dots,m_r$

2. Choose independently at random $m'_{i+1},\dots,m'_r \in \{0,1\}^\lambda$.

3. Compute $(q_j, s_j) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_j)$ for $2 \le j \le i$ and $(q'_j, s'_j) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_j)$ for $i+1 \le j \le r$.

4. Send $(q_2,\dots,q_i, q'_{i+1}\dots,q'_r)$ to $\tilde{P}_A$.

5. Upon receiving $(y, a_2,\dots,a_r)$ compute $b_j \leftarrow R^{\mathrm{PIR}}(\kappa, N, m_j, (q_j, s_j), a_j)$ for $2 \le j \le i$.

6. Emulate V_I as follows: first give y. Next, for $2 \le j \le i$ reply with b_j on m_j. For $i+1 \le j \le r$, on m_j reply with b'_j that maximizes the expected reward over the randomness of V_I conditioned on m_j and the history up to this round.

7. Output whatever V_I outputs.

By construction of the hybrids it holds that

$$\Pr[\mathsf{output}((\tilde{P}_I, V_I)(x)) = f(x)] = \Pr[\mathsf{output}((\tilde{P}_A, V_1)(x)) = f(x)] \text{ , and}$$
$$\mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] = \mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_1)(x))] \text{ .}$$

Moreover,

$$\Pr[\mathsf{output}((\tilde{P}_A, V_A)(x)) = f(x)] = \Pr[\mathsf{output}((\tilde{P}_A, V_r)(x)) = f(x)] \text{ , and}$$
$$\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_A)(x))] = \mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_r)(x))] \text{ .}$$

If equation (1) does not hold, then by the hybrid argument, there exists $1 \leq k \leq r-1$ such that

$$|\Pr[\mathsf{output}((\tilde{P}_A, V_k)(x)) = f(x)] - \Pr[\mathsf{output}((\tilde{P}_A, V_{k+1})(x)) = f(x)]| > \frac{\delta_0}{r-1} = \delta \text{ .}$$

We show that this contradicts the user privacy of the underlying PIR scheme $(Q^{\mathrm{PIR}}, D^{\mathrm{PIR}}, R^{\mathrm{PIR}})$. That is, we show that there exist an adversary $\mathcal{A}_x^k$ and two strings $m_{k+1}, m'_{k+1} \in \{0,1\}^\lambda$, such that $\mathcal{A}_x^k$ on input query q distinguishes whether q was created from m_{k+1} or m'_{k+1}. Let $\mathcal{A}_x^k$ be as follows:

1. Emulate V_I to obtain messages $m_2, \ldots, m_r$. In addition, choose $m'_{k+1}, \ldots, m'_r \in \{0,1\}^\lambda$ independently at random.
2. Compute $(q_j, s_j) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_j)$ for $2 \leq j \leq k$ and $(q'_j, s'_j) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_j)$ for $k+2 \leq j \leq r$.
3. Upon receiving q, internally run $\tilde{P}_A$ on $(q_2, \ldots, q_k, q, q'_{k+2}, \ldots, q'_r)$, and obtain the message $(y, a_2, \ldots, a_r)$.
4. Compute $f(x)$ and output 1 if $y = f(x)$, otherwise output 0.

Note that

$$\Pr[\mathsf{output}((\tilde{P}_A, V_{k+1})(x)) = f(x)] = \Pr[\mathcal{A}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_{k+1})] \text{ , and}$$
$$\Pr[\mathsf{output}((\tilde{P}_A, V_k)(x)) = f(x)] = \Pr[\mathcal{A}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_{k+1})] \text{ .}$$

Hence,

$$|\Pr[\mathcal{A}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_{k+1})] - \Pr[\mathcal{A}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_{k+1})]| > \delta \text{ .}$$

Moreover, the size of $\mathcal{A}_x^k$ is at most $\mathrm{poly}(2^\kappa) + T_{\tilde{P}_A} \leq 2^{\kappa^3}$ as desired.

Similarly, if equation (2) does not hold, then by the hybrid argument, there exists $1 \leq k \leq r-1$ such that

$$\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_{k+1})(x))] - \mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_k)(x))] > \delta \text{ .}$$

As before, we show that there exist an adversary $\mathcal{B}_x^k$ and two strings $m_{k+1}, m'_{k+1} \in \{0,1\}^\lambda$, such that $\mathcal{B}_x^k$ on input query q distinguishes whether q was created from m_{k+1} or m'_{k+1}. Let $\mathcal{B}_x^k$ be as follows:

1. Perform steps 1. - 3. as in the construction of $\mathcal{A}_x^k$.
2. Emulate V_I as follows: First give y. Next, for $2 \leq j \leq k$ reply with b_j on m_j. For $k+1 \leq j \leq r$, on m_j reply with b'_j that maximizes the expected reward over the randomness of V_I conditioned on m_j and the history up to this round. Note that each b'_j can be computed in time $\mathrm{poly}(2^{\lambda \cdot \ell}) \leq \mathrm{poly}(2^{\kappa^2})$.
3. Output whatever V_I outputs.

Note that

$$\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_{k+1})(x))] \geq \Pr[\mathcal{B}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_{k+1})] \text{ , and}$$
$$\mathrm{E}[\mathsf{reward}((\tilde{P}_A, V_k)(x))] = \Pr[\mathcal{B}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_{k+1})] \text{ .}$$

Hence,

$$\Pr[\mathcal{B}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m_{k+1})] - \Pr[\mathcal{B}_x^k(\kappa, N, q) = 1 | (q,s) \leftarrow Q^{\mathrm{PIR}}(\kappa, N, m'_{k+1})] > \delta \text{ ,}$$

To conclude the proof of the lemma, we note that the size of $\mathcal{B}_x^k$ is at most $\mathrm{poly}(2^{\kappa^2}) + T_{\tilde{P}_A} \leq 2^{\kappa^3}$ as desired.□

Output guarantee. Let $\tilde{P}$ be some prover of size at most 2^{κ^2} for (P_A, V_A) that achieves $\mathrm{E}[\mathsf{reward}((\tilde{P}, V_A)(x))] = \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] + \delta^*$. By Lemma 7, there exists a prover $\tilde{P}_I$ such that:

$$|\Pr[\mathsf{output}((\tilde{P}, V_A)(x)) \neq f(x)] - \Pr[\mathsf{output}((\tilde{P}_I, V_I)(x)) \neq f(x)]| \leq \delta_0 \text{ , and} \tag{3}$$

$$\mathrm{E}[\mathsf{reward}((\tilde{P}, V_A)(x))] - \mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] \leq \delta_0 \text{ .} \tag{4}$$

Let $\mu_0 = \Pr[\mathsf{output}((\tilde{P}_I, V_I)(x)) \neq f(x)]$. By the definition of reward gap,

$$\begin{aligned} &\mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] \\ &\leq \mu_0 \cdot (\mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] - \Delta) + (1 - \mu_0) \cdot \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] \\ &= \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] - \mu_0 \Delta \text{ .} \end{aligned}$$

By combining (4) and the above, we obtain

$$\mathrm{E}[\mathsf{reward}((\tilde{P}, V_A)(x))] \leq \mathrm{E}[\mathsf{reward}((\tilde{P}_I, V_I)(x))] + \delta_0 \leq \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] + \delta_0 - \mu_0 \Delta \text{ .}$$

On the other hand, from the already justified item (b) of Theorem 5 it follows that

$$\mathrm{E}[\mathsf{reward}((\tilde{P}, V_A)(x))] = \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] + \delta^* \geq \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] \cdot (1 - \delta_0) + \delta^* \text{ .}$$

Coupling the two above inequalities and the fact that the reward function assigns values from $[0,1]$ yields

$$\begin{aligned} \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] \cdot (1 - \delta_0) + \delta^* &\leq \mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] + \delta_0 - \mu_0 \Delta \text{ ,} \\ \mu_0 &\leq \frac{\delta_0(\mathrm{E}[\mathsf{reward}((P_I, V_I)(x))] + 1) - \delta^*}{\Delta} \text{ ,} \\ \mu_0 &\leq \frac{2\delta_0 - \delta^*}{\Delta} \text{ .} \end{aligned}$$

Since $\mu_0 \geq 0$, we get an upper bound $\delta^* \leq 2\delta_0$ on δ^*. Finally, combining the upper bound on μ_0 with equation (3) gives

$$\mu = \Pr[\mathsf{output}((\tilde{P}, V_A)(x)) \neq f(x)] \leq \mu_0 + \delta_0 \leq \frac{2\delta_0 - \delta^*}{\Delta} + \delta_0 \text{ ,}$$

which implies $\delta^* \leq 2\delta_0 - (\mu - \delta_0)\Delta = (2 + \Delta)\delta_0 - \mu\Delta$. Thus we obtain $(-\delta^*) \geq \mu\Delta - (2 + \Delta)\delta_0$ which concludes the proof of Theorem 5.

3.4 Main result

We can now state our main result obtained by applying the transformation in Theorem 5 on an efficient interactive rational proofs with noticeable reward gap from Section 4.4. Thus, showing an explicit construction of efficient one-round rational arguments.

Corollary 8 (Main result). *For any function $f : \{0,1\}^n \to \{0,1\}$ computable by log-space uniform* NC1 *of size $S(n) = \mathrm{poly}(n)$ and depth $d = O(\log n)$. Assume the existence of a PIR scheme with communication complexity* $\mathrm{poly}(\kappa)$ *and receiver work* $\mathrm{poly}(\kappa)$*, where $\kappa \geq \max\{d, \log n\}$ is the security parameter. Then f admits* efficient *rational argument which has the following properties:*

(a) The verifier runs in $d \cdot \mathrm{poly}(\kappa)$ and the prover runs in $\mathrm{poly}(\kappa, n)$.

(b) The length of prover's message and the verifier's challenge is $d \cdot \mathrm{poly}(\kappa)$. The verifier's challenge depends only on his random coins and is independent of the input x.

We note that for rational proofs with low complexity parameters, Corollary 8 is not that compelling due to the overhead introduced by the PIR. It is the case especially for constant round rational proofs with up to $O(\log n)$ communication and verification time, i.e., rational protocols for TC0 proposed both in this work or by [AM13].

We already mentioned that the reward gap of rational proofs limits the use of the transformation into rational arguments, however Corollary 8 could be in principle extended to all function in TC with respect to a slightly weaker notion of rational arguments. A rational prover in the corresponding *relaxed model* would need to be sensitive to negligible utility loss, and in a sense it would not be affected by the cost of computing the function. The only semantic difference between the original and relaxed variant of the definition is in the extra guarantee on the utility-loss with respect to deviating provers. That is, property (c) of Definition 2 would change as follows:

$$\left(\mathrm{E}[\mathsf{reward}((\tilde{P},V)(x))] \geq \mathrm{E}[\mathsf{reward}((P,V)(x))]\right) \Rightarrow \left(\Pr[\mathsf{output}((\tilde{P},V)(x)) \neq f(x)] \leq \mathsf{negl}(\kappa)\right).$$

Which demands slightly less, i.e., restricting only the computationally bounded provers achieving more than the prescribed prover to lie with at most negligible probability.

4 Downscaling Rational Proofs

In this section we first illustrate the power of rational proofs for #P (Section 4.1) and then explore the possibility of "scaling down" rational proofs for #P to NP and classes below. In Section 4.2, we give an extremely laconic rational proof where the prover only needs to send one bit for any language in NP, however the verification in this protocol is not sub-linear. In Section 4.3 we show evidence that existence of single round rational proofs with sub-linear verification time is unlikely. In Section 4.4 we consider multiple-round rational proofs, and we show a rational proof for any language computable by uniform threshold circuit of depth d, where the verification time is $\tilde{O}(d)$. In particular, we obtain sub-linear verification time when $d = o(n)$. In Section 4.5 we discuss the possibility to go beyond threshold circuits.

4.1 Rational Proofs for #P

The main tool used in the construction of rational Merlin Arthur games are *scoring rules* that allow to evaluate quality of a forecast about probability of an uncertain future event. Any scoring rule assigns a numerical value $S(\mathcal{Q},x)$ to some predictive distribution $\mathcal{Q}$ and event x drawn from the actual distribution $\mathcal{P}$ that the prediction tries to forecast. The expectation of the score is maximized when the forecaster reports $\mathcal{Q} = \mathcal{P}$.

Definition 9 (Strictly Proper Scoring Rule). *Let $\mathcal{P}$ be a probability distribution over a probability space Ω, and S a scoring rule. We say that S is a* strictly proper scoring rule *with respect to $\mathcal{P}$ if for all $\mathcal{Q} \neq \mathcal{P}$*

$$\sum_{\omega\in\Omega} \mathcal{P}(\omega)S(\mathcal{P},\omega) > \sum_{\omega\in\Omega} \mathcal{P}(\omega)S(\mathcal{Q},\omega),$$

where $\mathcal{P}(\omega)$ is the probability that ω is drawn from $\mathcal{P}$.

The study of scoring rules was initiated by Brier [Bri50], who introduced a first example of a strictly proper scoring rule in the context of meteorological forecasts. A variant of the *Brier's scoring rule* is given by the function $S(\mathcal{P},\omega) = 2\mathcal{P}(\omega) - \sum_{\omega\in\Omega} \mathcal{P}(\omega)^2 - 1$. Another example of strictly proper scoring rule is the *logarithmic scoring rule* $S(\mathcal{P},\omega) = \log(\mathcal{P}(\omega))$. For an extensive survey of scoring rules see Gneiting and Raftery [GR07].

Rational Merlin Arthur games posses a surprising power when compared to classical Merlin Arthur protocols. Azar and Micali [AM12] showed that any problem in #P, the class of counting problems associated with the decision problems in NP, has a rational Merlin Arthur game with a single n bits message from Merlin and Arthur running in time $\mathrm{poly}(n)$. However in the classical setting, such extremely efficient protocol is unlikely to exist.

A notable example of a problem in #P is #SAT i.e., given a Boolean formula ϕ to find the number of satisfying assignments of ϕ. Azar and Micali gave Rational Merlin Arthur game for #SAT defined by the following randomized reward function Brier. for which Arthur wants to know the number of satisfying assignments. He selects a random assignment $z \in \{0,1\}^n$ and sets $b = \phi(z)$. The reward function is then

$$\mathsf{Brier}(\phi, y) = \begin{cases} 2(y/2^n) - (y/2^n)^2 - (1-y/2^n)^2 + 1, & \text{if } b = 1, \\ \\ 2(1-y/2^n) - (y/2^n)^2 - (1-y/2^n)^2 + 1, & \text{if } b = 0. \end{cases}$$

It is obviously a randomized function that can be computed in polynomial time in $|\phi|$ and $|y|$. Let y^* be the number of satisfying assignment of ϕ, by simple calculation,

$$\mathrm{E}[\mathsf{Brier}(\phi, y)] = -2(y/2^n - y^*/2^n)^2 + 2(y^*/2^n)^2 - 2(y^*/2^n) + 2. \tag{5}$$

The fact that $\mathrm{E}[\mathsf{Brier}(\phi, y)]$ is maximized exactly when Merlin sends y^*, the number of satisfying assignments of ϕ, follows because $\mathsf{Brier}(\phi, y)$ is defined as the Brier's scoring rule. Note that any formula ϕ defines a probability distribution of a random variable that is one if a random assignment of ϕ is satisfying and zero otherwise. By sending some y to Arthur, Merlin gives out a prediction about this distribution. Since his reward is computed according to a strictly proper scoring rule, Merlin is incentivized to report the correct number of satisfying assignments.

4.2 Extremely Laconic Rational Prover for any NP Language

In the rational Merlin Arthur game for #P of Azar and Micali, Merlin needs to send n bits to Arthur. We investigate on the possibility of existence of extremely succinct rational proofs. Towards this direction, let us restrict the communication to be only one bit. How can then Merlin convince Arthur about anything? Or else, what information can Arthur obtain in the #SAT protocol?

Recall the expected reward for announcing y for a formula ϕ which has y^* satisfying assignments given by (5). Our main observation is that the expected reward $\mathrm{E}[\mathsf{Brier}(\phi, y)]$ has the following favourable property; the smaller $|y - y^*|$ is, the larger is the expected reward $\mathrm{E}[\mathsf{Brier}(\phi, y)]$. If Merlin can only communicate one bit to Arthur (i.e., $y = 0$ or $y = 1$), he will choose the y which is closer to the true y^*. In particular, rational Merlin will tell Arthur $y = 0$ if $y^* = 0$, and he will announce $y = 1$ in the other case when $y^* \geq 1$. With one bit message from Merlin, Arthur does not expect to learn y^*. Nevertheless, he will obtain one bit information about y^* which is either that $y^* \geq 1$ or $y^* = 0$. And this is sufficient information for Arthur to decide whether $\phi \in \mathrm{SAT}$. It implies SAT has extremely succinct rational proof, and in fact any language in NP has such laconic rational proof where prover sends only one bit to the verifier. This gives the following theorem, that illuminates the power of rational proofs.

Theorem 10. $\mathrm{NP} \subset \mathrm{DRMA}[1, 1, \mathrm{poly}(n)]$.

The general protocol for any NP language is given in Figure 2. Moreover, similar rational proof will work also for languages in coNP and PP. Thus, this two important classes are also contained in $\mathrm{FRMA}[1, 1, \mathrm{poly}(n)]$, and unlike NP, both PP and coNP do not have classical one round interactive proof. For PP, our rational proof improves the efficiency of communication to only one bit message, compared to the rational proof of Azar and Micali with n bits of communication.

We formalize a general observation as following lemma which immediately implies Theorem 10.

On common input x and $L \in \mathrm{NP}$.

1. Merlin will send a bit y to Arthur denoting whether $x \in L$.
2. Arthur will sample a random candidate witness w for x, and compute the randomized reward for Merlin in the following way:

$$\mathsf{Brier}(x,y) = \begin{cases} 2\left(\frac{y}{|W_x|}\right) - \left(\frac{y}{|W_x|}\right)^2 - \left(1 - \frac{y}{|W_x|}\right)^2 + 1, & \text{if } (x,w) \in \mathcal{R}_L, \\ \\ 2\left(1 - \frac{y}{|W_x|}\right) - \left(\frac{y}{|W_x|}\right)^2 - \left(1 - \frac{y}{|W_x|}\right)^2 + 1, & \text{if } (x,w) \notin \mathcal{R}_L. \end{cases}$$

Figure 2: Rational Merlin Arthur game for any NP language L.

Lemma 11 (Threshold Lemma). *For any polynomially bounded function $f : \{0,1\}^* \to \{0,1\}$. Let $C : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ be a randomized procedure, and let $\beta_0, \beta_1 : \mathbb{N} \to [0,1]$ be two functions such that $\beta_0(n) < \beta_1(n)$ for every $n \in \mathbb{N}$. If for any $x \in \{0,1\}^*$ it holds that*

(a) $\Pr_r[C(x;r) = 1 | f(x) = 1] \geq \beta_1(|x|)$*, and*

(b) $\Pr_r[C(x;r) = 1 | f(x) = 0] \leq \beta_0(|x|)$*,*

then $f \in \mathrm{FRMA}[1,1,t(\cdot)]$, where $t(\cdot)$ is the running time of C.

Proof. Let $x \in \{0,1\}^*$ be a common input of length n on which Arthur wants to evaluate f. We show that the following protocol constitutes a rational Merlin Arthur game for f:

1. Merlin sends $y \in \{0,1\}$ to Arthur.
2. Arthur uniformly at random selects the random coins r for C and sets $b = C(x;r)$. He then pays Merlin $\mathsf{Brier}(x,y)$, where

$$\mathsf{Brier}(x,y) = \begin{cases} 2\beta_y(n) - (\beta_y(n))^2 - (1 - \beta_y(n))^2 + 1, & \text{if } b = 1, \\ \\ 2(1 - \beta_y(n)) - (\beta_y(n))^2 - (1 - \beta_y(n))^2 + 1, & \text{if } b = 0. \end{cases}$$

Clearly, the running time of Arthur is $t(n)$.

Let $\beta = \Pr_r[C(x;r) = 1]$ (note that β would differ for another input $x' \in \{0,1\}^*$). The expected reward for Merlin when sending y is $\mathrm{E}_r[\mathsf{Brier}(x,y)] = -2(\beta_y(n) - \beta)^2 + 2\beta^2 - 2\beta + 2$. Intuitively, β corresponds to the true distribution that Arthur wants to learn. However, Merlin is restricted to report either $\beta_0(n)$ or $\beta_1(n)$ by sending $y \in \{0,1\}$ to Arthur in the above protocol. In order to maximize his reward, Merlin must choose y such that $\beta_y(n)$ is the closer one to β.

Formally, if $f(x) = 1$ then $\beta \geq \beta_1(n) > \beta_0(n)$, thus $\mathrm{E}_r[\mathsf{Brier}(x,1)] - \mathrm{E}_r[\mathsf{Brier}(x,0)] = 2((\beta_0(n) - \beta)^2 - (\beta_1(n) - \beta)^2) > 0$. Moreover, if $f(x) = 0$ then $\beta \leq \beta_0(n) < \beta_1(n)$, thus $\mathrm{E}_r[\mathsf{Brier}(x,0)] - \mathrm{E}_r[\mathsf{Brier}(x,1)] = 2((\beta_1(n) - \beta)^2 - (\beta_0(n) - \beta)^2) > 0$. In both cases, rational Merlin has a clear incentive to truthfully send $y = f(x)$. □

Notice that the proof of Lemma 11 makes mainly use of the fact that the expected reward is a distance function; the closer y is to the true answer, the more reward Merlin can get. Hence, we can replace the Brier's scoring rule with other scoring rules which have this property. For distributions over 0 and 1, most strictly proper scoring rules have this property.

Proof of Theorem 10. For any $x \in L$, let W_x be the space of all possible witnesses for membership of x in the language L. We define the following randomized procedure C_L. Select a candidate witness w uniformly at random from W_x. Set $C_L(x;w) = 1$ if w is a witness for $x \in L$, and $C_L(x;w) = 0$ otherwise. Since L is an NP language, $C_L(x;w)$ can be computed in polynomial time in $|x|$. Moreover, $\Pr_{w \in W_x}[C_L(x;w) = 1] \geq \frac{1}{|W_x|}$ for every $x \in L$ and $\Pr_{w \in W_x}[C_L(x;w) = 1] = 0$ for any $x \notin L$. The theorem follows from Lemma 11. □

4.3 Impossibility for Parity

A natural question to ask is if one can improve the verification time in our extremely laconic rational proofs for NP to be sub-linear. Our first observation is that, even for such a simple function as parity, it is impossible to have an extremely laconic rational proof with sub-linear verification. However, the verifier can compute parity in linear time on his own. Consequently, any rational proof for parity with such extremely laconic Merlin is not interesting from the perspective of delegation of computation.

Theorem 12. $\bigoplus x = x_1 \oplus \cdots \oplus x_n \notin \mathrm{FRMA}[1,1,o(n)]$.

Proof. Suppose $\bigoplus x \in \mathrm{FRMA}[1,1,o(n)]$. Then there exists a randomized reward function $\mathsf{reward}(x,y;r)$ (here we explicitly use an argument r to denote the randomness of the reward function) of running time $o(n)$, such that for any $x \in \{0,1\}^n$, Merlin will get higher expected reward by sending $y = \bigoplus x$ than by sending $y' = \overline{\bigoplus x}$. Formally, for any $x \in \{0,1\}^n$

$$\mathrm{E}_r\left[\mathsf{reward}(x, \textstyle\bigoplus x; r)\right] > \mathrm{E}_r[\mathsf{reward}(x, \overline{\textstyle\bigoplus x}; r)].$$

It implies that for all inputs, the average advantage of telling the truth to lying is greater than zero, i.e.,

$$\mathrm{E}_{x \sim \{0,1\}^n}\left[\mathrm{E}_r[\mathsf{reward}(x, \textstyle\bigoplus x; r)] - \mathrm{E}_r[\mathsf{reward}(x, \overline{\textstyle\bigoplus x}; r)]\right] > 0.$$

By linearity of expectation and after changing the order of variables, we obtain the following inequality

$$\mathrm{E}_r\left[\mathrm{E}_{x \sim \{0,1\}^n}[\mathsf{reward}(x, \textstyle\bigoplus x; r) - \mathsf{reward}(x, \overline{\textstyle\bigoplus x}; r)]\right] > 0.$$

To demonstrate the sought contradiction, it is sufficient to show that for any fixed r

$$\mathrm{E}_{x \sim \{0,1\}^n}\left[\mathsf{reward}(x, \textstyle\bigoplus x; r) - \mathsf{reward}(x, \overline{\textstyle\bigoplus x}; r)\right] = 0. \tag{6}$$

Recall that for any fixed r, the reward function $\mathsf{reward}(\cdot,\cdot;r)$ runs in time $o(n)$. Thus, $\mathsf{reward}(\cdot,\cdot;r)$ depends on at most $n-1$ bits of the input. Without loss of generality, we assume that $\mathsf{reward}(\cdot,\cdot;r)$ does not depend on x_n. In such case, for any choice of $x_1,\ldots,x_{n-1},y \in \{0,1\}$,

$$\mathsf{reward}(x_1,\ldots,x_{n-1},0,y;r) = \mathsf{reward}(x_1,\ldots,x_{n-1},1,y;r).$$

Therefore, for any choice of the bits $x_1,\ldots,x_{n-1}$, we have

$$\begin{aligned}
&\mathrm{E}_{x_n \sim \{0,1\}}\left[\mathsf{reward}(x, \textstyle\bigoplus x; r) - \mathsf{reward}(x, \overline{\textstyle\bigoplus x}; r)\right] \\
&= 1/2\Big(\big(\mathsf{reward}(x, b \oplus 0; r) - \mathsf{reward}(x, b \oplus 1; r)\big)|_{x_n=0} + \big(\mathsf{reward}(x, b \oplus 1; r) - \mathsf{reward}(x, b \oplus 0; r)\big)|_{x_n=1}\Big) \\
&= 1/2\Big(\big(\mathsf{reward}(x, b \oplus 0; r) - \mathsf{reward}(x, b \oplus 1; r)\big)|_{x_n=0} + \big(\mathsf{reward}(x, b \oplus 1; r) - \mathsf{reward}(x, b \oplus 0; r)\big)|_{x_n=0}\Big) \\
&= 1/2\big(\mathsf{reward}(x, b \oplus 0; r) - \mathsf{reward}(x, b \oplus 1; r) + \mathsf{reward}(x, b \oplus 1; r) - \mathsf{reward}(x, b \oplus 0; r)\big)|_{x_n=0} \\
&= 0,
\end{aligned}$$

where $b = x_1 \oplus \cdots \oplus x_{n-1}$. In fact, the above already shows that equation (6) holds, since for every fixed r,

$$\begin{aligned}
&\mathrm{E}_{x\sim\{0,1\}^n}[\mathsf{reward}(x,\bigoplus x;r) - \mathsf{reward}(x,\overline{\bigoplus x};r)] \\
&= \mathrm{E}_{x_1,\ldots,x_{n-1}\sim\{0,1\}}\,\mathrm{E}_{x_n\sim\{0,1\}}[\mathsf{reward}(x,\bigoplus x;r) - \mathsf{reward}(x,\overline{\bigoplus x};r)] \\
&= \mathrm{E}_{x_1,\ldots,x_{n-1}\sim\{0,1\}}\,[0] \\
&= 0.
\end{aligned}$$

□

This result demonstrates the difficulty of getting sub-linear verification with extremely laconic rational proof (even for a very restricted problem in P). It hints that in order to circumvent this issue, we have to increase the communication or the number of rounds.

4.4 Efficient Rational Proofs for TC

By using multiple rounds, we show a general solution for functions computable by log-space uniform threshold circuits of depth d with d rounds, $d-1$ bit communication overhead and $d\cdot\mathrm{polylog}(n)$ verification time. In particular, for the case of $d = o(n)$, both communication and the verification time are sub-linear. Another implication of this result is that we can give a rational proof with constant rounds, constant communication bits and polylog(n) verification for parity since parity function is computable by constant depth threshold circuit.

Theorem 13. *If $f\colon \{0,1\}^* \to \{0,1\}$ is computable by a family of $O(\log(S(n)))$-space uniform threshold circuits of size $S(n)$ and depth $d(n)$, then $f \in \mathrm{FRMA}[d(n), d(n), O(d(n)\cdot\mathrm{polylog}(S(n))]$.*

We provide the efficiency comparison with known delegation schemes in Section 1.4. Additional discussion regarding threshold circuits, non-uniform model of computation and multi-output case are provided in Section 4.4.2. Description of the protocol is given below.

Protocol. Let a circuit of depth d and $x \in \{0,1\}^n$ be the common input to Arthur and Merlin. The protocol begins with the following conversation,

1. Merlin: Evaluate the circuit on x and send y_1, the output value, to Arthur.
2. Arthur: Set $\gamma = 1/(1+2m_0^2)$, where m_0 is the largest fan-in over all gates in the circuit.[7] Identify the 'root' gate g_1, and invoke Round$(1, g_1, y_1)$,

where the procedure Round(i, g_i, y_i) is for $1 \le i \le d$ defined as

1. Arthur: Choose uniformly at random a "child" g_{i+1} of g_i.
   - If g_{i+1} is an input wire of value z then pay Merlin $\gamma^{d-i}/2\cdot\mathsf{Brier}(g_i, y_i)$ where $b = z$ and **STOP**.
   - Otherwise, ask M_{i+1} for the value of g_{i+1}.
2. Merlin: Send y_{i+1}, the value of g_{i+1}, to Arthur.
3. Arthur: Pay Merlin $\gamma^{d-i}/2\cdot\mathsf{Brier}(g_i, y_i)$ where $b = y_{i+1}$ and go to Round$(i+1, g_{i+1}, y_{i+1})$.

Notice that Brier is the reward function defined for single gate case (see (7)) and both Brier and the discounting factor γ are efficiently computable in $O(\mathrm{polylog}(m_0)) = O(\mathrm{polylog}(S(n)))$ time.

[7] In fact m_0 can be any number larger than the maximal fan-in (for example siz of the circuit). So that to settle m_0 we don't need exactly compute the maximal fan-in.

Efficiency. The protocol runs at most d rounds. In each round, Merlin sends one bit to Arthur, thus the communication complexity is at most d bits. The computation of Arthur is sampling a child and calculating the reward function in each round. By the uniformity of the circuit, any information about each gate (e.g. the type of the gate, identifying the children) can be computed in space $O(\log(S(n)))$ and time $O(\mathrm{polylog}(S(n)))$. In particular, sampling a child only takes $O(\mathrm{polylog}(S(n)))$ time. The reward function is computable in $\mathrm{polylog}(n)$ time. Hence, Arthur runs in $O(d \cdot \mathrm{polylog}(S(n)))$. Remark that the computation of Arthur can be done in space $O(\log(S(n)))$ since Arthur only needs to keep the information i, g_i, y_i for each $\mathrm{Round}(i, g_i, y_i)$ and compute the reward in space $O(\log(S(n)))$.

Reward Gap. Recall the Definition 3. of reward gap. The above protocol achieves reward gap proportional to the depth of the circuit, in particular

$$\Delta(|x|) > \frac{\gamma^{d-1}}{2} \cdot \frac{1}{m_0^2} \geq \frac{1}{(2m_0^2+1)^{d-1}} \cdot \frac{1}{2m_0^2} \geq \frac{1}{(2m_0^2+1)^d},$$

where m_0 is the maximal fan-in over all gates in the circuit. This gives noticeable reward gap for TC0, where $m_0 = O(\mathrm{poly}(|x|))$ and d is constant and similarly for NC1, where m_0 is constant and $d = O(\log|x|)$.

4.4.1 Proof of Theorem 13

First, we apply our Threshold lemma (Lemma 11) to obtain extremely laconic rational proof for any single threshold gate which is the main building block.

Lemma 14. *If $g(x)$ is a threshold gate with fan-in n and threshold a, i.e.,*

$$g(x) = \begin{cases} 1, & \text{if } x_1 + \cdots + x_n \geq a, \\ \\ 0, & \text{if } x_1 + \cdots + x_n \leq a-1, \end{cases}$$

then $g(x) \in \mathrm{FRMA}[1,1,O(\log n)]$.

Proof. Let $C(x;r)$ be: on input $x = (x_1, \ldots, x_n)$, uniformly at random choose $r \in \{1, \ldots, n\}$, and output x_r. Notice that the running time of $C(x;r)$ is $O(\log n)$. Since g is a gate with threshold a, we get

(a) $\Pr_r[C(x;r) = 1 | g(x) = 1] \geq \beta_1(n)$, and

(b) $\Pr_r[C(x;r) = 1 | g(x) = 0] \leq \beta_0(n)$,

where $\beta_1(n) = a/n, \beta_0(n) = (a-1)/n$. By Lemma 11, $C(x;r)$ allows us to construct a rational proof. In particular, if Merlin sends $y \in \{0,1\}$ to Arthur and the output of $C(x;r)$ is b, Arthur will pay Merlin $\mathsf{Brier}(g,y)$, where

$$\mathsf{Brier}(g,y) = \begin{cases} 2\beta_y(g) - (\beta_y(g))^2 - (1-\beta_y(g))^2 + 1, & \text{if } b = 1, \\ \\ 2(1-\beta_y(g)) - (\beta_y(g))^2 - (1-\beta_y(g))^2 + 1, & \text{if } b = 0. \end{cases} \tag{7}$$

It follows that $g(x) \in \mathrm{FRMA}[1,1,O(\log n)]$. □

MAJORITY, AND, and OR are all threshold gates, hence this lemma implies MAJORITY, AND, and OR are in $\mathrm{FRMA}[1,1,O(\log n)]$.[8] Let $d-\mathrm{FRMA}$ denote the class of languages that admit rational proof with d

[8] Remark that the NOT gate can be thought of as a threshold gate and has a similar rational proof, where one counts the number of zeros in the input instead of the number of ones. We therefore handle MAJORITY, AND, OR, and NOT gates.

independent Merlins. We show how to delegate threshold circuits using multiple independent Merlins, then replace independent Merlins by one Merlin.

Lemma 15. *If $f\colon \{0,1\}^* \to \{0,1\}$ is computable by a family of $O\big(\log(S(n))\big)$-space uniform threshold circuits of size $S(n)$ and depth $d(n)$, then $f \in d(n) - \mathrm{FRMA}[d(n), d(n), O\big(d(n) \cdot \mathrm{polylog}(S(n))\big)]$.*

Proof. To give an idea of our proof, let us make the simplifying assumption that for x of length n, f is computable by a tree of depth $d = d(n)$. We denote M_i the ith Merlin and $y_i \in \{0,1\}$ the bit sent by M_i to Arthur in our rational proof.

The protocol constitutes of d rounds. In the first round, M_1 will tell Arthur the value y_1 of the root gate g_1. In Lemma 14, Arthur incetivizes Merlin to tell the truth for a single gate g by randomly picking an input bit b and paying $\mathsf{Brier}(g,y)$ defined as in equation (7) to Merlin. Similarly, Arthur will randomly pick a subtree and pay $\mathsf{Brier}(g_1,y_1)$ to M_1 where b is the value of the chosen subtree. Arthur can avoid computing b by paying another independent Merlin M_2 to get the bit b. Conditioned on M_2 telling the correct value of b, M_1 only maximizes his reward when telling the truth.

Thus, we reduce incentivizing M_1 to tell the truth for the root of a depth d tree to making M_2 tell the truth for any depth $d-1$ subtree. By repeating this idea, we reduce the whole problem to making M_d tell the truth for any tree of depth 1 which is just a threshold gate, and this case is guaranteed by Lemma 14. The complete protocol is as follows.

Protocol with d Merlins. Let $x \in \{0,1\}^n$ and some depth d circuit be the common input to Arthur and $M_1, \ldots M_d$. The protocol begins with following conversation,

1. Merlin M_1: sends y_1 to Arthur.

2. Arthur: identifies the 'root' gate g_1 and invokes $\mathrm{Round}(1, g_1, y_1)$,

where the procedure $\mathrm{Round}(i, g_i, y_i)$ is for $1 \le i \le d$ defined as

1. Arthur: randomly and uniformly chooses a 'child' g_{i+1} of g_i.

   - If g_{i+1} is an input wire of value z, then pay $\mathsf{reward}(g_i, y_i)$ to M_i where $b = z$ and **STOP**.
   - Otherwise, ask M_{i+1} for the value of g_{i+1}.

2. Merlin M_{i+1}: sends y_{i+1}.

3. Arthur: Pay $\mathsf{Brier}(g_i, y_i)$ to M_i where $b = y_{i+1}$ and go to $\mathrm{Round}(i+1, g_{i+1}, y_{i+1})$. □

The Figure 3. depicts the process of selecting a random path in the circuit by Arthur from our protocol. Next, we prove the correctness. Note the achieved efficiency parameters are discussed in Section 4.4.

Correctness. We show by backward induction that for any $1 \le i \le d$, M_i prefers to answer Arthur with a correct value on the root for any depth $(d-i+1)$ threshold circuit. The base case is to show that M_d will tell the truth for any circuit of depth one, i.e., a threshold gate, which is guaranteed by Lemma 14. Assume that $M_d, \ldots, M_{i+1}$ are being truthful; we show that M_i will be consequently also truthful for any depth $d-i+1$ circuit X. Let the root gate g of X be connected to some depth (at most) $d-i$ sub-circuits $X_1, \ldots, X_m$. By the inductive hypothesis, $M_d, \ldots, M_{i+1}$ enable Arthur to learn the root value for each one of $X_1, \ldots, X_m$, and we can think of $X_1, \ldots, X_m$ as of fixed input bits each having depth zero, effectively reducing the depth of X to one. Hence, due to the help of $M_d, \ldots, M_{i+1}$, we only need to show that M_i will tell the truth for any threshold gate g. This case is the same as basic case ensured by Lemma 14. Hence, we can conclude that all of $M_1, \ldots, M_d$ prefer to truthfully report the value of the root for their respective sub-circuits.

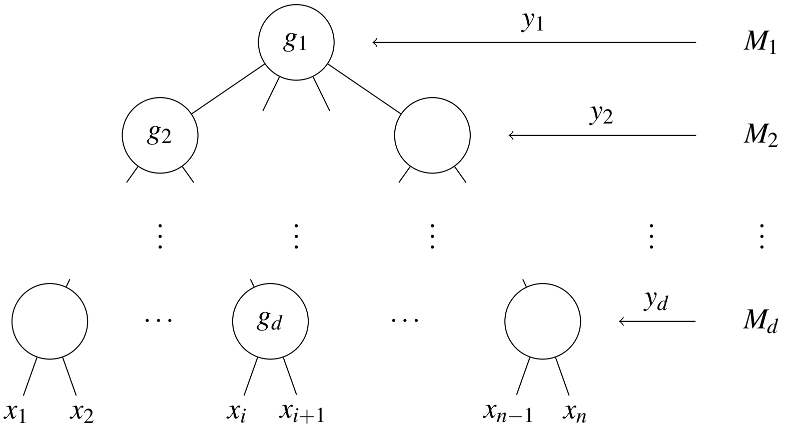

Figure 3: An independent Merlin M_i provides the intermediate value for each level i in the circuit according to Arthur's query g_i.

Arthur can in fact achieve the same when communicating with one Merlin. To merge the d independent Merlins into a protocol with a single Merlin, we discount the reward on the consecutive levels from the input gates to the root.[9]

Proof of Theorem 13. Besides the fact that Arthur is communicating with a single Merlin, the only difference between the new protocol and the protocol described in the proof of Lemma 15 is that instead of paying $\mathsf{Brier}(g_i, y_i)$ in $\mathrm{Round}(i, g_i, y_i)$, Arthur now computes the reward as $\gamma^{d-i}/2 \cdot \mathsf{Brier}(g_i, y_i)$, where $\gamma = 1/(1+2m_0^2)$ and m_0 is the largest fan-in over all gates. After the protocol finishes, Arthur pays Merlin the total sum over all rounds.

The communication complexity and the number of rounds of the new protocol is exactly the same as of the protocol in Lemma 15. To check that it also constitutes a rational proof, we use backward induction to show that Merlin is truthful in each round when maximizing the expected reward. The base case is to prove that Merlin will tell the truth in $\mathrm{Round}(d, g_d, y_d)$. If Merlin lies in this round his expected loss is

$$\begin{aligned}\gamma^{d-d}/2 \cdot |\,\mathrm{E}_r[\mathsf{Brier}(g_d,0)] - \mathrm{E}_r[\mathsf{Brier}(g_d,1)]| &= \gamma^{d-d}/2 \cdot |(\beta - \beta_0(m))^2 - (\beta - \beta_1(m))^2| \\ &\geq \gamma^{d-d}/2 \cdot |(\beta_0(m) - \beta_1(m))(2\beta - \beta_0(m) - \beta_1(m))| \\ &\geq \gamma^{d-d}/2 \cdot (\beta_0(m) - \beta_1(m))^2 = \gamma^{d-d}/2 \cdot \frac{1}{m^2} \geq \frac{1}{2m_0^2},\end{aligned}$$

where m is the fan-in of g_d. Since for every $1 \leq j \leq d$ we have $\mathsf{Brier}(g_j, y_j) \in [0,2]$, the maximal total reward over the previous rounds $1, \ldots, d-1$ can be bound, i.e.,

$$\sum_{j=1}^{d-1} \gamma^{d-j}/2 \cdot \mathsf{Brier}(g_j, y_j) \leq \sum_{j=1}^{d-1} \gamma^{d-j} = \sum_{j=1}^{d-1} \left(\frac{1}{1+2m_0^2}\right)^{d-j} < \frac{1}{2m_0^2}.$$

Note that the total reward conditioned on lying in the last round is strictly smaller than the reward in the last round alone when telling the truth, hence a rational Merlin must be truthful in this round in order to maximize his total reward.

Assume that Merlin is truthful in rounds $d, \ldots, i+1$. Similarly to the base case, if he lies in $\mathrm{Round}(i, g_i, y_i)$, then his expected loss is at least $\gamma^{d-i}/2 \cdot 1/m_0^2$ (conditioned on telling the truth in the next rounds). Moreover, the maximal total income over rounds $1, \ldots, i-1$ is also bounded, i.e.,

$$\sum_{j=1}^{i-1} \gamma^{d-j}/2 \cdot \mathsf{Brier}(g_j, y_j) \leq \sum_{j=1}^{i-1} \gamma^{d-j} < \gamma^{d-i} \cdot (1/2m_0^2).$$

[9] This idea resembles the proof of CH ⊂ DRMA presented in [AM12], however the discounting differs in our context.

By the same reasoning, Merlin is reporting the truth in Round (i,g_i,y_i) in order to maximize the total reward. Therefore, we can conclude that rational Merlin will report the truth in every round, and the protocol indeed constitutes a rational proof. □

4.4.2 Discussion on Rational Proofs for Threshold Circuits

Circuit Model. Our rational proofs work for any circuit in TC, whereas interactive proofs of Goldwasser *et al.*[GKR08] address circuits in NC. In general, the class TC and NC have the same computational power, indeed it is possible to simulate any threshold gate with AND and OR gates. However, once we take depth of the circuit into consideration, TC circuits may be much more powerful than NC circuits, and the advantage is remarkably evident in the case of constant depth.

Unlike TC0, the class NC0 can only compute functions which depend on constant number of input bits, since the fan-in of any gate is two. One can lift this restriction and allow the AND, and OR gates to have unbounded fan-in which corresponds to the class AC0. Nevertheless, we do know functions which are computable in TC0, but are outside AC0, e.g., parity, majority and some pseudorandom functions (under a computational assumption, see Naor and Reingold [NR04]). Note that TC0 is a class interesting also on its own (see Reif and Tate [RT92]). Many important practical problems are in TC0, such as sorting a list of n-bit numbers, multiplication of two n bit numbers, or integer division. Our result gives efficient delegation schemes for these problems in the rational setting with constant number of rounds, constant communication, and with a verifier running in $\mathrm{polylog}(n)$ time.

In the construction of both us and Goldwasser *et al.*[GKR08], the number of rounds and the amount of communication is proportional to the depth of the circuit. However, for the same depth d, we need only d rounds but Goldwasser *et al.*[GKR08] needs $\Omega(d\log n)$ rounds. Hence, for those function which are computable in TC0, our protocol only needs $O(1)$ rounds $O(1)$ bits communication that is in stark contrast to $\Omega(\log n)$ rounds and $\mathrm{polylog}(n)$ communication in [GKR08].

Multi-Output Case. Both our protocol and Goldwasser *et al.* [GKR08] can be transformed to protocols that work for multi-output circuits of say m output bits. The general idea is to view these output bits as inputs to a virtual AND gate. This AND gate then outputs 1 if and only if all output bits are correct, and turns the circuit into a single-output circuit. The AND gate is however limited to fan-in two in NC, so the transformation adds $\log m$ factor to the depth of the circuit, and consequently to the number of rounds. Our transformation for multi-output circuits introduces only one additional round in which the verifier picks an output bit at random and runs the rational verification for the corresponding single-output circuit. More specifically, we show any $f\colon \{0,1\}^* \to \{0,1\}^\ell$ which can be computable by a family of $O(\log(S(n)))$-space uniform threshold circuits of size $S(n)$ and depth $d(n)$ is in $\mathrm{FRMA}[d(n), d(n)+\ell-1, O(d(n)\cdot \mathrm{polylog}(S(n)))]$. The protocol only differs from the protocol of Theorem 13 in the initial conversation.

(a) Merlin: sends $z \in \{0,1\}^\ell$.

(b) Arthur: randomly chooses $r \in \{1,\ldots,\ell\}$. Finds the corresponding ”root” gate, denotes it as g_1, and starts $\mathrm{Round}(1, g_1, z_r)$.

Suppose Merlin sends an incorrect $z \neq f(x)$, and a is the number of bits where z and $f(x)$ differ. With probability of a/ℓ, Arthur will start the first round with an incorrect z_r. As shown in Theorem 13, conditioned on z_r being incorrect, the expected loss of Merlin is at least $\Delta^d/2\cdot 1/m_0^2$, where m_0 is the maximal fan-in over all gates. Thus, the expected loss of Merlin when lying on a bits of z will be at least $(a/\ell)(\Delta^d/2\cdot 1/m_0^2)$. Merlin will again maximize the expected reward by sending $z = f(x)$.

Non-Uniform Circuits. Uniformity of circuits describe how complex the circuits are. The statements of muggles and our protocol work demand $O(\log(S(n)))$ space uniform circuits where $S(n)$ is the size of circuit. It is because we want to get the information of circuit very efficiently without looking into the whole description of circuit. $O(\log S(n))$ space uniformity allows us to get the information of each gate in $\text{polylog}(S(n))$ time. As discussed in Goldwasser *et al.* [GKR08], if preprocessing allowed, verifier can run $\text{poly}(S(n))$ time in off-line phase but keep $\text{poly}(d, \log(S))$ bits of information of the circuit. And in the online phase, verifier can get the information of each gate in time $\text{poly}(d, \log(S(n)))$ such that the total running time is $O(d \cdot \text{poly}(d, \log(S(n))))$. The running time of Goldwasser *et al.* [GKR08] is $O(n \cdot \text{poly}(d, \log(S(n))))$.

4.5 Beyond TC

We show some limitation on extremely efficient rational proof for classes beyond TC. In the rational proof for any NP language described in Fig. 2, the assumption about rationality of Merlin allows him to convince Arthur while being extremely laconic. However, Goldreich *et al.* [GVW02] showed that in the classical setting it is impossible to construct an interactive proof with a single, one bit message from the verifier for any NP-complete language.[10] We show that similar impossibility transfers also into the rational setting.

As mentioned earlier, Brier's score is not inherent to laconic rational proofs. However, for any bounded scoring rule computable in polynomial time, we show in following theorem that the reward gap is related to the randomized time needed for deciding the language, and we get as simple corollary that the reward gap cannot be noticeable in every extremely laconic rational Merlin Arthur game for any NP language, unless NP $\subset$ BPP.

Theorem 16. *Let L be a language in* $\text{DRMA}[1, 1, \text{poly}(|x|)]$*, and* $\mathsf{reward}(\cdot,\cdot)$ *be any bounded reward function defining the corresponding rational Merlin-Arthur game with an extremely laconic Merlin. If $\Delta_{\mathsf{reward}}(\cdot)$ is a noticeable function, then L can be decided in randomized polynomial time with a constant error probability.*

Proof. Given $x \in \{0,1\}^*$, consider the following procedure. Evaluate t times both $\mathsf{reward}(x,0)$ and $\mathsf{reward}(x,1)$ to estimate $\mathrm{E}[\mathsf{reward}(x,0)]$ respectively $\mathrm{E}[\mathsf{reward}(x,1)]$. If $\mathrm{E}[\mathsf{reward}(x,0)] > \mathrm{E}[\mathsf{reward}(x,1)]$ reject, and otherwise accept.

We use the Hoeffding's inequality to give an upper bound on the number t of evaluations of the reward function that suffices for the above procedure to decide L with a constant probability of error. The inequality states that if $X_1, \ldots, X_t$ are i.i.d. random variables such that $\mathrm{E}[X_i] = \mu$ and $0 \le X_i \le 1$ for all i, then for any $\varepsilon > 0$,

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - \mu\right| \ge \varepsilon\right] \le 2\exp\left(-2t\varepsilon^2\right).$$

Assume without loss of generality that reward function assigns values form $[0,1]$. Since $\Delta_{\mathsf{reward}}(\cdot)$ is noticeable function, there exists a polynomial p, such that $\Delta_{\mathsf{reward}}(|x|) > p(|x|)^{-1}$ for all large enough x.

If we approximate $\mathrm{E}[\mathsf{reward}(x,0)]$, respectively $\mathrm{E}[\mathsf{reward}(x,1)]$ within $\varepsilon = \Delta_{\mathsf{reward}}(|x|)/3$, the proposed procedure successfully determines if $x \in L$. The Hoeffding's inequality guarantees that after $t = \ln(12)/2\varepsilon^2 = O(p(|x|)^2)$ evaluations of $\mathsf{reward}(x,b)$ we can approximate $\mathrm{E}[\mathsf{reward}(x,b)]$ within ε with probability greater than $5/6$. Therefore, one can decide L in randomized polynomial time with probability greater than $2/3$. □

Note that the proof can be easily extended for any $L \in \text{DRMA}[1, \log(|x|), \text{poly}(|x|)]$.

[10] Goldreich *et al.*[GVW02] in fact showed that NP-complete languages cannot have interactive proofs in which the prover sends only poly-logarithmically many bits to the verifier.

5 Conclusion and Open Problems

This work presents an alternative approach to constructing delegation schemes. Our approach, to the extent that it is meaningful, opens up the door for one-round protocols that demand minimal overhead for the prover and allow extremely fast verification. These dramatic efficiency improvements are a direct result of the relaxation in the soundness requirement. By penalizing the prover (on expectation) for even the slightest deviation from the correct computation, it is possible to focus the verification efforts on *local* parts of the computation (specifically paths along the computation), whose size is proportional to the depth of the circuit being computed. This stands in stark contrast to the more "rigid" approach to verification, whose soundness necessitates *global* dependence on the entire computation, and thus results in significantly more involved proof and verification procedures (inducing significant overhead beyond the actual computation).

Our work gives rise to several interesting open problems:

- While we give extremely laconic rational proofs for all NP languages (thus bypassing the impossibility result for interactive proofs), we also show that such succinct non-interactive public coin rational proofs NP cannot have non-negligible reward gap. An interesting research direction is to understand if this small gap is inherent. Specifically, can one construct succinct rational proofs for NP with large gap by increasing the number of rounds, or by having the verifier use private coins?
- In the context of the above question, it is natural to consider computationally bounded rational provers. Due to the impossibility result of Gentry and Wichs [GW11], constructions of succinct non-interactive arguments (SNARGs) in the classical setting must be based on non-falsifiable assumptions. Can one overcome this limitation in the rational model and construct rational SNARGs?
- The main advantage of our rational delegation is the utmost simplicity. Still, the complexity of verification does depend on the depth of the underlying circuit. In particular, for P-complete problems the scheme is no longer succinct. Kalai, Raz and Rothblum [KRR13] give a delegation scheme where the complexity of verification depends on the space of a Turing Machine instead of depth of circuit. Their scheme might potentially be simplified in the rational model, yielding a one-round succinct rational delegation for all problems in P.
- In this work, we propose to model rationality in presence of computational cost via noticeable reward gap. However, this modeling is not sensitive to different costs of computation while in reality a rational prover may behave differently on computations that have different complexity. It is an interesting open problem to refine our model, or even come up with a different approach, and capture rationality with respect to various measures of costs of computation.

References

[AIK10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2010.

[AM12] Pablo Daniel Azar and Silvio Micali. Rational proofs. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 1017–1028. ACM, 2012.

[AM13] Pablo Daniel Azar and Silvio Micali. Super-efficient rational proofs. In Michael Kearns, R. Preston McAfee, and Éva Tardos, editors, *ACM Conference on Electronic Commerce*, pages 29–30. ACM, 2013.

[BCI$^+$13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, pages 315–333, 2013.

[Bri50] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[CKV10] Kai-Min Chung, Yael Tauman Kalai, and Salil P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In Rabin [Rab10], pages 483–501.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern, editor, *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer, 1999.

[CMT12] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Practical verified computation with streaming interactive proofs. In Shafi Goldwasser, editor, *ITCS*, pages 90–112. ACM, 2012.

[Cra12] Ronald Cramer, editor. *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*. Springer, 2012.

[CRR13] Ran Canetti, Ben Riva, and Guy N. Rothblum. Refereed delegation of computation. *Inf. Comput.*, 226:16–36, 2013.

[DBL13] *Proceedings of the 45th Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, June, 1-4, 2013*. ACM, 2013.

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In Cramer [Cra12], pages 54–74.

[DLN$^+$04] Cynthia Dwork, Michael Langberg, Moni Naor, Kobbi Nissim, and Omer Reingold. Succinct NP proofs and spooky interactions. *Available at* `www.openu.ac.il/home/mikel/papers/spooky.ps`, 2004.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Rabin [Rab10], pages 465–482.

[GGPR12] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. *IACR Cryptology ePrint Archive*, 2012:215, 2012.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *STOC*, pages 113–122. ACM, 2008.

[GR07] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 321–340. Springer, 2010.

[GVW02] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1-2):1–53, 2002.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 99–108. ACM, 2011.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *STOC*, pages 723–732. ACM, 1992.

[KR09] Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 143–159. Springer, 2009.

[KR13] Gillat Kol and Ran Raz. Competing provers protocols for circuit evaluation. In Robert D. Kleinberg, editor, *ITCS*, pages 473–484. ACM, 2013.

[KRR13] Yael Tauman Kalai, Ran Raz, and Ron Rothblum. Delegation for bounded space. In *STOC* [DBL13]. To Appear.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Cramer [Cra12], pages 169–189.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[PRV12] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In Cramer [Cra12], pages 422–439.

[Rab10] Tal Rabin, editor. *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[RT92] John H. Reif and Stephen R. Tate. On threshold circuits and polynomial computation. *SIAM J. Comput.*, 21(5):896–908, 1992.

[RVW13] Guy Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: Delegating computation in sublinear time. In *STOC* [DBL13]. To Appear.

[Tha13] Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 71–89. Springer Berlin Heidelberg, 2013.